

# **Electronic Publishing**

Herbert Kopp  
Wintersemester 2001/02



<b>Inhaltsverzeichnis</b>	<b>i</b>
<b>Teil 1</b>	
<b>1. Einführung in das Elektronische Publizieren</b>	<b>1</b>
1.1 Historischer Überblick	1
1.2 HTML und PDF	3
1.2.1 Merkmale des HTML-Formats	3
1.2.2 Merkmale des PDF-Formats	5
1.2.3 Vergleich von HTML und PDF	6
1.3. Erzeugen von PDF-Dokumenten	7
1.4. Eigenschaften von PDF-Dokumenten	9
<b>2. Werkzeuge zum Publizieren mit PDF</b>	<b>11</b>
2.1 Acrobat-Installation	11
2.2 Distiller	11
2.2.1 Listbox „Einstellungen“ im Distiller-Hauptfenster	12
2.2.2 Menüfunktionen: Distiller-Grundeinstellungen	12
2.2.3 Menüfunktionen: Distiller-Voreinstellungen / Schriftordner ...	12
2.2.4 Menüfunktionen: Distiller-Voreinstellungen / Überwachte Ordner .	12
2.2.5 Menüfunktionen: Distiller-Voreinstellungen / Einstellungen ...	13
2.2.6 Menüfunktionen: Distiller-Voreinstellungen / Sicherheit ...	15
2.2.7 Menüfunktionen: Distiller-Voreinstellungen / Hilfe	15
2.3 PDFwriter	16
2.5 Acrobat	16
2.6 Catalog und Search	17
2.7 Capture und Reviewer	19
<b>3. PDF-Formulare</b>	<b>20</b>
3.1 Erstellen von PDF-Formularen	20
3.2 Serverseitige Maßnahmen	22
<b>4. Bilder in elektronischen Dokumenten</b>	<b>23</b>
4.1 Vektorbilder und Rasterbilder	23
4.2 Auflösung und Farbtiefe	24
4.3 Farbmodelle	27
4.3.1 Grundlegende Begriffe	27
4.3.2 Das RGB-Modell	28
4.3.2 Das CMY-Farbmodell	29
4.3.3 Das HSI-Modell	29
4.3.4 CIE-Modelle	30
4.3.5 Farbbereiche (Gamuts)	30
4.3.6 Arbeiten mit Farbmodellen	31

## Teil 2

<b>5.</b>	<b>Gestaltung elektronischer Dokumente</b>	<b>32</b>
5.1	Seitengröße und Seiten-Format	32
5.2	Lesbarkeit von Texten am Bildschirm	33
5.3	Methoden zur Erhaltung der Schriftqualität	34
5.4	Bilder und Grafiken	35
5.5	Typographische Grundlagen	39
<b>6.</b>	<b>PostScript-Grundlagen</b>	<b>44</b>
6.1	Die historische Entwicklung von PostScript	44
6.2	Merkmale von PostScript	44
6.3	PostScript-Interpreter (RIP)	46
6.4	PostScript-Treiber	48
6.5	DSC-Kommentare	50
6.6	Übertragungsprotokolle	51
6.7	PostScript-Fehler	52
<b>7.</b>	<b>PDF-Dateien</b>	<b>53</b>
7.1	Die PDF-Dateistruktur	53
7.2	PDF-Objekte	53
7.3	PDF-Datei-Aufbau	54
7.4.	PDFmark-Operatoren	59
<b>8.</b>	<b>PDF-Formulare – noch einmal betrachtet</b>	<b>61</b>
8.1	Vordefinierte Funktionen für Formularfelder	61
8.2	JavaScript in Formularen	64
8.3	Serverseitige Verarbeitung von Formulardaten	66
	<b>Literatur</b>	<b>68</b>

# Teil 1

# 1. Einführung in das Elektronische Publizieren

## 1.1 Historischer Überblick

Die zentrale Bedeutung der Schrift in (fast) allen entwickelten Kulturen, begründet auch die Tragweite, welche den Technologien zur Erstellung schriftlicher Dokumente zukommt.

### Die Anfänge

Bis zur Erfindung des Buchdrucks war das Erstellen und Lesen von Dokumenten wenigen Spezialisten vorbehalten, die daraus ihre soziale Stellung und ihren Einfluss herleiteten. Einige Stationen dieser Entwicklung sind

- die Keilschrift der Sumerer mit auf noch nicht erstarre Tontafeln eingedrückten Zeichen
- die Hieroglyphen der ägyptischen Tempelpriester, die stilisierte Bildzeichen waren
- die bereits auf Lautzeichen aufgebaute Schrift der phönizischen Kaufleute
- die Schrift der griechischen und römischen Gelehrten, deren Zeichensätze die direkten Vorläufer unserer heutigen Alphabete sind.

### Gutenberg

In der Mitte des 15. Jahrhundert brachte ein Zusammentreffen mehrerer Faktoren eine entscheidende Wende:

- Das Schriftaufkommen hatte in Europa zugenommen und immer mehr Menschen beherrschten das Lesen oder sogar das Schreiben.
- Die Herstellung von Papier aus Pflanzenfasern war gelungen.
- Erste Druckverfahren mit Holzschnitten wurden erprobt. Dabei wurde der Inhalt ganzer Seiten in Holzblöcke geschnitzt.
- Die metallurgischen Kenntnisse erlaubten nun einzelne Lettern zu gießen.

Damit war um 1445 der Weg frei für die Erfindung und Durchsetzung des Buchdrucks mit gegossenen, beweglichen Lettern durch den Goldschmied Johann Gensfleisch zum Gutenberg.

### Die Industrialisierung der Druckkunst

Bis 1799 war die Herstellung von Papier ein aufwendiger Prozess: Aus pflanzlichen Fasern, z.B. Holz, Hadern (Leinen- und Baumwoll-Lumpen) und Wasser wird dabei ein gereinigter, entfärbter Brei hergestellt. Das Mischungsverhältnis beträgt 1 Teil Fasern auf 100-200 Teile Wasser. Dazu werden Füllstoffe gemischt (Kaolin, Farben, Leim, Talkum, Gips). Daraus werden mit feinen Sieben einzelne Blätter von Hand geschöpft, gepresst, getrocknet und beschnitten.

Im Jahr 1799 wurde in England von Robert eine Maschine erfunden, die eine industrielle Herstellung von Papier ermöglichte. Das bis dahin seltene und teure Papier wurde in kurzer Zeit ein billiges Massenprodukt.

Im Jahr 1812 folgte die Entwicklung der Schnellpresse mit einem Durchsatz von 1100 Bögen in der Stunde. Sie wurde bei der London Times eingesetzt. Ein weiterer Technologiesprung erfolgte 1848 mit der Rotationspresse, die 8000 Drucke pro Stunde anfertigen konnte.

In der zweiten Hälfte des 19. Jahrhunderts wurden mit der Erfindung photographischer Techniken die Photogravur und die Photolithographie möglich. Parallel dazu wurde ab etwa 1885 der manuelle Satz durch Setzmaschinen abgelöst (Linotype).

## **Digitale Techniken in der Dokumentenproduktion**

Seit der Mitte des 20. Jahrhunderts haben nach und nach neue Technologien die traditionellen Verfahren in alle Phasen der Dokumentenerstellung ersetzt:

Der Bleisatz wurde zunächst ersetzt durch Photosatzsysteme und Belichter.

- Die elektronische Bildbearbeitung ersetzte die photographischen Verfahren der Bildproduktion und Bildbearbeitung
- Die Einführung von DTP-Systemen führte viele der zuvor getrennte Herstellungsprozesse auf einem Arbeitsplatz zusammen.
- Die Einführung hochauflösender digitaler Kameras wird in den nächsten Jahren einen weiteren Technologiesprung zur Folge haben.

## **Elektronische Dokumente**

Digitale Verfahren haben bis in die jüngste Vergangenheit hauptsächlich der Herstellung von Druckerzeugnissen auf Papier gedient. Seit wenigen Jahren spielen aber auch elektronische Medien als Endprodukt eine Rolle:

- CD-ROM-Produkte wie Lexika, Lernsysteme, Warenhauskataloge, Fahr- und Flugpläne, Telefonverzeichnisse
- Elektronische Zeitschriften und Bücher, die entweder ausschließlich oder zusätzlich zur Papierversion über Netze genutzt werden können.

Es wird vermutet, dass die Marktanteile solcher Medien im Vergleich zu papiergebundenen Medien für die überschaubare Zukunft relativ gering sein werden. (15% Marktanteil bis 2000). Da auch konventionelle Printprodukte weiterhin Vorteile aufzuweisen haben, rechnet man nicht mit einer schnellen oder gar vollständigen Verdrängung. Ungeklärt sind für elektronische Medien noch zahlreiche Fragen, die für konventionelle Printmedien längst geklärt sind, z.B.

- das Problem der langfristigen Archivierung von Dokumenten
- die Gefahr der nachträglichen Veränderung und Verfälschung von Schriften
- die Gefahr der wirtschaftlichen Schädigung der Verlage und Autoren
- die Erleichterung geistigen Diebstahls von Inhalten
- die Lieferungs- und Abrechnungsverfahren für elektronische Dokumente im Netz.

Dem stehen aber auch unübersehbare Vorteile digitaler Medien gegenüber:

- In elektronischen Dokumentenbeständen können die benötigten Informationen mit automatisierten Verfahren viel leichter als in konventionellen Bibliotheken gesucht und gefunden werden.
- Die Lagerung und der Transport werden billiger.
- Mit Printing on Demand könnten wesentliche Kosteneinsparungen realisiert werden.
- Digitale Medien können gegenüber Printmedien echte Mehrwerte bieten. Beispiele sind
  - die Einbindung weiterer Informationsträger wie Animationen und Audio- und Video
  - die Interaktivität, wie sie z.B. in Lernprogrammen und Spielen üblich sind
  - Zusatzfunktionen, z.B. Bestellverfahren in einem elektronischen Warenhauskatalog.

## 1.2 HTML und PDF

Während bei der Erstellung von Dokumenten die unterschiedlichsten Werkzeuge und Formate verwendet werden, sind als Distributions- und Präsentationsformate nur zwei von universeller Bedeutung, HTML und PDF.

### 1.2.1 Merkmale des HTML-Formats

Die Voraussetzung dafür, dass schriftliche Dokumente mit Hilfe von Computern bearbeitet und ausgewertet werden können, sind Beschreibungssprachen, mit denen man die logische Struktur von Textdokumenten repräsentieren kann.

Unter den verschiedenen Alternativen hat sich der ISO-Standard SGML etabliert (Standard Generalized Markup Language). Das Grundprinzip von SGML ist die Trennung der Dokumentenstruktur und des Dokumenten-Layout. SGML hat als Standardformat für wissenschaftliche Publikationen samt Abstracts, Glossaren, Literaturlisten usw. Bedeutung erlangt (IEEE-Zeitschriften, Springer/Link, ISO-Normen, ... ). Wegen seiner hohen Komplexität hat SGML bisher keine allgemeine Verbreitung gefunden.

Das Dokumentenformat HTML (Hypertext Markup Language) entstand aus der Sprache SGML. HTML ist wesentlich einfacher, eignet sich allerdings nicht zum Strukturieren umfangreicherer Dokumente z.B. kompletter Bücher, denn Strukturmerkmale lassen sich nur sehr begrenzt damit darstellen und sind meist eng mit Layout-Konventionen verknüpft, die einem Strukturelement wie einer Kapitelüberschrift eine ganz bestimmte Darstellung zuordnen.

Das HTML-Format ist optimiert auf die folgenden Aspekte der Dokumenten-Darstellung:

- Beschreibung der Dokumentenstruktur  
Überschriften, Absatzformate, Font-Einstellungen werden als "Tags" in den Text eines Dokuments eingestreut und sind genau wie der Dokumenten-Inhalt als ASCII-Text codiert.
- Hypertext-Funktionalität  
Verknüpfungen (Links) zu anderen Dokumenten auf dem selben Rechner oder über das http-Protokoll zu anderen Rechnern im weltweiten Netz.
- Effiziente Übertragung über das Netz  
HTML-Dokumente bestehen in der Regel aus Einzeldateien, die entweder nur eine Seite oder einen Abschnitt des Dokuments umfassen. Bilder und andere Bestandteile (z.B. Shockwave-Objekte) sind als selbständige Dateien eingebunden. Damit lassen sich die zur Darstellung einer Seite benötigten Informationen zur Übertragung exakt auswählen.

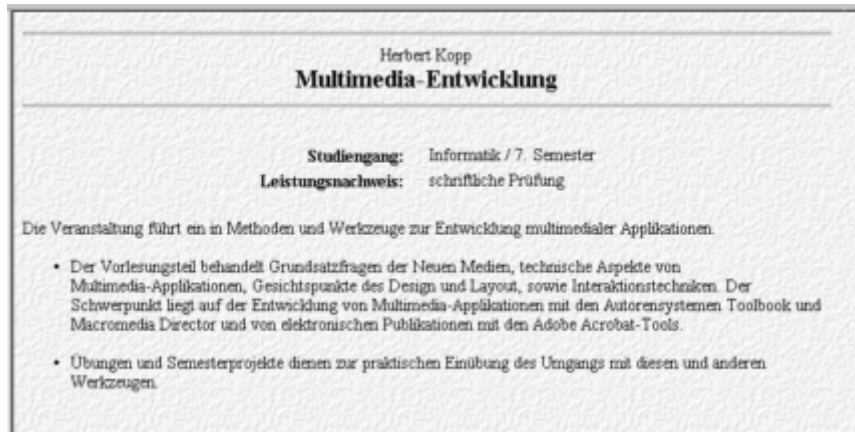
Es war nicht das Ziel, mit HTML komplexe Layouts von Dokumenten zu beschreiben. Es bleibt dem Viewer überlassen, die für die jeweilige Plattform optimale Darstellung zu finden. Sobald jedoch ein Dokument mehr als schlichten Text enthält, ist dieser Ansatz nicht mehr durchzuhalten mit der Folge, dass

- phantasievolle Tricks notwendig sind, um mit dem jeweiligen HTML-Standard eine bestimmte Darstellung zu erreichen und
- Erweiterungen der HTML-Syntax um neue Stil-Elemente geschaffen wurden, um eine bessere Steuerung des Layout durch den Autor zu ermöglichen, z.B. "Cascading Style Sheets"

Unter der Bezeichnung XML (Extensible Markup Language ) wird daher derzeit vom World-Wide-Web-Consortium(W3C) eine SGML-Adaption für das WWW erarbeitet, die diese Nachteile beheben soll.



## Beispiel eines HTML-Dokuments



```
<HTML>
<HEAD>
<TITLE>H.Kopp:  Bildverarbeitung </TITLE>
</HEAD>

<BODY TEXT="#000000" BGCOLOR="#FFFFFF" LINK="#0000EE" VLINK="#551A8B"
ALINK="#FF0000" BACKGROUND="./bilder/hg_main.gif">

<HR>
<CENTER><FONT SIZE=-1>Herbert Kopp</FONT></CENTER>
<CENTER><B><FONT SIZE=+1>Multimedia-Entwicklung</FONT></B></CENTER>

<HR>
<BR>
<TABLE BORDER=0 COLS=2 WIDTH="100%" >
<TR>
  <TD ALIGN=RIGHT><B><FONT SIZE=-1>Studiengang:&nbsp;</FONT></B></TD>
  <TD ALIGN=LEFT><FONT SIZE=-1>Informatik / 7. Semester&nbsp;</FONT></TD>
</TR>

<TR>
  <TD ALIGN=RIGHT><B><FONT SIZE=-1>Leistungsnachweis:&nbsp;</FONT></B></TD>
  <TD ALIGN=LEFT><FONT SIZE=-1>schriftliche Pr&uuml;fung</FONT></TD>
</TR>
</TABLE>

<P>
<FONT SIZE=-1>
Die Veranstaltung führt ein in Methoden und Werkzeuge zur Entwicklung mul-
timedialer Applikationen.
<UL>
  <LI> Der Vorlesungsteil behandelt Grundsatzfragen der Neuen Medien, tech-
nische Aspekte von Multimedia-Applikationen, Gesichtspunkte des De-
sign und Layout, sowie Interaktionstechniken. Der Schwerpunkt liegt
auf der Entwicklung von Multimedia-Applikationen mit den Autorensys-
temen Toolbook und Macromedia Director und von elektronischen Pu-
blikationen mit den Adobe Acrobat-Tools. <P>
  <LI> Übungen und Semesterprojekte dienen zur praktischen Einübung des Um-
gangs mit diesen und anderen Werkzeugen.
</UL>
</BODY>
</HTML>
```

### 1.2.2 Merkmale des PDF-Formats

Das PDF (Portable Document)-Format wurde von der Firma Adobe, dem 'Erfinder' von Postscript entwickelt als ein Distributions- und Präsentationsformat, das

- unabhängig von der Generierungs-Plattform und
- unabhängig von der Darstellungsplattform

das Layout von Dokumenten beschreibt. Die Basis von PDF ist die im Bereich der Druckvorstufe zum Standard gewordene Dokumentenbeschreibungssprache Postscript. Wie Postscript garantiert auch PDF eine überall identische Darstellung des Dokumenten-Layout unabhängig vom speziellen Ausgabegerät, dem Darstellungs-Maßstab usw.

Das PDF-Format ist optimiert auf die folgenden Aspekte der Dokumenten-Darstellung:

- *Layout-Beschreibung von Dokumenten*  
Wie mit Postscript können die Bestandteile eines komplexen, hochwertigen Seiten-Layout in einem PDF-Dokument beschrieben werden. Da PDF als Distributionsformat konzipiert ist, sind Änderungen am Inhalt eines PDF-Dokuments nur sehr beschränkt möglich.
- *Kompressionstechniken für Texte und graphische Objekte*  
Die Kompression von Texten und unterschiedlicher Typen von Graphiken kann exakt auf die geplante Verwendung des Dokuments ausgerichtet werden.
- *Übertragung von Dokumenten über das Netz*  
PDF-Dateien enthalten in der Regel ein vollständiges Dokument, z.B. ein ganzes Buch einschließlich der Abbildungen. Für die Übertragung in Netzen sind sie mit einem Index ausgestattet, so dass auf die zur Darstellung einer Seite benötigten Objekte gezielt zugegriffen werden kann. Neuere WEB-Server nutzen dies mit Hilfe der Byte-Serving-Technik aus, um die Übertragung zu optimieren.

Im Gegensatz zu Postscript ist PDF jedoch keine Programmiersprache. Es übernimmt die Postscript-Elemente zur Beschreibung des Seiten-Layout, nicht aber Features wie Variablen, Bedingungen, Schleifen usw. Im Gegensatz zu SGML/HTML sind in PDF keine Struktur-Informationen enthalten. Adobe arbeitet jedoch an einer Integration von SGML in das PDF-Format.

Als primäre Einsatzbereiche hatte man folgende Bereiche im Auge

- den Schriftverkehr in Unternehmen und Behörden (papierloses Büro)  
Dabei dachte man nicht nur an die Erstellung, Vervielfältigung, Zustellung und Archivierung elektronischer Dokumente, sondern auch an die Digitalisierung und Archivierung konventioneller Papierdokumente.
- den Dokumentenaustausch zwischen kooperierenden Partnern in der Druckvorstufe (Autoren, Graphiker, Satzbetriebe, Druckereien, ...).

Das PDF-Format ist lizenzpflichtig, d.h. Gerätehersteller, die PDF nutzen, müssen Lizenzgebühren bezahlen. Tools zum Erzeugen und zum Bearbeiten von PDF-Dateien sind bis auf wenige Ausnahmen (z.B. Ghostscript) kostenpflichtig. PDF-Viewer werden von Adobe jedoch ohne Gebühren an jedermann abgegeben.

Das PDF-Format ist offengelegt, so dass die Entwicklung von Software-Werkzeugen rund um PDF jedermann möglich ist. Dazu liefert Adobe eine Development Kit und für zahlreiche Funktionen gibt es Zusatzmodule als Plugins.

Die folgende Seite zeigt das Beispiel einer PDF-Seite. Verifizieren Sie, dass die Qualität der Text- und Liniengraphik, nicht aber der eingebundenen Pixelgraphik unabhängig vom Darstellungsmaßstab sind.

### 1.2.3 Vergleich von HTML und PDF

Bedingt durch die unterschiedliche Entstehungsgeschichte haben beide Formate ihre eigenen Stärken und Schwächen. Sie werden sich aber durch die bereits angekündigten Erweiterungen künftig weiter einander annähern. Die folgende Tabelle (nach [Merz1]) stellt ihre wesentlichen Unterschiede gegenüber.

Merkmal	HTML	PDF
Verantwortung für das Format	W3-Konsortium und Browser-Hersteller	Adobe
Standardisierung	zahlreiche Varianten	ja
Viewer für verschiedene Plattformen	frei	frei
Werkzeuge zur Bearbeitung	Zahlreiche kostenpflichtige und freie Tools	kostenpflichtige und wenige freie Tools
Konvertierung existierender Dokumente	Teuer, zeitaufwendig	einfach
Editieren existierender Dokumente	einfach, mit Text- und HTML-Editoren	nur eingeschränkt möglich
Erzeugen neuer Dokumente	mit HTML-Editoren	durch Generieren
Metadaten	Meta-Tags	Document Info-Felder
Erweiterungen	Plugins	Plugins
Struktur-Information	vorhanden	nicht vorhanden
Suchbarkeit	gut	gut
Hypertext-Funktionalität	sehr gut	sehr gut
Layout-Qualität	schwach	sehr gut
Einfluss auf die Darstellung	Leser und Autor	Autor
Anpassung des Layout an das Gerät	ja	nur bzgl. Maßstab
Dokument in einer einzigen Datei	nein	ja (Ausnahme: Filme)
Eignung für Screen-Darstellung	sehr hoch	hoch, falls dafür formatiert
Eignung für Ausdruck	niedrig	hoch, falls dafür formatiert
Layout und Typographie	schlecht	sehr gut
Datei-Format	Einfache ASCII-Tags	komplex
Dynamische Generierung am Server	leicht	schwierig
Größe einzelner Dateien	klein	unterschiedlich
Formulartechniken	gut	sehr gut

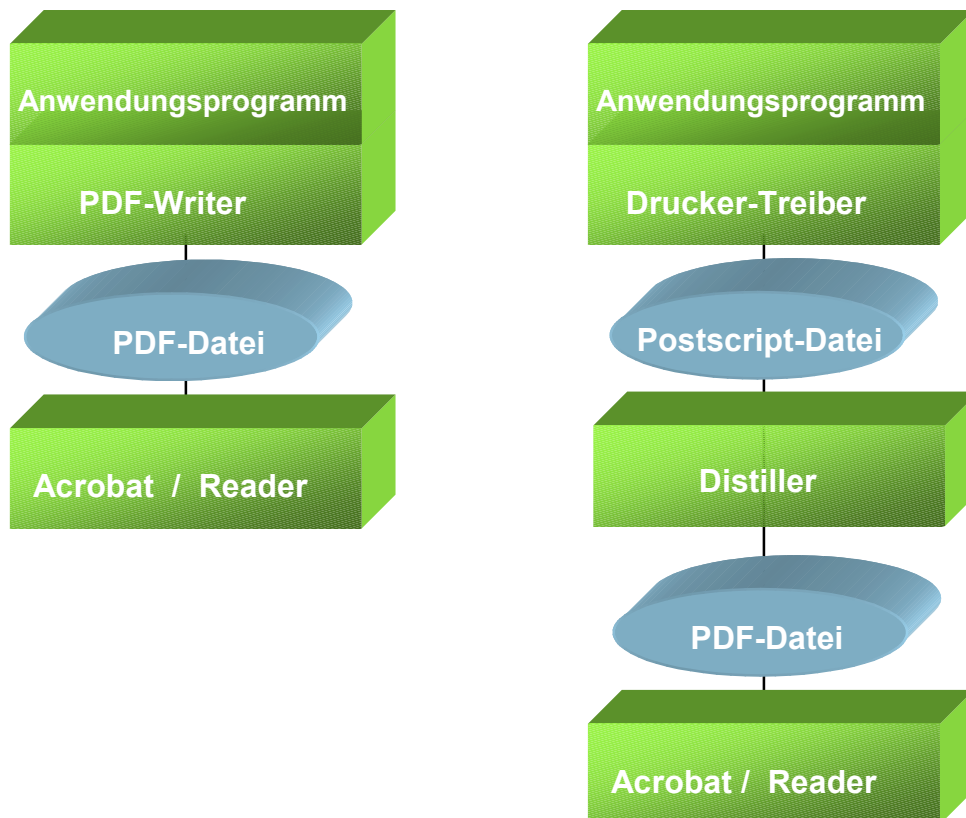
### 1.3. Erzeugen von PDF-Dokumenten

#### Werkzeuge von Adobe für PDF

Adobe selbst stellt mit den Acrobat-Tools folgende Werkzeuge bereit zur Erzeugung und Bearbeitung von PDF-Dokumenten:

- PDF-Reader: frei nutzbarer Reader für PDF-Dateien
- PDF-Writer: Druckertreiber zur Generierung von PDF-Dateien für den Bürobereich
- Distiller: Utility zum Generieren von PDF-Dateien aus Postscript-Dateien
- Acrobat: Utility zum Bearbeiten von PDF-Dateien
- Capture Acrobat-Plugin zum Erfassen konventioneller Dokumente als PDF-File
- Search Acrobat-Plugin zur Suche in indexierten Datenbeständen
- Catalog Indexierer für einen Datenbestand

Die folgenden Abbildungen zeigen die unterschiedlichen Möglichkeiten zum Generieren von PDF-Dateien.



#### Erzeugen von PDF-Dateien mit dem PDF-Writer

Dieser wird als Druckertreiber im Betriebssystem installiert, so dass alle Anwendungen, die eine Druckfunktionalität besitzen, ihn nutzen können. Der PDF-Writer ist für den Bürobereich gut geeignet, nicht jedoch dort, wo Dokumente mit EPS-Graphiken oder pdfmarks konvertiert werden. Gegenüber dem Distiller sind seine Einstellmöglichkeiten geringer (z.B. für Fonteinbettung, Kompression von Bildern)

### **Erzeugen von PDF-Dateien mit dem Acrobat Distiller**

Der Distiller ist ein Konverter, der eine Postscript-Datei liest und sie in eine PDF-Datei umwandelt. Er kann in verschiedener Weise eingesetzt werden:

- Im Hintergrund laufend kann er selbständig Verzeichnisse überwachen und alle PS-Dateien in diesen übernehmen und umwandeln.
- Als selbständige Anwendung kann man PS-Dateien mit ihm öffnen und umwandeln.
- Durch Doppelklick auf Dateien mit der Extension ".PS" startet man ihre Konvertierung.
- Über den Distiller-Assistenten wird er wie der PDF-Writer über das Druck-Menü von Anwendungen aufgerufen.

Voraussetzung ist, dass ein Postscript-Druckertreiber installiert ist. Das kann ein beliebiger PS-Druckertreiber sein. Es wird jedoch empfohlen die Original-Adobe Postscript-Treiber zu verwenden.

### **Acrobat**

Acrobat ist die Anwendung, mit der PDF-Dokumente bearbeitet werden können. Einige der Funktionalitäten von Acrobat sind:

- Seiten einfügen/löschen/ersetzen/drehen/beschneiden
- Links erzeugen zu anderen Stellen innerhalb des Dokumentes und nach außen
- Formulare erstellen und Bearbeiten
- Texterkennung (mit Hilfe des Capture-Plugin)
- Kleine Änderungen im Text durchführen. Acrobat ist aber kein Textverarbeitungssystem !

### **Acrobat Capture**

Acrobat Capture liest eingescannte Papierdokumente, die als Bitmap-Graphik vorliegen und erzeugt daraus PDF-Dokumente. Dabei werden Texte soweit wie möglich mit OCR erkannt, alle übrigen Bestandteile der eingescannten Dokumente werden als Bitmap eingebunden. Das Resultat sind Dokumente, die im Layout dem Original entsprechen, aber trotzdem durchsuchbar sind. Das Hauptanwendungsgebiet ist die Archivierung von Dokumenten, zu denen es keine elektronischen Quelltexte gibt.

### **pdfmarks**

Mit der pdfmark-Schnittstelle hat Adobe eine weitere Schnittstelle geschaffen, über die Funktionalitäten des PDF-Dokuments (z.B. Hyperlinks) bereits im Quelltext eingebaut werden können. pdfmarks werden beim Erstellen der Postscript-Datei in den Postscript-Code eingeschleust und von normalen Druckern ignoriert. Der Distiller erzeugt daraus aber die entsprechenden Objekte im PDF-Code.

Diese Technik ist z.B. für die automatische Generierung von Verweisen in Dokumenten wichtig, z.B. vom Inhaltsverzeichnis auf die Text-Abschnitte, vom Text auf Abbildungen, vom Stichwortverzeichnis auf den Text. Sie ist bisher nur in wenigen Text- und Publishing-Programmen integriert (Framemaker, Ventura Publisher). Mit MS Word kann man über die Feldfunktion {DRUCK ... } pdfmarks "von Hand" erstellen, jedoch ist noch keine automatische Generierung möglich.

## 1.4. Eigenschaften von PDF-Dokumenten

### Portabilität des Formats

Es sind bei der Erstellung von PDF-Dateien mit dem Distiller zwei Dateiformate möglich:

- 7-bit ASCII-Format und
- binäres PDF-Format

Es wird empfohlen, das Binärformat zu verwenden, da ASCII-Textfiles bei der Übertragung oft in unvorhersehbarer Weise verändert werden (z.B. Zeilenvorschub-Behandlung).

### Kompressionstechniken

Objekte werden je nach Objekttyp mit unterschiedlichen Verfahren komprimiert:

- JPEG                      wird für Farbige und graue Pixelbilder eingesetzt.
- LZW und RLE            wird für Binärbilder eingesetzt
- LZW                        wird für Texte und Zeichnungen eingesetzt

Beim Distiller können die Art und die Stärke der Kompression detailliert eingestellt werden.

### Font-Behandlung bei PDF (Multiple Master-Fonts)

Voraussetzung einer originalgetreuen Darstellung ist immer, dass auf der Darstellungs-Plattform die gleichen Fonts vorhanden sind wie auf der Autorenplattform. Bisher hatte man dazu zwei Möglichkeiten:

- Der Autor lieferte zusammen mit seinem Dokument auch die Fonts und Einstellparameter für den Viewer mit.
- Jede Seite wird bereits auf der Autorenplattform in eine Graphik verwandelt mit fester Auflösung.

PDF bringt für dieses Problem eine dritte sehr wichtige Alternative:

Für jede Font enthält das Dokument einen Font-Deskriptor mit dem Fontnamen, der Zeichenmetrik und dem Zeichenstil. Der Platzbedarf dafür beträgt 1 - 2 KB pro Dokument. Diese Information wird auf der Zielplattform benutzt, um die Originalfont zu simulieren, falls diese nicht selbst vorhanden ist.

Alternativ können Fonts auch in die Dokumente eingebunden werden, sogar eingeschränkt auf die tatsächlich benötigten Zeichen. Für Symbolfonts und nicht-lateinische Schriften ist das der einzige Weg.

### Random Access zu einzelnen Seiten

Eine Postscript-Datei verlangt zu ihrer Darstellung, dass zunächst alle vorausgehenden Seiten interpretiert werden müssen. Die Darstellung einer Seite hängt dort also nicht nur von ihrer eigenen Komplexität, sondern auch von allen vorausgehenden Seiten ab.

Demgegenüber enthalten PDF-Dateien immer eine Cross Reference-Tabelle für den direkten Zugriff auf einzelne Seiten. Das Layout einer Seite wird unabhängig von den vorausgehenden Seiten vollständig beschrieben.

### Inkrementelles Update und Optimierung von PDF-Dateien

Wenn eine PDF-Datei bearbeitet wird, bleiben die Originaldaten trotzdem in ihr erhalten. Modifikationen werden ans Ende angehängt. Die Vorteile dieser Methode sind:

- kurze Modifikations-Zeiten
- Möglichkeit für eine Undo-Funktion

Es empfiehlt sich daher mit der Funktion "Speichern unter ..." und aktiviertem Schalte "Optimize" alle nicht mehr aktiven Objekte entfernen zu lassen.

### **Verschlüsselung und Passwortschutz**

PDF-Dokumente können mit Acrobat bei der Funktion "Speicher unter ..." mit Sicherheitseinstellungen versehen werden. Im einzelnen sind dies

- Ein Passwort zum Öffnen der Datei
- Ein Passwort zum Verändern der Sicherheitsoptionen. Diese Optionen können einzeln eingestellt werden und sperren:
  - das Drucken
  - das Übernehmen von Texten/Graphiken ins Clipboard
  - das Verändern des Dokuments

das Einfügen/Verändern von Notizen im Dokument.

### **Digitale Unterschriften in PDF-Dokumenten**

Mit Acrobat kann man ein Dokument elektronisch signieren und die in einem Dokument enthaltenen Unterschriften anzeigen und überprüfen. Jede digitale Unterschrift speichert Informationen über die unterzeichnende Person und den Status des Dokuments bei der Unterzeichnung.

## 2. Werkzeuge zum Publizieren mit PDF

In diesem Kapitel werden die Programme vorgestellt, die zum Acrobat-Toolset gehören. Der Text ist an der Programmversion Acrobat 4.0 orientiert.

### 2.1 Acrobat-Installation

Die Installation von Acrobat erfolgt mit einem Setup-Programm, so dass zumindest für Standard-Installationen keine Probleme auftreten. Was man zusätzlich beachten sollte:

- Wenn Sie ältere Acrobat-Versionen nach der Installation von Acrobat 4 deinstallieren, müssen Sie Acrobat 4 neu installieren.
- Wenn Sie Ihren Internet-Browser neu installieren, kann es vorkommen, dass das Acrobat Plugin des Browser verloren geht. Am einfachsten ist es, danach Acrobat noch einmal zu installieren. Alternativ können Sie die Plugins auch direkt in das Plugin-Verzeichnis des Browser kopieren (mehr dazu in [1], Seite 9ff)
- Wenn Sie keine Acrobat-Standard-Installation durchführen, sollten Sie vor der Installation von Acrobat in jedem Fall den Adobe-PostScript-Treiber Version 5 installieren (steht auf der Link-Seite zum Download). Dieser hat mehr Einstellmöglichkeiten, als der Standard-PostScript-Treiber von Windows, z.B. die Fähigkeit, mehrere Seiten auf ein Blatt zu drucken.

### 2.2 Distiller

Der Acrobat Distiller ist das Programm zum Erzeugen von PDF-Dateien. Der Distiller kann PostScript-Dateien lesen und daraus PDF-Dateien erzeugen. Ab der Version 4.0 versteht der Distiller das PostScript Level 3. Er kennt dazu im wesentlichen drei Verfahren:

- Distiller kann als Hintergrund-Prozess laufen (z.B. als NT-Dienst) und Verzeichnisse daraufhin überwachen, ob in ihnen PostScript-Dateien abgelegt werden, diese selbständig übernehmen und daraus PDF-Dateien erzeugen.
- Im Drucker-Menü beliebiger Anwendungen erscheint der Eintrag „Acrobat Distiller“. Wenn Sie darüber eine Druckausgabe erzeugen, wird (ohne, dass das für die Anwender erkennbar ist) zunächst eine PostScript-Datei erzeugt, der Distiller zum Erzeugen der PDF-Datei aktiviert und die PDF-Datei ggf. am Bildschirm angezeigt.
- Sie können Distiller als Anwendung starten und mit der Funktion „Datei / öffnen ...“ eine PostScript-Datei öffnen, die dann in eine PDF-Datei konvertiert wird.

Damit können alle Anwendungen, die einen PostScript-Drucker bedienen können, auch PDF erzeugen.

Im folgenden werden die Einstellungen beschrieben, die man vornehmen kann, wenn man den Distiller über das Windows-Startmenü aufruft. In der Regel wird man diese Einstellungen nach der Installation überprüfen und ggf. ändern und danach nur selten modifizieren.

Es werden hier nur die für das tägliche Arbeiten Relevanten Optionen genauer beschrieben. Eine ausführliche Dokumentation enthalten Das Acrobat-online-Handbuch und [4]



### **2.2.1 Listbox „Einstellungen“ im Distiller-Hauptfenster**

Wenn man den Distiller über das Windows-Startmenü aufruft, zeigt das Anwendungsfenster eine Listbox an, über Betriebsarten mit voreingestellten Parameter-Kombinationen ausgewählt werden können:

- Druckoptimiert Diese Einstellungen sind für PDF-Dateien konzipiert, die für Drucker oder digitale Satzsysteme bestimmt sind oder auf einer CD werden.
- KompriOptimiert Diese Einstellungen sind für PDF-Dateien konzipiert, die als hochwertige Endausgabe gedruckt werden sollen. Kompatibel mit Acrobat 4.0 und neuer.
- MonitorOptimiert: Diese Einstellungen sind für PDF-Dateien konzipiert, die im World Wide Web oder einem Intranet angezeigt oder über ein E-Mail-System für die Online-Anzeige verteilt werden sollen. Kompatibel mit Acrobat 3.0 und neuer.

Je nach dem beabsichtigten Einsatz der konvertierten Dateien wird aus diesen vordefinierten Betriebsarten die geeignetste ausgewählt und nur in besonderen Fällen die Parameter-Einstellungen einzeln ändern.

### **2.2.2 Menüfunktionen: Distiller-Grundeinstellungen**

Mit der Funktion „Datei / Grundeinstellungen“ können einige Betriebseinstellungen gewählt werden. Empfehlenswert für die normale interaktive Nutzung ist es

- alle Warnungen zu aktivieren
- Die Ausgabeoption „Eingabeaufforderung für die PDF-Datei“ zu aktivieren. Dadurch wird vor dem Erzeugen einer PDF-Datei ein Datei-Auswahl-Menü gezeigt, mit dem man den Namen und Ort der PDF-Datei einstellen kann.
- Die Optionen „Bei Verwendung des Distiller(-Druckers) PDF anzeigen, zeigt die PDF-Datei nach ihrer Erzeugung am Bildschirm an.“

### **2.2.3 Menüfunktionen: Distiller-Voreinstellungen / Schriftordner ...**

Bei der Acrobat-Installation wird hier nur das Verzeichnis eingetragen, in dem die zusammen mit Acrobat installierten PostScript-Schriften liegen, z.B. das Directory D:\Programme\Adobe\Acrobat4.0\Resource\Font

Wenn Sie auf Ihrem Rechner weitere PostScript-Fonts installiert haben, sollten Sie die zugehörigen Verzeichnisse zusätzlich hier eintragen.

### **2.2.4 Menüfunktionen: Distiller-Voreinstellungen / Überwachte Ordner ...**

Mit dieser Einstellung können Sie Verzeichnisse angeben, die überwacht werden sollen. Wenn in diesen PostScript-Dateien abgelegt werden, kann Der Distiller diese selbständig übernehmen und daraus PDF-Dateien erzeugen.

Diese Einstellung ist dann sinnvoll, wenn Acrobat auf einem Server als Konverter für eine große Zahl von Benutzern eingesetzt ist. Für einen Einzelanwender dürfte sie weniger in Frage kommen.

## **2.2.5 Menüfunktionen: Distiller-Voreinstellungen / Einstellungen ...**

Diese Funktionsgruppe ist für die Feineinstellung von Distiller am wichtigsten.

### **Allgemein / Kompatibilität:**

Wenn man Wert darauf legt, dass PDF-Dateien auch von Anwendern problemlos gelesen werden können, die noch nicht den (kostenlosen) Acrobat Reader 4 installiert haben, sollte man auf Acrobat 3-Kompatibilität achten.

Meistens werden aber Acrobat 4-kompatible Dateien von Acrobat 3 problemlos gelesen und angezeigt. Beim Laden wird eine Warnungsmeldung ausgegeben.

### **Allgemein / ASCII-Format**

Mit dieser Option kann man PDF-Dateien im ASCII-Format generieren. Diese sind wesentlich größer, können aber mit einem ASCII-Editor bearbeitet werden.

### **Allgemein / PDF optimieren**

Sollte immer gewählt werden. Ermöglicht u.a. das Übertragen und Anzeigen einzelner Seiten einer PDF-Datei ohne dass dazu die ganze Datei vom Server geholt werden muss.

### **Allgemein / Thumbnails erstellen**

Sollte nur in begründeten Fällen benutzt werden, da sie pro Seite ca 3 KB zusätzlichen Speicherbedarf kosten.

### **Allgemein / Bindung**

Sollte immer auf „links stehen“. Außer bei arabischen, japanischen o.ä. Texten, die von rechts nach links oder von oben nach unten gelesen werden

## **Komprimierung / ....**

Distiller erlaubt es, unterschiedliche Bestandteile eines Dokuments nach verschiedenen Methoden zu komprimieren:

- Einstellung der Auflösung
- Neuberechnung / Downsampling:  
Dabei wird das Bild mit der eingestellten Auflösung neu abgetastet. Die Pixel des Zielbildes können nach unterschiedlichen Verfahren aus der Quelle ermittelt werden.
- Komprimierung:  
Es werden verlustlose Komprimierungsverfahren für Text und Vektorgrafik eingesetzt, Rasterbilder werden mit dem verlustbehafteten JPEG-Verfahren komprimiert. Dabei hat der Grad der Kompression entscheidenden Einfluss auf die Qualität.

Kompression ist immer ein Kompromiss zwischen der Qualität des Endprodukts und dem dafür erforderlichen Speicherplatz. Es ist daher empfehlenswert, die Werte der Voreinstellungen „Druckoptimiert“, „KompriOptimiert“ und „MonitorOptimiert“ nur in begründeten Fällen zu ändern.

Für eine ausführliche Beschreibung der angebotenen Kompressions-Optionen wird auf das Acrobat.Benutzerhandbuch verwiesen.

## **Schriften / ...**

Diese Parametergruppe ermöglicht es, die Einbettung von Schriften in die PDF-Dateien zu steuern. Diese Thematik spielt dann eine Rolle, wenn die PDF-Datei auf Rechnern angezeigt werden soll, bei denen die von ihr benutzten Fonts nicht installiert sind.

- Wenn eine PDF-Datei alle Schriften, die sie zur Darstellung braucht, selbst enthält, kann sie unabhängig von den Gegebenheiten der Zielplattform originalgetreu überall dargestellt werden.
- Wenn eine Schrift weder eingebettet noch auf der Zielplattform installiert ist, dann verwendet Acrobat zu ihrer Simulation Serifen- oder serifenlose Multiple Master-Fonts verwendet, die bei jeder Acrobat-Installation enthalten sind.
- Da jede eingebettete Schrift Speicherplatz beansprucht, kann man zusätzlich verlangen, dass von einer Font nur die tatsächlich verwendeten Zeichen eingebettet werden.

Die Optionen im einzelnen:

- Alle Schriften einbetten:  
Sollte man, wenn Dokumente weiterverteilt werden immer aktivieren.
- Untergruppen aller eingebetteten Schriften nur unter xx%:  
Wenn Sie zum Beispiel einen Prozentsatz von 35 angeben und weniger als 35 % der Zeichen verwendet werden, bettet Distiller nur diese Zeichen ein. Mit xx%=100% erreichen Sie also, dass soweit wie möglich Untergruppen statt kompletter Fonts eingebettet werden.

Mit der Untergruppe „Einbettung“ kann man einstellen, dass bestimmte Schriften immer oder niemals eingebettet werden.

## **Farbe / ...**

Die Einstellungen dieser Parameter-Karte die Behandlung von Farb-Profilen durch Distiller und bestimmen, ob Farben bei der Verarbeitung von Dateien konvertiert werden sollen. Diese Einstellungen wird festgelegt, welche Farbbereiche zugrundegelegt werden, wenn die resultierenden PDF-Dateien zwischen verschiedenen Anwendungen und Ausgabegeräten ausgetauscht werden.

## **Erweitert / ...**

Die erweiterten Einstellungen steuern

- ob DSC-Kommentare in einer PDF-Datei beibehalten werden sollen,
- wie groß das Standardpapierformat ist,
- wie zusätzliche Angaben in der PostScript-Datei die Konvertierung beeinflussen.

Für Erstbenutzer dürften höchstens die Festlegung des Papierformats von Interesse sein.

## 2.2.6 Menüfunktionen: Distiller-Voreinstellungen / Sicherheit ...

Mit dieser Einstellung können Sie angeben, welche Sicherungen in die konvertierten Dokumente per Default eingefügt werden sollen.

Für einzelne Dokumente kann man die hier eingestellten Werte auch beim Abspeichern modifizieren (siehe 2.2.4). Mögliche Einstellungen betreffen:

Es kann ein Passwort zum Öffnen der Dateien vorgegeben werden.

Die folgenden Operationen können gesperrt werden:

- Drucken des Dokuments
- Verändern des Dokuments
- Auswählen von Texten/Grafiken zur Übernahme in das Clipboard und in andere Anwendungen.
- Einfügen und Ändern von Notizen und Formularfeldern in das Dokument.

Sind solche Funktionen gesperrt, dann kann man sie bei „Speichern unter ...“ verändern, sofern man das Passwort für die Veränderung kennt.

Wichtig: PDF-Dokumente sind nur mit Acrobat-Werkzeugen anzuzeigen, aber nicht mit Ghostview

## 2.2.7 Menüfunktionen: Distiller-Voreinstellungen / Hilfe

Acrobat-Handbuch:	Standard-Hilfe für interaktive Anwender Eine besser zum Drucken geeignete Version ist [4]
Distiller-Parameter-Handbuch:	Gibt an, wie man das Verhalten des Distiller mit Post-Script-Befehlen steuern kann. Für Normalanwender eher ohne Bedeutung.
pdfmark-Handbuch:	Beschreibt wie Anwendungsprogramme Einfluss auf die erzeugte PDF-Datei nehmen können. Für Normalanwender eher ohne Bedeutung.

## 2.3 PDFwriter

PDFwriter wird bei der Acrobat-Installation ins Drucker-Menü eingetragen und durch Drucken auf dem Drucker „PDFwriter“ wird eine PDF-Datei erzeugt.

Im Gegensatz zum Distiller, enthält der PDFwriter keinen PostScript-Interpreter, sondern benutzt direkt das Graphik-Interface des Betriebssystems (bei Windows das GDI-Interface). PDFwriter ist für den Bürobereich ein einfaches und ausreichendes Werkzeug, für anspruchsvollere Aufgaben fehlen ihm aber wesentliche Fähigkeiten:

- Bei EPS-Graphiken in Dokumenten werden nur die eingebettete Preview-Pixelgraphiken in die PDF-Datei übernommen. Diese sind von geringerer Qualität und wenn sie in der EPS-Datei fehlen, werden nur Platzhalter erzeugt.
- Es werden keine pdfmarks unterstützt. Damit können Links und andere in der Textquelle vorhandenen Zusatzinformationen nicht in die PDF-Datei übernommen werden.
- PDFwriter arbeitet mit festen Voreinstellungen für die Font-Einbettung und wenigen Optionen bzw. festen Einstellungen für die Downsampling und Kompression.

## 2.5 Acrobat

Acrobat ist das Programm, mit dem PDF-Dateien interaktiv bearbeitet werden können. Sie können dabei unter anderem

- Links, Lesezeichen und Thumbnails interaktiv eintragen und bearbeiten
- Seiten, ersetzen, einfügen löschen
- Seiten drehen, beschneiden,
- Filme, Audios und andere Komponenten in ein PDF-Dokument aufnehmen
- Texte scannen, Texterkennung durchführen und das Ergebnis als PDF-Datei ablegen.
- einzelne WEB-Seiten oder ganze WEB-Sites in einer PDF-Datei lokal speichern.
- Anmerkungen in ein PDF-Dokument eintragen
- Interaktive Formulare erstellen
- Textblöcke in einem PDF-Dokument ändern (In geringem Umfang)
- Plugins für Funktionserweiterungen einsetzen.

Eine vollständige Darstellung aller Funktionen würde den Rahmen dieses Textes sprengen. Zur Einarbeitung wird daher in der 2. Übung eine „Guided Tour“ angeboten, bei der Sie die Kernfunktionen von Acrobat kennenlernen. Verwenden Sie dabei die online-Hilfefunktion.

Eine systematische Beschreibung von Acrobat in Buchform finden Sie in [4].

In der Übungen 3 werden Sie in analoger Weise Funktionen kennenlernen, die über die Kernfunktionen hinausgehen, insbesondere das Erstellen von PDF-Formularen.

## 2.6 Catalog und Search

Acrobat und der Acrobat Reader bieten zwei unterschiedliche Funktionen zur Suche an:

### Die einfache Textsuche

Dabei handelt es sich um eine lineare Suchen nach einem Suchbegriff in der aktuell geöffneten PDF-Datei. Diese Form der Suche ist zwar immer möglich, aber bei großen Dateien langsam und wenn man in mehreren Dateien suchen will, muss man sie einzeln dazu öffnen.

### Die indizierte Suche

Damit ist das Suchen in einem ganzen Dokumentenbestand mit Hilfe eines automatisch generierten Index möglich. Das setzt voraus, dass die PDF-Dokumente in einer Directory-Hierarchie abgelegt sind und dass zuvor ein Volltext-Index erzeugt worden ist. Diese Suchtechnik ist wesentlich flexibler:

- Die Option „Wortstamm“ sucht nach Wörtern, deren Wortstamm mit dem des Suchbegriffs identisch ist.
- Durch die Option „Groß-/Kleinschreibung“ wird die Suche auf Entsprechungen mit der angegebenen Groß- und Kleinschreibung beschränkt.
- Mit der Option „Bedeutung“ werden auch synonyme Begriffe gesucht.
- Mit den Operatoren UND und ODER können Suchbegriffe logisch verknüpft werden.
- Die Option „Umgebung“ beschränkt die Ergebnisse bei einfachen Suchvorgängen mit UND auf ein Paar verknüpfter Wörter pro Dokument, nämlich auf das Paar, das am dichtesten beieinander liegt.
- Die englischsprachige Acrobat-Version besitzt die Option „Sounds like“, mit der nach Begriffen gesucht wird, die ähnlich wie der eingegebene Suchbegriff ausgesprochen werden.

Wichtig:

Die indizierte Suche ist für Dokumentenbestände, auf die über das WWW zugegriffen wird nur möglich, wenn auf dem Server zusätzliche dafür Software installiert ist

### Aufbau eines Index mit Catalog

Zum Indizieren eines Dokumentenbestandes dient die Anwendung Acrobat Catalog. Um zu einem einzelnen Dokument oder einem Dokumentenbestand einen Index zu generieren, geht man wie folgt vor:

1. Die PDF-Dokumente in ein Verzeichnis oder eine Verzeichnis-Hierarchie kopieren.
2. Catalog über das Startmenü aufrufen und „Index / New ...“ wählen.
3. Die Voreinstellungen für die Generierung setzen:
  - „Index Title“ und „Index Description“ sind beschreibende Informationen zu dem Index.
  - Mit „Include directories“ werden Verzeichnisse ausgewählt, die zu indizieren sind.
  - Mit „Exclude directories“ können Unterverzeichnisse davon ausgenommen werden.
  - Mit Options werden die Vorgaben für die Indexgenerierung gesetzt.
    - Es können einzelne Wörter ausgeschlossen werden (Artikel, Füllwörter, ... )
    - Es wird festgelegt, welche Suchoptionen vom Index unterstützt werden (ob Groß/Kleinschreibung unterschieden werden kann, Suche nach Wortstämmen möglich ist, usw. )
    - „Optimize for CD-ROM“ optimiert den Index für die Verwendung auf CD-ROM .
1. Mit dem Button Build auf der Parameterbox „Index / New ...“ wird der Index erzeugt. Dabei wird eine Datei mit der Erweiterung .pdx und eine Verzeichnis-Hierarchie erzeugt.

### **Hinweise für die Praxis**

1. Der erzeugte Index kann zusammen mit dem Dokumentenbestand an einen anderen Platz kopiert werden.
2. Wenn sich eines der indizierten Dokumente nach der Generierung des Index geändert hat, wird eine Warnung angezeigt. Der Index sollte in der Regel dann neu generiert werden mit der Menü-Funktion „Index / Build ...“
3. Mit Passwort geschützte Dokumente lassen sich nicht indizieren. Daher gehe man wie folgt vor:
  - Den Index mit dem noch ungeschützten Dokument generieren, dabei die Option „Optimize for CD-ROM“ wählen. Dadurch werden später Warnungen unterdrückt.
  - Den Passwortschutz im Dokument setzen und dieses wieder speichern.

## 2.7 Capture und Reviewer

Das Capture-Plugin für Acrobat ermöglicht es, Vorlagen einzuscannen und in eine PDF-Datei umzuwandeln. Wesentliche Merkmale sind dabei:

- Die gescannte Vorlage wird im einfachsten Fall als Bild in die PDF-Datei importiert. außer einer guten Komprimierung des Pixelbildes wird dabei nichts gegenüber der Speicherung in einem Bildformat (z.B. TIFF) gewonnen.
- Die gescannte Vorlage kann aber auch mit der Texterkennungskomponente in „Paper Capture“ erfasst werden. Dabei wird die gescannte Vorlage in PDF-Text umgewandelt. Ihr Textinhalt wird damit suchbar. Im Text enthaltene Bilder und nicht umwandelbare Textelemente bleiben als Graphik erhalten.
- Über eine Option kann man sogar einstellen, dass bei der Anzeige immer das gescannte Pixelbild gezeigt wird und der Text unsichtbar dahinter liegt.
- Mit der Reviewer-Komponente kann man Textstellen, die nicht sicher erkannt wurden, interaktiv korrigieren. Dabei erhält man für jede Problemstelle einen Textvorschlag, den man akzeptieren oder ändern kann.



### 3. PDF-Formulare

PDF-Formulare entsprechen vom Grundkonzept und von der Funktionalität her weitgehend den bekannten HTML-Formularen. Unterschiede ergeben sich vor allem in folgender Hinsicht:

- Das Layout lässt sich wesentlich flexibler gestalten
- Normale PDF-Formulare lassen sich ganz ohne technisches Know-How erstellen, mit Scripting-Techniken hat man allerdings zusätzliche Möglichkeiten .
- Der Aufwand zum Erstellen von PDF-Formularen ist wesentlich geringer als bei HTML-Formularen.

#### 3.1 Erstellen von PDF-Formularen

Das Entwickeln von PDF-Formularen ist ein zweistufiger Prozess:

1. Zunächst wird der Formulartext mit einem Textsystem erstellt und als PDF-Datei gespeichert . Alternativ kann man auch ein vorhandenes Papierformular scannen und in eine PDF-Datei importieren.
2. Danach werden in einem zweiten Schritt mit dem Formular-Tool von Acrobat die Formularfelder interaktiv erzeugt.

#### Nutzung von PDF-Formularen

PDF-Formulare können in unterschiedlicher Weise genutzt werden:

- Als lokal gespeicherte Dateien kann man sie mit Acrobat oder dem Acrobat Reader ausfüllen und ausdrucken oder als Datei versenden.
- Sie können von einem WEB-Server geladen und mit dem Browser ausgefüllt werden. Der Formularinhalt kann danach an den Server zurückgeschickt und dort ausgewertet oder weiterversandt werden.
- Der Server kann auch direkt Daten an das Formular zurücksenden, wo sie dargestellt werden. Im Unterschied dazu kann der Server bei HTML-Formularen nur neue Seiten aufbauen und als Quittung zurücksenden.

#### Feldtypen in PDF-Formularen

PDF-Formulare besitzen die gleichen Feldtypen wie HTML-Formulare:

- Textfelder
- Listboxen
- Comboboxen
- Pushbuttons

Die Eigenschaften dieser Felder werden mit Acrobat interaktiv über Parameterboxen festgelegt, wobei bezüglich Layout und Funktionalität mehr Möglichkeiten als bei HTML-Formularen zur Verfügung stehen.

#### Textfelder

Die wesentlichen Optionen, die das Layout und die Funktion von Textfeldern steuern, sind:

- Schrifttyp, Schriftgröße, Schriftfarbe
- Begrenzung der eingebbaren Zeichenzahl
- Einzeiliger/mehrzeiliger Text, Text-Ausrichtung
- Darstellung des eingegebenen Textes durch ein Ersatzzeichen (für Passworteingaben)

## Listboxen

Sie erlauben die Selektion eines oder mehrerer Elemente aus einer Liste. Für jedes Listenelement kann individuell festgelegt werden, welcher Wert exportiert wird.

## Comboboxen

- Sie erlauben ebenfalls die Selektion eines oder mehrerer Listenelemente, allerdings aus einer Pop up-Liste.
- Vorgegebene Werte können modifiziert werden.

## Checkboxen und Radio Buttons

- Sie können nur die Zustände ON und OFF annehmen
- Checkboxen können vom Anwender unabhängig voneinander ein/ausgeschaltet werden.
- Radio Buttons sind eine Gruppe von Checkboxen, von denen aber immer nur eine eingeschaltet werden kann. Die Gruppierung von Checkboxen zu einer Gruppe von Radio Buttons erfolgt dadurch, dass alle Elemente der Gruppe den gleichen Namen erhalten.

## Schaltflächen (Push Buttons)

Sie werden normalerweise zum Auslösen von Aktionen durch Ereignisse (z.B. Mausklick) verwendet. Schaltflächen können eine Beschriftung tragen, aber auch eine Graphik darstellen. Sie lassen sich daher auch zum Importieren von Graphiken, z.B. eines Logo in eine PDF-Datei verwenden. Der Button bleibt dann funktionslos.

## Gegenüberstellung von HTML- und PDF-Formularen

Über die von HTML-Formularen her bekannten Möglichkeiten hinaus besitzen PDF-Formulare die folgenden Features:

- umfangreichere Gestaltungsmöglichkeiten
- Es gibt keine Einschränkungen beim Drucken von Feldern
- Der WEB-Server kann Daten direkt an ein Formular zur Darstellung senden.
- Eine wesentlich umfangreichere Palette von Aktionen für Push Buttons.

## Hinweis zum Duplizieren von Feldern

Mit der Funktion „Edit/Fields/Duplicate“ kann man Felder duplizieren und Duplikate sogar auf unterschiedlichen Seiten des Formulars ablegen. Wenn ein Anwender den Inhalt eines Feldes verändert, ändern sich die Inhalte der Duplikate in gleicher Weise. In bestimmten Fällen ist dieser Effekt nützlich, er führt aber oft zu nicht beabsichtigten Seiteneffekten..

## Export von Formulardaten

Die Feldinhalte von Formulardaten können mit der Acrobat-Funktion „Datei/exportieren...“ aus einem Formular extrahiert werden. Es entsteht dabei eine Datei mit der Datei-Erweiterung **.FDF**. Diese enthält die exportierten Daten in einem Format, das gegenüber PDF stark vereinfacht ist.

Im nächsten Abschnitt betrachten wir wie die Formulardaten an einen Server versandt werden und dort weiterverarbeitet werden können.

## 3.2 Serverseitige Maßnahmen

### Übermittlung der Formulardaten an den Server

Die Inhalte eines Formulars können von einem Client zum Server übertragen und dort weiterverarbeitet werden. Im Formular selbst wird dazu eine Schaltfläche angelegt und für diese festgelegt:

- eine Aktion „Formular senden“ , z.B. bei Mausklick.
- das Ziel, an das die Formulardaten zu senden sind ist die URL eines CGI-Skripts auf dem Server, das die Verarbeitung übernimmt.
- Als Exportformat kann man das HTML- oder das FDF-Format wählen

### Maßnahmen auf dem Server

Auf dem Server muß ein CGI-Skript (CGI = Common Gateway Interface) vorhanden sein, das die Formulardaten in Empfang nimmt, auswertet und weiterverarbeitet. Aus Sicherheitsgründen darf nur der Systemverwalter solche CGI-Skripts auf dem Server installieren.

Es können auch Daten vom Server zu einem Formular im FDF-Format zurückgeschickt und dort wieder dargestellt werden. (Bei HTML-Formularen ist ein Transport in dieser Richtung nicht möglich.)

In der 3. Übung untersuchen wir dies an einem Beispielformular und seiner Weiterverarbeitung genauer.

## 4. Bilder in elektronischen Dokumenten

In diesem Kapitel betrachten wir die grundlegenden Konzepte und Begriffe, die für das Erstellen von Graphiken für elektronische und gedruckte Dokumente wichtig sind. Im Praktikumsteil lernen Sie ein professionelles Werkzeug zum Erstellen von Graphiken kennen, nämlich Photoshop.

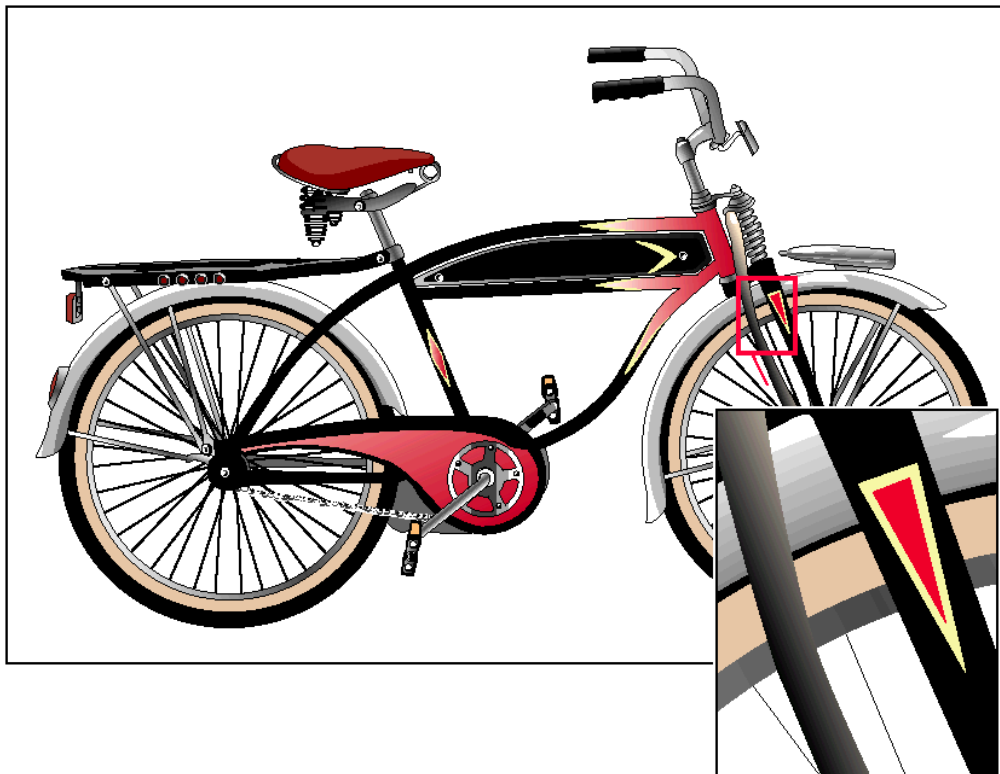
### 4.1 Vektorbilder und Rasterbilder

Die beiden Haupttypen von Graphiken sind Vektorgraphiken und Rastergraphiken. Aus der unterschiedlichen Art und Weise, wie diese im Rechner gespeichert werden ergeben sich zahlreiche Konsequenzen für ihre Einsatzbereiche, ihre Verarbeitung, ihren Platzbedarf usw.

#### Vektorgraphiken

beschreiben Objekte durch ihre Randlinien, die sich aus Vektoren und Kurven zusammensetzen. Sie eignen sich daher besonders gut für technische Zeichnungen, Bilder deren Objekte klar umrissene Begrenzungen besitzen, Logos usw. Ihr Platzbedarf ist in vielen Fällen geringer als der vergleichbarer Rasterbilder. Sie lassen sich ohne Informationsverlust in vielfältiger Weise transformieren: vergrößern, verkleinern, verschieben, verzerren, animieren. Ein Kreis läßt sich z.B. durch die Position seines Mittelpunkts und seines Radius beschreiben. Erst für die Darstellung auf einem Bildschirm oder zum Ausdruck wird der Kreis in eine Pixeldarstellung umgewandelt.

Bekannte Werkzeuge zum Erstellen von Vektorgraphiken sind. z.B. Adobe Illustrator, Corel Draw, Micrografx Designer.



## Rasterbilder

bestehen aus einer Matrix von Bildpunkten (Pixeln). Sie eignen sich gut für photorealistische Bilder, bei denen es eine Vielzahl von Farbabstufungen und keine scharf begrenzten Objekte gibt. Bei der Bearbeitung verändert man Gruppen von Pixeln, deren Zuordnung zu bestimmten Objekten aber in den Bilddaten selbst nicht enthalten ist. Techniken zur Auswahl von Bildbereichen spielen daher hier eine große Rolle. Bei Transformationen können Bildinformationen unwiederbringlich verloren gehen. Besonders wichtig ist die Auflösung, mit der Rasterbilder erstellt und verarbeitet werden. Wenn sie zu gering ist, sind Objektränder bei vergrößerter Darstellung am Bildschirm und vor allem beim Ausdruck treppenförmig.

Bekannte Werkzeuge zum Erstellen von Rastergraphiken sind. z.B. Adobe Photoshop, Micrografx Picture Publisher, Paintshop Pro.



## 4.2 Auflösung und Farbtiefe

Für das Erstellen und die Verarbeitung von Rasterbildern ist ein Verständnis der Auflösung besonders wichtig. Eine zu geringe Auflösung kann ein Bild unbrauchbar machen für den Ausdruck, eine zu hohe kann die Übertragungszeit von WEB-Seiten extrem erhöhen. Ein Bild, das für die Darstellung auf einer WEB-Seite nur wenige Kilobyte gross sein muss, kann für den Druck in einem Prospekt viele Megabyte groß sein. Im folgenden werden daher unterschiedliche Arten der Auflösung unterschieden:

### Die Bild-Auflösung

bezeichnet den Abstand der Pixel im Bild. Sie wird in Pixel pro Inch (ppi) gemessen. Ein quadratisches Bild mit einer Auflösung von 300 ppi und 10 cm Kantenlänge besitzt somit

$$\left( \frac{10}{2,54} * 300 \right)^2 = 1.395.002 \text{ Pixel.}$$

Ein gleich grosses Bild mit eine Auflösung von 72 ppi besitzt nur

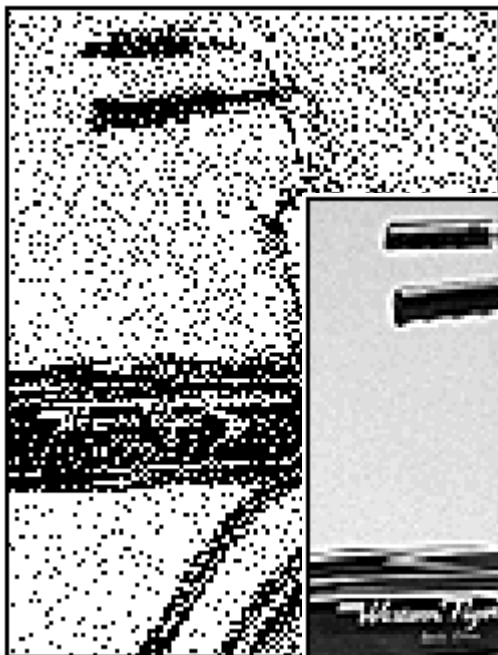
$$\left( \frac{10}{2,54} * 72 \right)^2 = 80.352 \text{ Pixel}$$

Die Bildauflösung wird bei der Erstellung des Bildes, z.B. in Photoshop eingestellt. Sie ändert sich nicht, wenn das Bild in ein Textdokument importiert wird. Auch die mögliche Auflösung beim Ausdruck wird damit festgelegt.

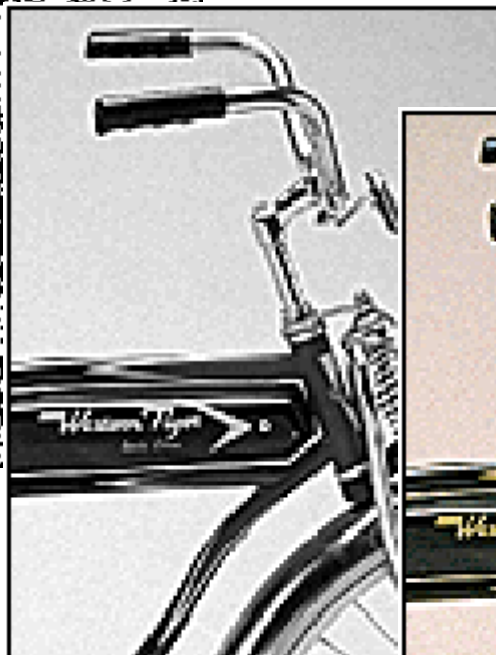
### Die Farbtiefe (Pixelauflösung, Bit-Tiefe) von Bildern

gibt die Anzahl von Bits an, die für den Farbwert eines jeden Pixel zur Verfügung stehen. typische Werte für die Farbtiefe von Bildern sind:

- 1 Bit für Binärbilder, die nur 2 Farben enthalten, z.B. schwarz und weiß
- 8 Bit für Palettenbilder, die bis zu 256 verschiedene, beliebige Farbtöne enthalten können und für Graustufenbilder mit bis zu 256 Graustufen.
- 24 Bit für RGB-Bilder, die 3 Farbkanäle (rot, grün, blau) mit je 8 Bit darstellen können. Damit sind 16,7 Mio verschiedene Farben in einem Bild darstellbar, mehr als unser Auge unterscheiden kann.



Binärbild  
Farbtiefe 1 Bit



Graustufenbild  
Farbtiefe 8 Bit



RGB-Bild  
Farbtiefe 24 Bit

### Bildschirm-Auflösung

ist die Dichte der Monitor-Bildpunkte. Sie wird in dots per inch (dpi) oder pixels per inch (ppi) angegeben. Die Auflösung von Röhren-Monitoren beträgt heute 72 oder 96 dpi, bei LCD-Monitoren sind heute 85 lpi Standard und 123 lpi technisch schon möglich.

Die Pixel-Auflösung eines Bildes und eines Monitors wirken zusammen, wenn ein Bild ohne Vergrößerung/Verkleinerung durch die Software am Bildschirm dargestellt wird: Ein Bild mit einer Bildauflösung von 144 ppi wird auf einem Bildschirm mit einer Bildschirmauflösung von 72 dpi in doppelter Größe dargestellt.

## **Rasterfrequenz**

Bei vielen Druckverfahren, werden Halbtöne dadurch erzeugt, dass man Pixel zu Halbtönen zusammenfasst. Sollen z.B. auf einem S/W-Laserdrucker, der eine Auflösung von 600 dpi hat, Halbtöne gedruckt werden, kann man Zellen mit je 8x8 Druckerpixeln zusammenfassen und so 256 verschiedene Graustufen in einer Halbtonzelle darstellen. Die Anordnung der schwarzen und weißen Pixel in der Halbtonzelle wird durch das verwendete Ditherverfahren festgelegt.

Der lineare Abstand der Halbtonzellen ist damit aber nur noch ein Achtel so groß wie die Pixelauflösung des Druckers. Man bezeichnet den Abstand der Halbtonzellen als Linienauflösung oder Rasterauflösung (lpi, lines per inch). Im vorstehend beschriebenen Beispiel ist damit die Linienauflösung 75 lpi.

## **Ausgabe-Auflösung**

Die Ausgabe-Auflösung bezeichnet die technische Auflösung des Ausgabegeräts und wird in dots per inch (dpi) gemessen. Gute Laserdrucker besitzen eine Auflösung von 600 – 1200 dpi, die Auflösung von Lichtsatzgeräten liegt bei 1200 oder 2400 dpi. Für den S/W-Druck ist i.a. eine höhere Auflösung erforderlich als für den Druck von Farbbildern. Die Auflösung von Bildern und Texten, die für den Druck vorgesehen sind, sollte sich an der Auflösung der Druckverfahren orientieren.

## **Bild-Auflösung und Bildgröße**

Mit der Bildgröße bezeichnet man die physikalischen Abmessungen des Bildes bei der Darstellung. Da die Auflösung eines Bildes unabhängig von einer speziellen Darstellung feststeht, bewirkt eine Erhöhung der Bildgröße eine Verringerung der Auflösung des dargestellten Bildes.

Beispiel: Ein Bild das in Originalgröße mit 144 dpi gedruckt wird, kann bei einer linearen Vergrößerung um den Faktor 2 nur noch mit einer Auflösung von 72 dpi dargestellt werden. Der Qualitätsverlust fällt beim Druck stärker auf, da wir dort im Gegensatz zum Bildschirm hohe Auflösungen gewohnt sind.

Werkzeugen wie Photoshop können

- die Bildauflösung nachträglich erhöhen bei gleichbleibender Bildgröße oder
- die Bildgröße nachträglich erhöhen bei gleichbleibender Auflösung.

In beiden Fällen müssen dazu zusätzliche Pixel durch Interpolationsverfahren berechnet werden. Das Ergebnis ist in jedem Fall eine Verminderung der Bildqualität: Die Bilder wirken dadurch verschwommener. Eine Verringerung der Auflösung bei gleichbleibender Bildgröße hat solange keine negativen Auswirkungen, wie die Auflösung des Ausgabegerätes nicht unterschritten wird.

## **Wichtig:**

Bei der Erstellung, beim Einscannen und beim Bearbeiten von Bildern sollte mit hoher Auflösung und hoher Farbtiefe gearbeitet werden, eine Reduktion darf erst im letzten Schritt unmittelbar vor der Ausgabe erfolgen, wenn dies notwendig sein sollte.

## **Bild-Auflösung und Bild-Dateigröße**

Die Dateigröße eines Bildes ist proportional zum Quadrat seiner Auflösung und proportional zu seiner Farbtiefe, solange keine komprimierenden Bildformate verwendet werden.

Beispiel:

- Ein 10x10 cm großes RGB-Bild mit einer Auflösung von 72 dpi belegt ca 230 K Bytes
- Ein 10x10 cm großes RGB-Bild mit einer Auflösung von 600 dpi belegt ca 16 M Bytes
- Ein 10x10 cm großes Graustufenbild-Bild mit 256 Graustufen und einer Auflösung von 600 dpi belegt ca 5,3 M Bytes

## 4.3 Farbmodelle

### 4.3.1 Grundlegende Begriffe

Farbe ist kein rein physikalisches Phänomen, sondern entsteht durch das Zusammenwirken dreier Faktoren:

#### Licht

Es muß elektromagnetische Strahlung eines relativ eng begrenzten Spektralbereichs vorhanden sein. Sie wird charakterisiert durch ihre Wellenlänge, die zwischen 400 und 700 nm liegen muß und durch ihre Intensität.

#### Objekte

Die Strahlung muß auf ein Objekt treffen, das einen Teil der Strahlung absorbiert, einen anderen Teil reflektiert. Das einfallende Licht bewirkt demnach zusammen mit einem Objekt, auf das es trifft, einen Farbreiz.

#### Beobachter

Der Farbreiz wird vom visuellen System eines Beobachters aufgenommen und verarbeitet. Das Ergebnis dieses Verarbeitungsprozesses empfinden wir als Farbe. Schematisch können wir also sagen:

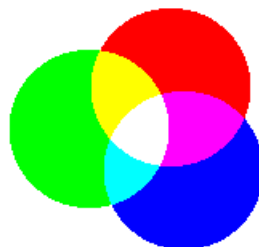
$$\begin{aligned}\text{Licht} + \text{Objekt} &= \text{Farbreiz} \\ \text{Farbreiz} + \text{Beobachter} &= \text{Farbe}\end{aligned}$$

Zur Beschreibung von Farben werden mathematisch-physikalische Modelle benutzt, die sich entweder an technischen Aspekten (RGB-, CMYK-Modelle) oder eher an physiologischen Aspekten (HIS-Modell) orientieren. Ein Verständnis der Farbwahrnehmung und der gebräuchlichen Farbmodelle ist hilfreich für den Umgang mit Farbe in elektronischen und gedruckten Dokumenten.

#### Additive Farbmodelle

Bei diesen Farbmodellen tragen die einzelnen Grundfarben additiv zum Gesamteindruck einer Farbe bei. Werden von einem Farbreiz alle Rezeptoren gleich stark angeregt, dann interpretieren wir dies als den Farbton Grau bzw. Weiß. Werden die Zapfentypen verschieden stark angeregt, dann entstehen Sekundärfarben. Die folgende Tabelle zeigt, wie durch additive Mischung der drei Primärfarben die acht Grundfarben entstehen.

Additive Farbmischung liegt z.B. bei der Ausgabe auf Bildschirmen vor: In jedem Punkt des Bildschirms können ein roter, ein blauer und ein grüner Phosphorpunkt unabhängig voneinander angesteuert werden. Ihre Strahlungssumme ergibt so den Farbreiz für diesen Bildpunkt.



#### Subtraktive Farbmodelle

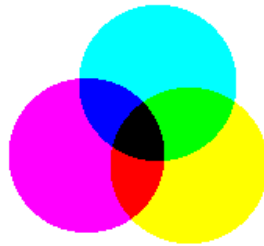
Beim Zusammentreffen mit Materie werden Lichtwellen teilweise absorbiert und erreichen somit den Beobachter nicht. Nur der reflektierte Anteil des Lichtes verursacht beim Beobachter einen Farbreiz. Ein Objekt wirkt weiß, wenn es (fast) kein Licht absorbiert und schwarz, wenn es das einfallende Licht (fast) vollständig absorbiert.

Bei Farbdruckern werden verschiedene Farbpigmente übereinander auf das Papier aufgebracht. Die oberliegenden Farbpigmente wirken daher als Filter für die darunterliegenden. Ein Rotfilter absorbiert die grünen und blauen Anteile des Spektrums, also z.B. Cyan. Ein



Bildpunkt, in dem ein roter Farbpunkt über einem mit der Farbe Cyan liegt, wirkt daher auf den Beobachter schwarz.

Farbdrucker verwenden die Primärfarbstoffe Cyan, Magenta und Gelb und stellen daraus durch subtraktive Mischung die acht Grundfarben her. Zusätzlich verwendet man noch Schwarz als vierte Primärfarbe, u.a. deshalb, weil durch subtraktive Mischung keine perfekten schwarzen Farbtöne zu erzielen sind.

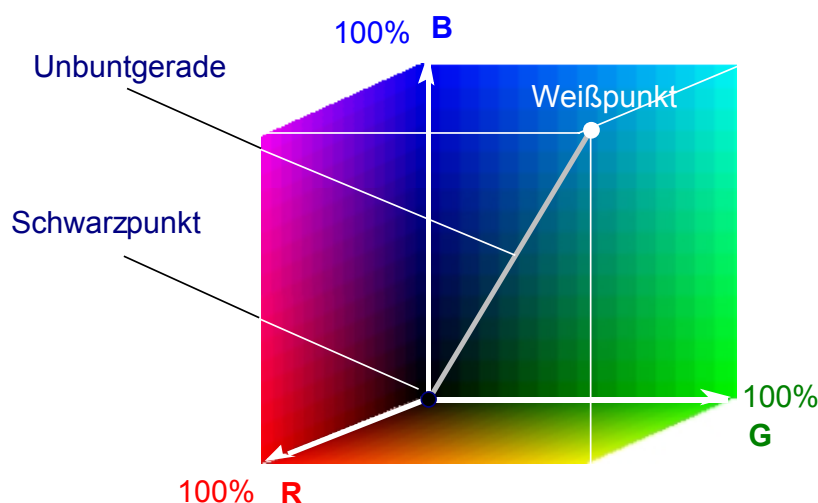


#### 4.3.2 Das RGB-Modell

Der RGB-Farbraum betrachtet eine Farbe als eine additive Mischung der Primärfarben Rot, Grün und Blau. Dies entspricht z.B. den durch die Monitortechnik gegebenen technischen Randbedingungen: Die Farbe eines Bildschirm-Pixel wird dort mit einem roten, einem grünen und einem blauen Phosphorpunkt realisiert. Auch viele Bildaufnahmegeräte liefern ein RGB-Signal.

Die folgende Abbildung zeigt ein Würfel-Modell des RGB-Farbenraumes. Jede mögliche Farbe ist eine Konvexkombination der Vektoren  $\vec{R}$ ,  $\vec{G}$  und  $\vec{B}$ . Auf der Diagonalen des Würfels liegen die Farbtöne mit gleich großen Rot-, Grün- und Blau-Anteilen, d.h. die Grautöne. Sie wird deshalb auch als Unbunt-Gerade bezeichnet.

Wenn jede der drei Primärfarben mit einer Auflösung von 256 Werten dargestellt werden kann, dann erhalten wir  $256^3 = 16,7$  Mio. verschiedene Farbtöne.



### 4.3.2 Das CMY-Farbmodell

In der Drucktechnik spielen subtraktive Farbmodelle eine wichtige Rolle. Das am häufigsten anzutreffende Modell basiert auf den Primärfarben Cyan, Magenta und Gelb (Yellow). Das CMYK-Modell ist ein subtraktives Farbmodell

Als zusätzliche Primärfarbe wird meist noch Schwarz verwendet, denn:

- Die Überlagerung der drei Primärfarben ergibt in der Praxis kein Tiefschwarz und
- bei Mischfarben kann man den Grau-Anteil subtrahieren und mit schwarzer Farbe drucken, wodurch der Farbverbrauch reduziert wird.

Die Transformation zwischen dem RGB- und dem CMY-System ist sehr einfach:

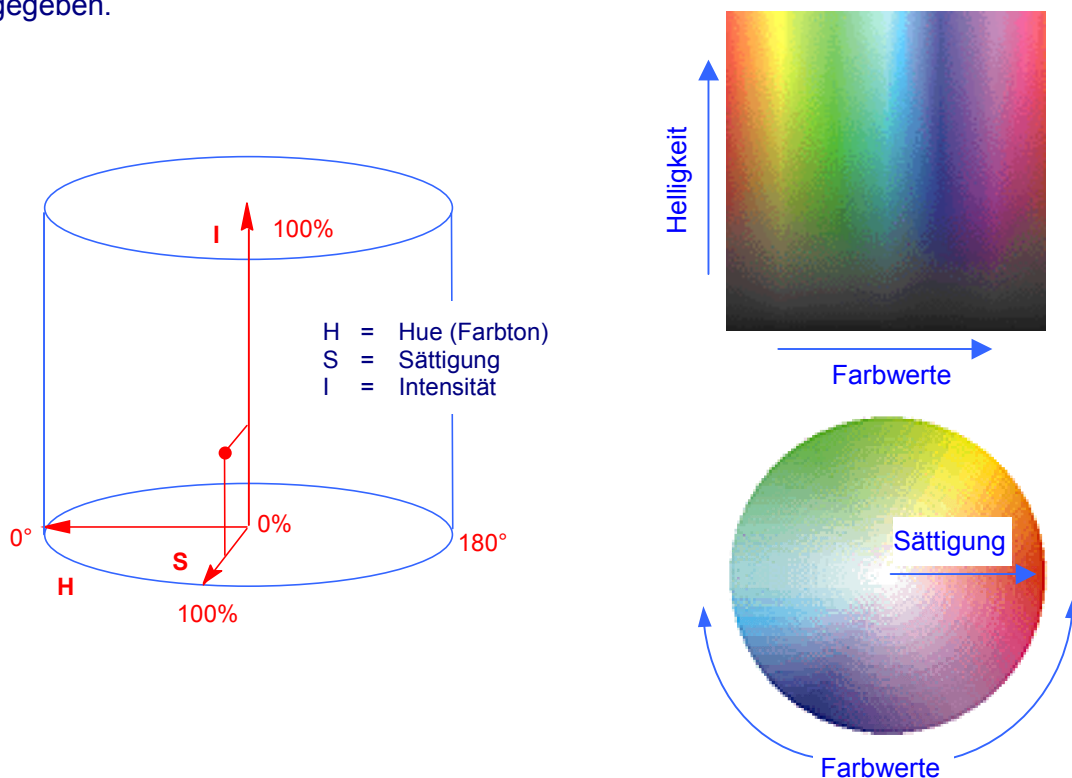
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} S \\ S \\ S \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad \text{und} \quad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} W \\ W \\ W \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

wobei S und W die Maximalwerte für Weiß im CMY- bzw. im RGB-System darstellen. Ein Würfel-Modell des CMY-Farbenraums läßt sich analog zum Würfelmodell des RGB-Farbraums zeichnen. Der Nullpunkt liegt aber im Weißpunkt des RGB-Systems.

### 4.3.3 Das HSI-Modell

Das HSI-Modell basiert auf der physiologischen Wahrnehmung von Farben. Zur Charakterisierung von Farben verwendet das HSI-Modell drei Parameter:

- Der Farbwert (Hue) charakterisiert die die Wellenlänge des Lichts. Er wird entweder mit Farbnamen, rot, orange, gelb, ... bezeichnet oder durch einen Winkel zwischen 0° und 360° auf einem Standard-Farbkreis.
- Die Sättigung (Saturation, Chroma) bezeichnet die Reinheit einer Farbe. Damit wird der Grau-Anteil am Farbwert bezeichnet. Sie wird durch einen Prozentsatz zwischen 0% (reines Grau) und 100% (reine Farbe) angegeben.
- Die Helligkeit (Intensity) wird durch einen Wert zwischen 0% (schwarz) und 100% (weiß) angegeben.



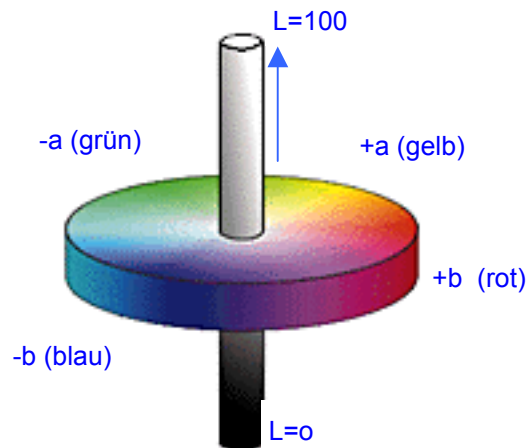
#### 4.3.4 CIE-Modelle

Um den Farbeindruck eines Beobachters unabhängig von individuellen Besonderheiten und einer bestimmten Gerätetechnik zu beschreiben, wurde von der Commission International de l'Eclairage, CIE, ein Modell definiert, das einen Farbeindruck durch drei unabhängige Farbvalenzen beschreibt und als CIE -  $L^*a^*b$  - Modell bezeichnet wird.

Die Bedeutung der CIE-Normvalenzen liegt darin, daß sie eine objektive Möglichkeit zur Kennzeichnung von Farben darstellen. Insofern kommt ihnen z.B. bei der Kalibrierung von Bildschirmen und Farbdruckern durch Farbmanagementsysteme eine große Bedeutung zu. Sie entsprechen aber weder unseren intuitiven Vorstellungen von Farben noch besteht ein direkter Zusammenhang zu technischen Farbaufnahme- und Wiedergabegeräten.

Das  $L^*a^*b$ -Farbmodell charakterisiert eine Farbe durch drei Zahlenwerte:

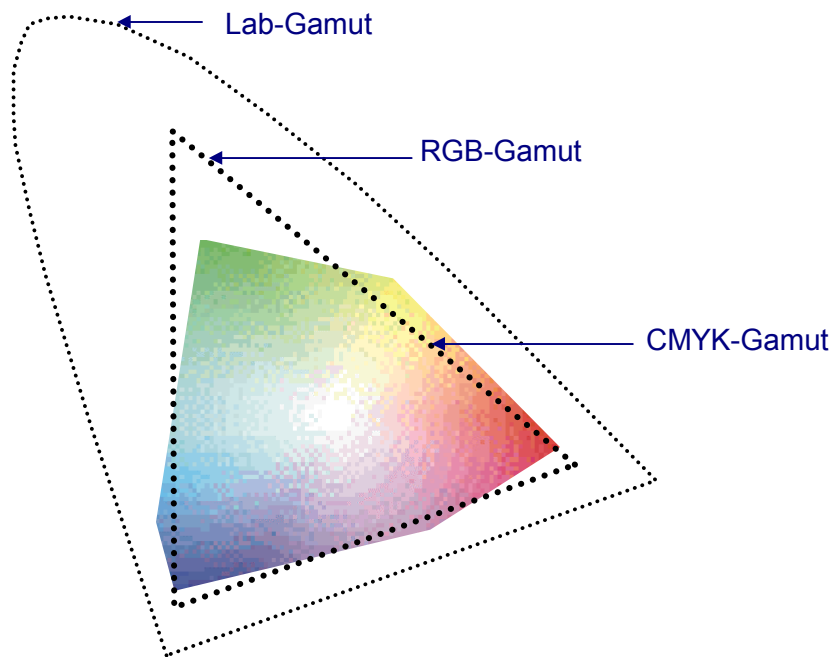
- die Luminanz (L) gibt eine Helligkeit an der Chroma-Wert a variiert im Farbbereich von grün bis rot
- der Chroma-Wert b variiert im Farbbereich von blau bis gelb:



#### 4.3.5 Farbbereiche (Gamuts)

Der Gamut eines Farbmodells ist der Bereich aller in ihm darstellbaren Farben. Das sichtbare Spektrum aller sichtbaren Wellenlängen besitzt die größte Farb-Variationsbreite. Dabei können unterschiedliche Mischungen verschiedener Wellenlängen den selben Farbeindruck hervorrufen. Unter den Farbmodellen besitzt das Lab-Modell den umfangreichsten Gamut und umfasst die Gamuts der RGB- und CMYK-Farbmodelle.

Die Abbildung auf der nächsten Seite zeigt auch, dass das CMYK-Modell, welches die druckbaren Farben beschreibt, den kleinsten Gamut besitzt. Andererseits sind im RGB-Modell, das die am Monitor darstellbaren Farben beschreibt, nicht alle druckbaren Farben darstellbar, z.B. reines Cyan und reines Gelb.



Gamuts unterschiedlicher Farbmodelle

#### 4.3.6 Arbeiten mit Farbmodellen

Beim Erstellen von Grafiken und Dokumenten spielen unterschiedliche Farbmodelle eine Rolle. Im folgenden betrachten wir ihre Einsatzbereiche:

- der Graustufenmodus  
besitzt in der Regel eine Skala von 256 Helligkeitsstufen, wobei 0=schwarz und 255=weiß entspricht. Es kann auch der Grad der Überdeckung mit schwarzer Tinte angegeben werden: 0% = weiß und 100% = schwarz.
- der RGB-Modus  
ist der am häufigsten benutzte Modus. Auch wenn Bilder in anderen Farbmodellen bearbeitet werden, müssen sie zur Darstellung am Bildschirm ins RGB-Modell transformiert werden. Die RGB-Werte variieren normalerweise im Bereich zwischen 0 (Minimum) und 255 (Maximum) für jeden der drei Kanäle.
- der CMYK-Modus  
Bevor ein Bild oder Dokument farbig gedruckt werden kann, muss es ins CMYK-Format transformiert werden. Zur Bearbeitung wird trotzdem der RGB-Modus empfohlen. Jeder Farbwert wird im CMYK-Modell durch die prozentualen Anteile der Prozeßfarben angegeben. Kleine Werte entsprechen dabei den hellen Farbtönen, der Wert (0,0,0) entspricht z.B. reinem Weiß.
- der Lab-Modus  
Dieser Modus wird zur Bearbeitung von Photo-CD-Bildern benutzt, aber auch wenn es auf hohe Farbtreue ankommt, z.B. beim von Bildern zwischen verschiedenen Systemen oder beim Ausdruck auf PostScript-Druckern. Die Helligkeitswerte liegen im Lab-Modell auf einer Skala von 0 bis 100, die die Farbwerte a und b variieren im Bereich von -120 bis +120.

# Teil 2

## 5. Gestaltung elektronischer Dokumente

Bei der Gestaltung elektronischer Dokumente sind einerseits ähnliche Regeln zu beachten wie bei Printprodukten: Eine klare inhaltliche Struktur, präzise Formulierungen, die Lesbarkeit unterstützende Textformatierungen, eine klare und übersichtliche Seitengestaltung und technisch tadellos komponierte Bilder sind wesentliche Faktoren eines gut gestalteten Dokuments.

Andererseits gelten für elektronische Dokumente eigene, durch die besonderen Eigenschaften der elektronischen Medien begründete Gestaltungsrichtlinien. So gibt schon beim Seitenformat und der Seitengröße eines Monitors und eines Buchs große Unterschiede, ebenso in der Auflösung und den Möglichkeiten zur Farbdarstellung. Hinzu kommen bei elektronischen Dokumenten völlig neue Aspekte z.B.

- Links von einer Stelle eines Dokuments zu anderen Stellen innerhalb desselben Dokuments oder zu anderen Dokumenten irgendwo auf der Welt,
- die Integration anderer Medien in ein Dokument (Audio, Video, Anwendungen,
- die Zeit zum Übertragen und der Platz zum Speichern eines Dokuments

Das folgende Kapitel gibt Hinweise zur Gestaltung elektronischer Dokumente.

### 5.1 Seitengröße und Seiten-Format

Während Druckseiten in der Regel höher als breit sind, ist dies bei Monitoren genau umgekehrt. Eine Textseite, die zum Ausdrucken vorgesehen ist, passt daher in der Regel nicht auf eine Bildschirmseite. Sie ist daher nicht vollständig sichtbar und zwingt beim Lesen zum Scrollen. Bei der Gestaltung von Seiten für die Darstellung am Bildschirm sollte man daher die folgenden Regeln beachten:

- Wählen Sie Seitenformat und Seitengröße so, dass ein Scrollen möglichst ganz überflüssig ist. Wenn das nicht realisierbar ist, sollte das Scrollen sich auf eine einzige Richtung beschränken, so dass man z.B. stets nur vertikal scrollen muss.
- Legen Sie die Seitengröße mit Blick auf das Bildschirmformat der Leser fest: Wenn eine Seite auf Ihrem 19“-Monitor noch gut lesbar ist, werden die Leser auf einem 15“-Monitor u.U. kleine Details kaum mehr erkennen können.
- Berücksichtigen Sie, dass die Viewer und Browser, mit denen die Leser eine Seite betrachten, einen Teil der Monitorfläche für Menüleisten, Navigationsbuttons usw. verbrauchen.
- Testen Sie ein Dokument mit Bildschirmen und Auflösungen, wie sie die Leser haben. Diese sind in der Regel kleiner als Ihre eigenen.

Falls eine Publikation sowohl am Bildschirm dargestellt als auch gedruckt werden soll, sind weitere Regeln zu beachten:

- Vermeiden Sie mehrspaltigen Text, damit die Leser beim Lesen nicht vom unteren Seitenrand zum Seitenanfang zurückscrollen müssen, wenn sie die nächste Spalte lesen wollen.
- Achten Sie darauf, dass die Zeilen nicht zu lang sind, damit die Leser nicht zum horizontalen Scrollen gezwungen werden.
- Verwenden Sie Querformat für die Seiten. Einerseits sind Bildschirme breiter als hoch, andererseits kann der Drucker das Papier im Querformat bedrucken oder je zwei Seiten auf ein Blatt im Hochformat drucken.

## 5.2 Lesbarkeit von Texten am Bildschirm

Die Auflösung handelsüblicher Monitore beträgt 72 oder 96 dpi und ist damit viel geringer als die heutiger Drucker, auch der einfachsten. Texte, die auf Papier gut lesbar, sind können daher am Bildschirm oft nur mit großer Anstrengung gelesen werden, insbesondere, wenn sie lang sind. Überprüfen Sie deshalb stets Ihre Dokumente mit den gleichen Monitortypen und den gleichen Viewern, wie sie auch die Leser verwenden. Die folgenden Hinweise sollten Sie im Hinblick auf eine gute Lesbarkeit der Dokumente beachten:

### Schrifttypen (Fonts)

Bei den Schrifttypen unterscheidet man Serifenschriften und serifenlose Schriften. Gedruckte Texte mit Serifenschriften sind in der Regel besser lesbar, weil Serifen das visuelle Erfassen eines Textes erleichtern. Die geringe Auflösung von Monitoren bewirkt jedoch bei Serifenschriften eine Qualitätsminderung, so dass bei normaler Schriftgröße die Lesbarkeit schlechter wird.

Verwenden Sie daher für Texte, die vorwiegend am Bildschirm gelesen werden sollen serifenlose Schriften deren Zeichen mit einfachen, klaren Linien dargestellt werden.

Regensburg

Serifenschrift (Times, 14 pt )

Regensburg

serifenlose Schrift (Helvetica Narrow, 14 pt)

### Schriftgröße

Zum Lesen am Monitor sollte die Schrift größer sein als für den Druck. Verwenden Sie für normalen Fließtext eine Schriftgröße von mindestens 12 Punkten. Wenn Sie verschiedene Schriftgrößen benutzen, dann sollten die Unterschiede deutlich erkennbar sein.

### Zeilenabstand (Durchschuss)

Der Zeilenabstand ist der Abstand zwischen den Ober- und Unterlängen der Textzeilen. Er sollte für Bildschirm-Texte größer als für gedruckte Texte sein. Empfohlen werden mindestens 20% bis 30% des Schriftgrades. So ist bei einer Schriftgröße von 12 Punkten ein Durchschuss von mindestens 4 Punkte zu wählen, so dass sich eine Gesamthöhe einer Zeile 16 Punkten ergibt.

### Hintergrund

Die Lesbarkeit eines Textes wird durch unruhige, strukturierte Hintergründe drastisch reduziert. Verwenden Sie daher hinter Texten entweder nur einfarbige oder wenig strukturierte Hintergründe die nicht vom Text ablenken. Auch der Helligkeits- und Farbkontrast sollte möglichst groß sein. Hintergründe mit einem Farbverlauf bergen daher die Gefahr, dass der Kontrast zur Schriftfarbe entweder in den hellen oder in den dunklen Bereichen des Verlaufs zu gering wird.

### 5.3 Methoden zur Erhaltung der Schriftqualität

Beim Austausch von Textdokumenten hängt die originalgetreue Darstellung auf der Zielplattform sehr davon ab, wie sie die im Dokument verwendeten Schriften darstellt. Wenn die Originalschriften auf der Zielplattform nicht zur Verfügung stehen und auch nicht gut simuliert werden können, kann sich das Erscheinungsbild eines Dokuments stark vom Original unterscheiden. Es sind im wesentlichen die folgenden Alternativen zu unterscheiden:

- Die Originalfont steht auch auf der Zielplattform zur Verfügung:  
In diesem Fall ist eine originalgetreue Darstellung möglich. Diese Voraussetzung ist aber nur für wenige Schriften zu erwarten: Für Windows-Rechner gehören z.B. die Schriften Courier New, Arial und Times Roman zur Standard-Ausstattung.

## The Fox and the Crow

Originalfont Handel Gothic BT, 28 Punkte

- Die Originalfonts werden ins Dokument eingebunden:  
In diesem Fall ist eine originalgetreue Darstellung auf der Zielplattform auch dann möglich, wenn die Originalfont dort nicht installiert ist. Der Preis dafür ist eine Vergrößerung des Dokuments um ca 30 kB pro Font.  
  
PDF-Dateien können eingebundene Fonts enthalten. Um Platz bzw. Übertragungskapazität zu sparen, kann man auch Teilmengen einer Font einbinden, wenn nur wenige Zeichen von Ihr im Dokument vorkommen. Fonts können bei der Generierung einer PDF-Datei mit dem Distiller und mit dem PDFwriter eingebunden werden.
- Simulation mit Multiple Masterfonts für PDF-Dateien  
Falls weder eingebundene Fonts noch Ersatzfonts zur Verfügung stehen, können Acrobat und der Acrobat Reader die Originalfonts simulieren. Jede Acrobat-Installation enthält dazu „Multiple Masterfonts“ (eine Serifenfont und eine serifenlose Font). Diese lassen sich an die stets mit einem Dokument gelieferte Fontmetrik exakt anpassen. Die einzelnen Zeichen können dabei etwas anders aussehen als die Originalzeichen, aber Ihre Abmessungen sind identisch, so das Die Zeilen- und Seitenstruktur des Dokuments erhalten bleibt.

## The Fox and the Crow

Simulation mit Adobe Sans MM

- Auf der Zielplattform gibt es eine geeignete Ersatzfont:  
In diesem Fall kann ein Dokument meistens ähnlich wie das Original angezeigt werden. Unter Windows wird z.B. die True Type Font Arial als Ersatzschrift für die PostScript-Font Helvetica verwendet.



## 5.4 Bilder und Grafiken

### Vergleich von Raster- und Vektor-Formaten

Bilder können, wie in Kapitel 4 bereits ausgeführt wurde, als Raster- oder als Vektorbilder in Dokumente eingebunden werden. Welches Format das geeignetere ist, hängt vom Bildmaterial und von der geplanten Verwendung des Dokuments ab:

- Bitmap-Bilder enthalten für jedes Pixel einen Farbwert. Falls keine Kompressionstechniken eingesetzt werden, hängt ihr Platzbedarf nur von der Bildgröße ab und kann damit schnell sehr hoch sein. Dagegen ist der Platzbedarf eines Vektorbildes unabhängig von der Größe der Darstellung aber abhängig von der Komplexität der Objekte im Bild.
- Das Anzeigen von Bitmap-Bildern ist einfacher als das von Vektorbildern. Browser und Viewer können daher immer Bitmap-Formate anzeigen, während sie Vektorformate oft nur mit Zusatzmodulen oder gar nicht anzeigen können.
- Rasterbilder besitzen eine feste Anzahl von Zeilen und Spalten. Eine vergrößerte Darstellung führt zu einem gröberen Raster, bei einer Verkleinerung geht Bildinformation verloren. Vektorbilder lassen sich ohne Qualitätsverlust vergrößern und verkleinern. Zur Darstellung auf einem Monitor müssen sie im letzten Schritt in ein Rasterbild verwandelt werden.
- True Type und PostScript-Fonts stellen die Form der Zeichen mit Vektor- und Kurvenzügen dar. Sie sind somit beliebig skalierbar, ohne dass die Schriftqualität sich verschlechtert.

### Grafikdarstellung am Bildschirm und beim Druck

Bei Print-Publikationen werden im Allgemeinen Vektorgrafiken bevorzugt, zumindest wenn feine Strukturen, Linien usw. darzustellen sind. Rastergrafiken verwendet man, wenn es sich um photorealistische Bilder handelt. Die beim Druck erforderliche Bildqualität führt aber zu sehr großen Bilddateien. Dateigrößen von 50 M Bit für eine A4-Seite sind keine Seltenheit.

Dagegen überwiegen bei elektronischen Publikationen Rasterformate, zumindest bei WEB-Dokumenten. Wegen der geringen Auflösung von Monitoren und der kleineren Bildformate kann man die Größe von Dateien in einem vertretbaren Rahmen halten. HTML-Browser können selbst gar keine Vektorformate darstellen; sie benötigen dazu geeignete Zusatzmodule, PDF-Viewer sind in der Lage, sowohl Raster als auch Vektorformate in Dokumenten darzustellen.

Nehmen Sie bei der Auswahl von Bildern Rücksicht auf die zur Verfügung stehende Auflösung der Zielplattform:

Auf einem Bildschirm sind großformatige Zeichnungen mit vielen kleinen Details oft nur mit unzureichender Qualität darstellbar, unabhängig davon, ob sie als Raster oder Vektorbild vorliegen.

Grafiken, die nur am Bildschirm dargestellt werden sollen, kann man vor der Distribution auf 72 dpi reduzieren und so Platz und Übertragungskapazität sparen.

Grafiken, die sowohl am Bildschirm angezeigt als auch gedruckt werden sollen, sollten diejenige Auflösung besitzen, die der Druck erfordert, also mindestens 200 dpi oder größer.

## Grafik-Dateiformate

Je nach Einsatzbereich werden unterschiedliche Dateiformate verwendet, um Grafiken zu speichern. Verwenden Sie jedoch nur weit verbreitete Standard-Formate, denn Spezialformate begrenzen die Portabilität der Dokumente. Auch von der Verwendung von Mal- und Zeichenprogrammen, die in Textverarbeitungs- und DTP-Systeme integriert sind, ist abzuraten, da bei einem Versionswechsel inkompatible Format-Änderungen nicht selten sind. Die gebräuchlichsten Datei-Formate im Publishing-Bereich sind:

- Das TIF-Format  
ist ein flexibles Bitmap-Format, das von fast jedem Bildbearbeitungs- und Seitenlayoutprogramm unterstützt wird. Scanner können ebenfalls Bilder im TIF-Format speichern. Das TIF-Format unterstützt die Farbmodelle CMYK-, RGB-, Lab, Paletten-, und Graustufen. Die Bilder können wahlweise unkomprimiert oder verlustfrei komprimiert sein. TIF-Dateien können auch mehrere Bilder gleichzeitig sowie Meta-Informationen enthalten.
- Das JPEG-Format  
ist ein Rasterformat, das bei der Kompression Bildinformationen nicht verlustfrei komprimiert. Die Kompression kann gesteuert werden, wobei stets ein Kompromiss zwischen Bildqualität und Platzbedarf zu finden ist. JPEG-Bilder eignen sich zur Distribution von Rasterbildern, die auf Monitoren angezeigt werden sollen, weniger für Zeichnungen mit feinen Strukturen und Linien.
- Das GIF-Format  
ist ein Rasterformat, das für die Distribution von Bildern im WEB weit verbreitet ist. Es arbeitet mit einer Farbpalette mit bis zu 256 Farben und ist daher relativ kompakt. Farbtöne, die nicht in der Palette vorkommen werden durch Dithertechniken approximiert. Das GIF-Format ist für Bilder, die zum Druck vorgesehen sind selten geeignet.
- Das EPS-Format (Encapsulated PostScript)  
ist das Standard-Format für hochwertigen Druck. EPS-Bilder erfordern zur Darstellung einen PostScript-Interpreter. Sie können jedoch ein Preview-Rasterbild enthalten, das zur Anzeige am Monitor verwendet werden kann. Ist kein Preview-Rasterbild enthalten, zeigen Viewer und Textverarbeitungssysteme wie Word einen leeren Rahmen als Ersatzdarstellung an.

## Screen Shots

Das Erstellen von Screen Shots, die in ein elektronisches oder ein gedrucktes Dokument eingefügt werden sollen, führt leicht zu qualitativ minderwertigen oder sogar völlig unbrauchbaren Bildern, wenn nicht einige grundlegende Regeln beachtet werden.

Ein Screen Shot, der in der Originalauflösung von 72 dpi und in der Originalgröße in ein anderes elektronisches Dokument eingefügt wird, wird dort in der Originalqualität dargestellt. Diese Situation ist also problemlos, aber selten gegeben.

Viel häufiger kommt es vor, dass Screen Shots verkleinert eingefügt werden. Dabei wird die Auflösung des Bildes zwar zunächst größer, aber die nachfolgende Bildkompression und die erneute Reduktion auf die Monitorauflösung bei der Darstellung durch den Browser können die Bildqualität stark reduzieren.

Vektorgrafiken sollten daher nicht als Screen Shot, sondern in einem Vektorformat in Dokumente eingefügt werden und nicht über den Umweg eines Screen Shot. Als Standard ist dafür das EPS-Format geeignet.

Optimieren Sie vor der Erstellung eines Screen Shot seine Originaldarstellung am Bildschirm:

- Den Bildschirm auf maximale Auflösung und Farbtiefe einstellen ( $\geq 1024 \times 768$ , 24 Bit)
- Das Anwendungsfenster maximieren

- Den Fensterinhalt auf das Wesentliche beschränken, keine leeren Flächen verschenken
- Eine große Font Size verwenden
- Notfalls einen Screenshot der Abbildung ohne Text machen und den Text als Text-Feld darüber legen.
- Überflüssige Status- und Icon-Leisten bei Browsern abschalten
- Screenshots nicht mit dem Windows Clipboard, sondern mit einem, Capture-Tool erstellen
- Bereits den Screenshot mit 24-Bit Farbtiefe aufnehmen.
- Nachbearbeitung mit einem leistungsfähigen Programm, z.B. Photoshop, durchführen.
- Vor der Nachbearbeitung die Bildauflösung des Screen Shot erhöhen, z.B. auf 300 dpi und diese erst ganz zum Schluss, falls nötig, wieder reduzieren.

### **Psychologische Wirkung von Farben**

Farben wirken auf einen Betrachter immer auch auf einer unbewussten Ebene. Die Farbenpsychologie hat nachgewiesen, dass ihre Wirkungen bei allen Menschen ähnlich ist. Ein bewusster Umgang mit Farbe ist einerseits notwendig, um Fehler zu vermeiden, andererseits können Farben aber auch zur Manipulation der Zielgruppe eingesetzt werden. Im folgenden werden typische Wirkungen von Primärfarben angegeben. Man nimmt an, dass die Wirkungen von Farbmischungen sich aus den Eigenschaften der Farbbestandteile ergeben.

- Weiß  
gilt als Symbolfarbe der Reinheit und Unschuld, in Asien auch der Trauer. Es ist überstrahlt wegen seines hohen Helligkeitswertes benachbarte Farben, so dass weiße Flächen größer wirken als gleich große dunkle.
- Schwarz  
vermittelt den Eindruck von Schwere und Ernst und Trauer, symbolisiert den Tod. Schwarze Farbflächen wirken optisch kleiner als gleichgroße anders gefärbte.
- Rot  
vermittelt Wärme, Kraft und Dynamik, aber auch Aggressivität und Gefahr. Rote Töne dominieren jede andere Farbe in ihrer Nähe.
- Grün  
wirkt erfrischend und beruhigend. Es symbolisiert Natur und Wachstum.
- Blau  
wirkt sachlich und kühl. Helle Blautöne wirken beruhigend. Blaue Gegenstände scheinen weiter entfernt zu sein. Die Wirkung blauer Farbtöne wird durch „warme“ Farben noch gesteigert.
- Gelb  
wirkt anregend und dynamisch. Gelb ist nach Weiß der hellste Farbton und überstrahlt daher alle anderen Farben. In Nachbarschaft mit kontrastierenden Farbtönen (z.B. Violett) steigert es seine Wirkung noch.
- Orange  
Ist ähnlich warm und anregend wie Rot, durch den Gelbanteil aber weniger aggressiv. Orange erweckt immer Aufmerksamkeit und wird oft als Warnfarbe benutzt.
- Violett  
ist als Mischung zweier gegensätzlicher Primärfarben – Blau und Rot – selbst sehr spannungsgeladen und vermittelt eine feierliche Wirkung. Je nachdem welcher der beiden Farbbestandteile überwiegt wirkt Violett eher kühl oder eher warm.

Zusätzlich zum Farbton gehen auch von den weiteren Farb-Parametern psychologische Wirkungen aus:

- Farben mit hoher Sättigung wirken lebendig und ziehen die Aufmerksamkeit auf sich
- Farben mit geringer Sättigung, also hohem Grauanteil vermitteln Weichheit und Romantik.
- Ein hoher Helligkeitsgrad vermittelt Weite und Entspannung.
- Dunkle Farben symbolisieren Schwere, Kraft, Enge.

## 5.5 Typographische Grundlagen

Die Typographie blickt auf eine Jahrhunderte alte Tradition im Bereich der Printmedien zurück. Ihre Regeln gelten auch für elektronische Publikationen, denn Texte sind immer noch der wichtigste Bestandteil von Dokumenten, auch dann wenn sie zusätzlich multimediale Komponenten enthalten. Es kommen durch technische Randbedingungen und die Art der Nutzung jedoch neue Aspekte hinzu. Dieser Abschnitt stellt die wichtigsten typographischen Begriffe und Regeln unter besonderer Berücksichtigung dieser Aspekte zusammen.

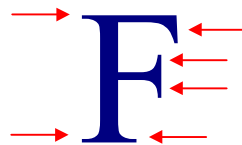
### Merkmale der Schriftzeichen

Wichtige Merkmale, die das Erscheinungsbild einer Schrift prägen, sind:

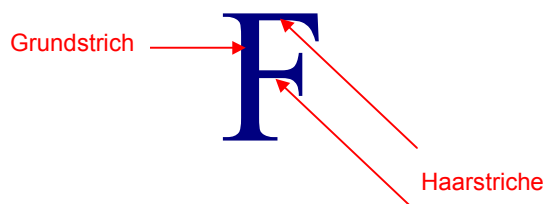
- die Ober-, Mittel- und Unterlängen



- die Serifen, das sind die kleinen Endstriche der Buchstaben:



- die Grund- und Haarstriche; mit dem Duktus einer Schrift bezeichnet man den Wechsel zwischen Grund- und Haarstrichen. Er kann mehr oder weniger stark ausgeprägt sein.



### Ziffernarten

Die meisten Schriften enthalten heute Versalziffern, das sind Ziffern, deren Höhe derjenigen von Großbuchstaben entspricht und die alle gleich breit sind. Sie sind vor allem für Tabellen und Formeln geeignet, bei denen eine gleichmäßige Ausrichtung die Lesbarkeit erhöht.

In Fließtexten werden auch Minuskel-Ziffern verwendet. Sie haben wie die Kleinbuchstaben (=Minuskeln) der Schrift Ober- und Unterlängen sowie verschiedene Zeichenbreiten und ergeben somit ein gleichmäßigeres Schriftbild.

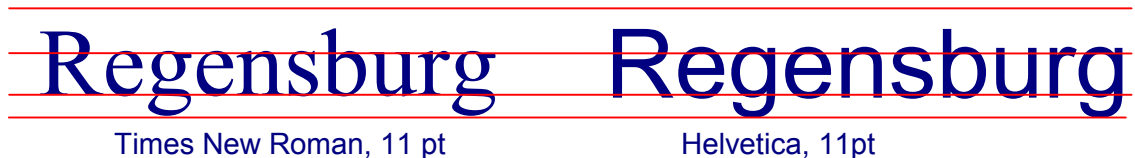
### Merkmale von Schriften

Die geringe Auflösung von Bildschirmen beeinflusst die Lesbarkeit einer Schrift und führt zu anderen Regeln als bei Printmedien:

- Serifen und Haarstriche  
Während bei Printmedien Serifen die Lesbarkeit einer Schrift verbessern, ist dies am Monitor umgekehrt: Feine Serifen und Haarstriche verringern die Lesbarkeit vor allem bei kleiner Schriftgröße.

- Laufweite  
Die Laufweite, also der Abstand der Zeichen innerhalb von Wörtern sollte am Bildschirm größer gewählt werden, damit sich die Buchstaben besser voneinander unterscheidbar werden.
- Kursive Schriftformen  
Sie sind weniger gut durch die Pixelmatrix des Bildschirms darstellbar als geradlinige Schriftformen und sollten deshalb vermieden werden.
- Mischen unterschiedlicher Schriften  
Im Regelfall sollten in einem Dokument nicht mehr als zwei unterschiedliche Schriften verwendet werden, z.B. eine für den Fließtext und eine andere für Überschriften. Um besondere Wirkungen zu erzielen, kann von dieser Regeln auch abgewichen werden.
- Schriftgröße  
Beim Bleisatz wurde als Schriftgröße die Höhe der Bleilettern, die „Kegelhöhe“, bezeichnet. Wegen der konischen Form der Lettern war die Höhe der gedruckten Zeichen etwas kleiner und diese waren von einem Leerraum, dem „Fleisch“ umgeben. Die Schriftgröße wurde in Punkten gemessen.  
Bei elektronischen Schriften wurden diese Bezeichnungen beibehalten. Wie viel Leerraum um die Zeichen vorhanden ist, ist nicht mehr technisch bedingt, sondern hängt vom Schrifttyp ab.

Für die optische Wirkung ist auch die Größe der Mittellängen wichtig. Wenn unterschiedliche Schriften kombiniert werden, sollten vor allem die Mittellängen gleich groß gewählt werden.



Die Schriftgröße sollte mindestens 12 dpi betragen, denn einerseits sind kleinere Zeichen bei dem üblichen Lese-Abstand von 40 cm zum Bildschirm schlecht erkennbar, andererseits stehen bei größerer Font Size mehr Pixel für die Darstellung eines Zeichens zur Verfügung, was die Detail-Wiedergabe verbessert.

### **Absatzformate**

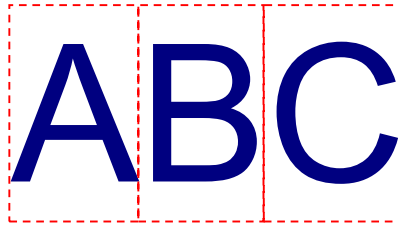
Das Schriftbild eines Textes hängt wesentlich von der Wahl des Absatzformates ab.

- Der Blocksatz  
hat eine ruhige, statische Wirkung. Durch Einfügen von Zwischenraum zwischen Wörtern und/oder Buchstaben innerhalb von Wörtern wird der linke und rechte Textrand gerade ausgerichtet. Ungleichmäßige Wortabstände verschlechtern aber die Lesbarkeit und können bei schmalen Textspalten zu halb leeren Zeilen führen.
- Der Flattersatz  
richtet der Text nur am linken oder rechten Rand geradlinig aus. Das Problem ungleichmäßiger Wortabstände entfällt, die Lesbarkeit ist der des Blocksatzes vergleichbar. Rechtsbündiger Flattersatz erschwert dem Auge die Identifikation der Zeilenanfänge und ist daher für längere Texte ungeeignet.
- Der zentrierte Satz  
eignet sich nur für kurze Texte, z.B. für Überschriften. Die Längen aufeinanderfolgender Zeilen sollte deutlich differieren.

## Abstände zwischen den Zeichen

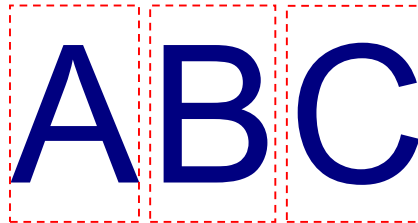
Der Leerraum zwischen den Elementen eines Textes ist für seine Wirkung ebenso von Bedeutung, wie die Elemente selbst.

- Abstand zwischen Buchstaben  
Durch den freien Raum, um die Zeichen, das „Fleisch“ ist bereits ein Standardabstand zwischen den Zeichen vorgegeben.



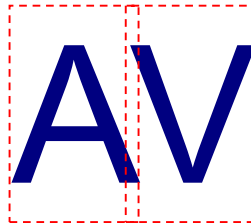
Arial, Schriftgröße 72 pt

Die Laufweite kann durch zusätzlichen Zwischenraum noch vergrößert werden, z.B. bei „gesperrtem“ Druck.



Arial, Schriftgröße 72 pt, Laufwerte um 4 pt erweitert

Die Verringerung der Laufweite (Unterschneiden) eines Textes kann bei großen Schriftgrößen ein geschlosseneres Schriftbild ergeben. Auch für bestimmte Zeichenkombinationen kann eine individuelle Verkleinerung der Laufweite, das Kerning, ein gleichmäßigeres Schriftbild ergeben.



Arial, Schriftgröße 72 pt, Unterschneiden um 4 pt.

## Abstände zwischen Wörtern

Als horizontaler Abstand zwischen den Wörtern eines Textes ist ein Drittel der Schriftgröße üblich. Das entspricht bei vielen Schriften der Breite eines kleinen „i“. Der Wortzwischenraum sollte jedoch immer kleiner als der Zwischenraum zwischen den Zeilen sein, damit der horizontale Zusammenhalt einer Zeile größer ist als der vertikal aufeinanderfolgender Zeilen.

Als wir jüngst in Regensburg waren, .....

Als*i*wir*i*jüngst*i*ni*i*Regensburg*i*waren, .....

## Abstände zwischen Zeilen im Absatz

Der Zwischenraum zwischen den Unterlängen einer Zeile und den Oberlängen der darauffolgenden Zeile wird als Durchschuss bezeichnet. Er sollte größer als der Wortzwischenraum sein, im Normalfall also größer als ein Drittel der Schriftgröße.

## Abstände zwischen Absätzen

Eine optische Trennung von Absätzen unterstützt die inhaltliche Struktur des Textes. Wichtig ist, dass eine einmal gewählte Methode im gesamten Text konsequent durchgehalten wird. Einige verbreitete Techniken der Absatztrennung sind:

- zusätzlicher Abstand zwischen Absätzen  
kann entweder durch Leerzeilen oder durch einen größeren Durchschuss in der ersten Zeile eines Absatzes realisiert werden.
- Einzug der ersten Absatzzeile  
Dabei wird das Erste Wort des Absatzes nach rechts eingerückt oder nach links aus dem Absatz herausgerückt.
- Absatzen als auslaufende Zeile:  
Die letzte Zeile eines Absatzes ist auch beim Blocksatz nicht durch Leerräume aufgefüllt und daher kürzer als innere Zeilen.
- graphische Markierungen:  
z.B. Pfeile, Punkte, Dreiecke vor der ersten Zeile eines Absatzes, Trennlinien zwischen Absätzen

## Hervorhebungen

Überschriften, einzelne Wörter oder ganze Absätze können durch Variationen im Schriftsatz hervorgehoben werden. Dabei ist es auch hier wichtig konsequent zum gleichen Zweck die gleiche Methode zu verwenden.

Zunächst stellen wir einige gute Methoden zur Hervorhebung vor:

- Fettdruck  
ist eine wirksame Möglichkeit zur Hervorhebung und durch verschiedene Strichstärken – z.B. sei bald, bald, extra bald – gut steuerbare Art der Hervorhebung

nicht fett    **fett**  
nicht fett    **fett**

- Kapitälchen  
Dabei werden Kleinbuchstaben durch Großbuchstaben ersetzt, die die Höhe der normalen Kleinbuchstaben besitzen. Damit erreicht man eine gute Hervorhebung bei gleichzeitig guter Lesbarkeit.

KAPITÄLCHEN  
KAPITÄLCHEN

- farbiger Text  
ist eine wirksame und durch die Wahl des Kontrastes zur normalen Textfarbe gut steuerbare Methode zur Hervorhebung.

Regensburg    Regensburg    Regensburg    Regensburg

farbige Text-Unterlegung  
ist ebenfalls eine wirksame Methode der Hervorhebung

Regensburg    Regensburg    Regensburg



Im folgenden werden einige weniger empfehlenswerte Hervorhebungstechniken aufgeführt:

- Versalien  
Dabei werden nur normale Großbuchstaben verwendet. Es wird eine gute Hervorhebung erreicht, aber die Lesbarkeit wird verringert

VERSALIEN  
VERSALIEN

- Unterstreichung  
erzielt eine wirksame Hervorhebung, ist aber aus typographischer Sicht unbefriedigend, insbesondere, wenn dabei Unterlängen gestört oder ganz überdeckt werden

Jahr_2000	<u>Jahr 2000</u>
Jahr_2000	<u>Jahr_2000</u>
Jahr_2000	<u>Jahr 2000</u>

- Kursive Schriften  
sollten für Hervorhebungen am Bildschirm nicht verwendet werden, da – bedingt durch die geringe Auflösung der Monitore – der Kontrast zu regulären Schrift nicht groß genug ist und die Darstellungsqualität geringer ausfällt.

nicht kursiv	<i>kursiv</i>
nicht kursiv	<i>kursiv</i>

- Gespernte Schrift  
ist weniger empfehlenswert, weil dadurch der Lesefluss gebremst wird und die gleichmäßige mittlere Helligkeit des Schriftbildes gestört wird.
- Blinkende Schrift  
sollte unter allen Umständen vermieden werden, da sie alle Aufmerksamkeit auf sich zieht und der Rest einer Seite kaum noch wahrgenommen wird.

### **Text-Hintergründe**

Die Struktur und Farbe des Hintergrundes kann die Wirkung eines Textes wesentlich beeinflussen. Es folgen einige Hinweise die Sie in diesem Zusammenhang beachten sollten:

- Ein weißer Bildschirm als Hintergrund strahlt zu hell und ermüdet die Augen beim Lesen. Vermeiden Sie dies durch eine leichte farbige Tönung des Hintergrundes.
- Durch begrenzte farbige Flächen lassen sich Texte gliedern und gegenüber anderen Informationen auf einer Seite (z.B. Navigationselementen) hervorheben.
- Durch unterschiedliche Hintergründe in einzelnen Kapiteln einer Publikation, kann dem Leser implizit vermittelt werden, wo er sich gerade befindet.
- Der Text kann dunkler als der Hintergrund sein (Positivfläche) oder heller als der Hintergrund (Negativfläche).
- Ein leicht strukturierter Hintergrund kann den Text gut hervorheben und dekorativ wirken, dagegen kann eine zu starke Struktur im Hintergrund den text völlig unleserlich machen, insbesondere bei kleinen Schriftgrößen.

## 6. PostScript-Grundlagen

### 6.1 Die historische Entwicklung von PostScript

Die Seitenbeschreibungssprache PostScript wurde Anfang 1985 von Adobe Systems eingeführt, zunächst auf dem Apple LaserWriter. Heute ist PostScript ein anerkannter Industriestandard, der wegen der hohen Investitionen in Hard- und Software und der weiten Verbreitung noch lange Bestand haben wird.

Aufgrund der reichhaltigen graphischen Möglichkeiten und der Vielfalt PostScript-fähiger Geräte unterstützen viele Software-Hersteller PostScript in ihren Produkten. Ebenso gibt es eine große Palette von freien Tools für das Arbeiten mit PostScript.

Außerdem lizenziert Adobe die PostScript-Technologie an Druckerhersteller. PostScript-fähige Drucker müssen leistungsfähiger ausgestattet sein als andere Drucker. Zusammen mit den Lizenzgebühren führt das zu einem deutlich höheren Preis für PostScript-fähige Drucker. Beispiel: (Beim HP LaserJet 2100 macht das z.Zt. ca 375,- DM aus, 5/2000)

Da Adobe die PostScript-Spezifikation offengelegt hat, gibt es inzwischen auch PostScript-Drucker mit anderen PS-Interpretern, Die früher noch deutlichen Qualitätsunterschiede sind eingeebnet, aber Unterschiede in den Ausdrucken sind durchaus möglich, schon deshalb, weil die Spezifikation selbst nicht immer eindeutig ist..

Um PostScript und dessen Grafikmodell zur Ausgabe und Darstellung nicht nur für Drucker, sondern auch für Bildschirme zu nutzen, wurde 1988 PostScript unter der Bezeichnung Display PostScript als Bestandteil des Betriebssystems NEXTSTEP implementiert und später auch unter Unix in das X Window System übernommen.

Mit der Zeit wurde die PostScript-Spezifikation erweitert, aktuell ist heute die Version 3.

Adobe nutzte die Erfahrungen mit PostScript bei der Entwicklung des PDF-Formats, das als künftiger PostScript-Nachfolger angesehen werden kann. Die auf dem PDF-Format aufbauende Acrobat-Technologie besitzt einerseits die Möglichkeiten von PostScript zur plattformunabhängigen Layout-Beschreibung, vermeidet aber viele Nachteile von PostScript.

### 6.2 Merkmale von PostScript

PostScript ist eine Seitenbeschreibungssprache, die unabhängig von einer bestimmten Darstellungsplattform das Layout von Dokumenten beschreibt. PostScript ist seitenorientiert; die Elemente, die auf einer Seite dargestellt werden, können von unterschiedlichem Typ sein:

- Texte in verschiedenen Schriftarten mit verschiedenen Attributen (Größe, Farbe, Schriftschnitt, Schreibrichtung usw.,) in hoher Qualität.
- Raster-Bilder, d.h. rechteckige Bereiche, deren Inhalt durch die Farbwerte ihrer Pixel beschrieben wird.
- Vektorgraphiken, also Linien, Rechtecke, Kreise, Polygone, Kurven usw.

PostScript beschreibt einerseits die Form dieser Objekte, andererseits können sie geometrischen Transformationen unterworfen werden (Größenänderungen, Drehungen, Scherungen, Verzerrungen, ....).

Zusätzlich zur Beschreibung des Seiten-Layout legt PostScript auch fest, wie die Inhalte einer Seite auf Ausgabegeräten dargestellt werden:

- Die Objekte einer Seite müssen in eine gerasterte Darstellung für Ausgabegeräte transformiert werden, wobei diese sehr unterschiedlich hohe Auflösungen besitzen können (z.B. 72 dpi bei Bildschirmen und 2540 dpi bei einem professionellen Belichter)
- Die im Layout enthaltenen Farben müssen in die Gerätefarben unterschiedlicher Ausgabegeräte möglichst originalgetreu transformiert werden.

PostScript ist eine eigenständige Programmiersprache, mit Funktionen, Variablen und Datentypen. Somit besitzt PostScript eine weit umfangreichere Funktionalität als andere Druckerkontrollsprachen, z.B. PCL. Bei der Ausgabe eines PostScript-Dokuments auf einem Drucker wird ein PostScript-Programm an diesen geschickt. Seine Abarbeitung durch den PostScript-Interpreter des Geräts führt dann zur Darstellung der Seite. Gerade wegen seiner umfassenden Funktionalität besitzt PostScript aber auch eine hohe Komplexität.

In der gesamten Druckvorstufe hat sich PostScript auch als Datenaustauschformat durchgesetzt für den Austausch von einzelnen Grafiken und mehrseitigen Dokumenten bis hin zu vollständigen Büchern. Für einzelne Grafiken wird das EPS-Format verwendet (Encapsulated PostScript), das auf PostScript aufbaut. EPS hat sich in der Druckvorstufe als Standard für den Austausch von Bildern etabliert.

### **Vorteile**

Der wichtigste Vorteil von PostScript gegenüber anderen Dokumenten-Beschreibungssprachen ist die sehr hochwertige, Plattform-unabhängige Layout-Beschreibung. Bei der Darstellung auf einem speziellen Gerät wird dieses Layout vom PostScript-Interpreter dann entsprechend den Geräte-Eigenschaften möglichst genau reproduziert. Eine PostScript-Datei wird daher auf unterschiedlichen Geräten weitgehend identisch dargestellt. PostScript wird zur Ausgabe auf sehr verschiedenen Geräten verwendet: außer Druckern aller Art, auch auf Belichtungsmaschinen, großformatigen Plottern usw.

Ein weiterer Vorteil von PostScript ist die Unabhängigkeit vom Betriebssystem. PostScript-Dateien selbst sind ASCII-Dateien und somit auf allen Betriebssystemplattformen lesbar. Somit konnte sich auch unter allen relevanten Betriebssystemen ein PostScript-fähiges Softwareangebot entwickeln.

Zum Standard wurde PostScript aber auch wegen der ausgefeilten Unterstützung von Zeichensätzen und Schriftarten. Ein zentraler Bestandteil der Schriftenbehandlung in PostScript ist die flexible Zuordnung von Schriftarten und Zeichensätzen. Buchstaben werden durch Bezier-Kurven beschrieben und sind daher wie andere Grafik-Elemente transformierbar. Für die schnelle und hochwertige Darstellung der Zeichen durch den Interpreter sorgen gut optimierte Algorithmen. Dies alles führte dazu, dass PostScript-Schriften heute in der gesamten Druckvorstufe zum Standard geworden sind und sich ein unübersehbares Angebot solcher Schriften entwickelt hat.

### **Nachteile**

Trotzdem sollte man sich auch der Nachteile von PostScript bewußt sein:

- Die Komplexität von PostScript stellt hohe Anforderungen an die Leistungsfähigkeit der PostScript-Interpreter im Drucker in Bezug auf Rechengeschwindigkeit und Speicher.
- Die Größe von PostScript-Dateien ist ein Nachteil bei der Übertragung über das Netz und bei ihrer Speicherung.
- PostScript ist als Austauschformat konzipiert, d.h. eine nachträgliche Bearbeitung von PostScript-Dateien ist in der Regel nicht möglich.

### **Sicherheitsaspekte**

PostScript enthält Operatoren zum Bearbeiten und Löschen von Dateien, insofern kann jede PostScript-Datei Schaden anrichten, wenn sie angezeigt wird. Für Ghostscript können kritische Dateioperationen mit der Aufrufoption `-dSAFER` verhindert werden. Diese läßt sich auch in Ghostview im Menüpunkt „Optionen/Konfiguriere Ghostscript...“ einstellen.

### 6.3 PostScript-Interpreter (RIP)

Die Anweisungen in einer PostScript-Datei, die zum Ausgabegerät übertragen wird, werden dort von einem Interpreter ausgewertet, dem „Raster Image Processor“ (RIP). Dieser analysiert zunächst die PostScript-Daten, um darin Anweisungen zu identifizieren. PostScript-Anweisungen können entweder Objekte einer Seite beschreiben oder Berechnungen durchführen. Als Ergebnis liefert der RIP zunächst eine interne Darstellung der Seite ab, oft in Form einer Display-Liste. Daraus kann er unterschiedliche Daten generieren, z.B.:

- In einem Laserdrucker steuert der RIP den Laserstrahl. Endprodukt ist die bedruckte Seite oder der belichtete Film zum Erstellen der Druckplatten.
- Andere RIPs setzen PostScript-Daten in eine andere Druckersprache um, z.B. in PCL oder in Rasterdaten zur Anzeige am Bildschirm (Beispiel: Ghostview)
- Manche RIPs rastern die PostScript-Daten und geben sie in einem Dateiformat für Rastergraphiken aus, z.B. TIF, PICT, WMF oder in einem objektorientierten Dateiformat, z.B. PDF, PICT oder WMF.

RIPs in Geräten stellen über die Interpretation der Layout-Beschreibung hinaus Funktionen zur Verfügung, wie man sie in Betriebssystemen vorfindet, z.B.

- die Kommunikation zwischen Rechner und Ausgabegerät
- die Steuerung der Drucker-Hardware (Papiereinzug, Platten, .....
- die Interpretation von interaktiv eingegebenen PostScript-Anweisungen.

Die Hardwareausstattung, die ein PostScript-fähiger Drucker benötigt, kann über die eines Standard-PC durchaus hinausgehen: Wichtig sind in erster Linie ein schneller Prozessor und ein großer Arbeitsspeicher, eventuell auch Plattenspeicher. Die Größe des Frame Buffer, d.h. des Arbeitsspeicherbereichs, in dem eine Druckseite aufgebaut wird, hängt vom Seitenformat, der Auflösung und der Farbtiefe ab. Für eine A4-Seite (8x11,5 Zoll) werden z.B. benötigt:

- auf einem Schwarz-weißen Laserdrucker mit 600 dpi Auflösung: ca 4 MB
- auf einem Farbdrucker mit 400 dpi Auflösung und 24 Bit Farbtiefe: ca 43 Megabyte

Auch wenn der Speicherbedarf durch komprimierte Speicherung des Druckbildes verringert werden kann, ist er immer noch beträchtlich.

#### RIP-Einsatzbereiche

PostScript-RIPs werden in verschiedenster Weise eingesetzt. Die wichtigsten Einsatzbereiche sind:

- PostScript-Interpreter im Drucker  
Die häufigste Einsatzform ist der PostScript-Interpreter in Laserdruckern. Entweder sind solche RIP fest im Drucker eingebaut oder als Einschub nachrüstbar.
- Hardware-RIPs  
Belichtungsmaschinen sind in der Druckvorstufe von großer Bedeutung. Sie erzeugen die Filme, mit denen die Druckplatten für den Offsetdruck geätzt werden oder belichten Dias. Meist werden sie von einem externen RIP angesteuert, einem Spezialrechner, auf dem nur der PostScript-Interpreter läuft. Digitale Druckmaschinen mit integriertem PostScript-RIP können aus den PostScript-Daten ohne Umweg über einen Film Druckplatten erzeugen.
- RIPs für Farbkopierer  
Damit kann man einen hochwertigen Farbkopierer zu einem Farb-PostScript-Drucker umfunktionieren. (Zwei der RIO-Farbdrucker im URZ sind (Minolta-)Kopierer mit Hardware-RIPs „Fiery XJ300“ der Firma EFI)

- Software-RIPs  
Software-RIPs sind PostScript-Interpreter, die auf einem normalen Rechner ablaufen. Während solche Lösungen ursprünglich eher als Notlösung zur Kosteneinsparung entwickelt wurden, stellen Sie heute bei ausreichender Rechenleistung eine sinnvolle Alternative dar. Bekannte Beispiele sind:
  - Ghostscript  
Dieser weit verbreitete Public Domain Software-RIP zeigt PostScript-Dateien am Bildschirm an. Zur als Ausgabe können über 100 verschiedene Druckermodelle angesteuert werden.
  - Distiller  
In der Version 4 verarbeitet Distiller PostScript Level 3 und erzeugt daraus PDF-Dateien.
- Display PostScript  
Wenn man PostScript zur Ansteuerung von Bildschirmen verwendet, treten gegenüber der statischen Ausgabe auf Drucker neue Anforderungen auf: Fensterinhalte ändern sich dynamisch durch Scrollen, Überlagern usw. und müssen daher ständig neu gezeichnet werden. Das einheitliche Beschreibungsmodell für Bildschirm und Drucker erhöht die Qualität der Bildschirmdarstellung und erleichtert die Programmierung. Bei der Druckausgabe müssen die Daten nicht mehr umgewandelt werden, sondern es genügt, die PostScript-Anweisungen statt auf den Bildschirm zum Drucker zu leiten.
- PostScript-Fax  
PostScript-Fax arbeitet mit einem Faxmodem, das in einen PostScript-Drucker integriert ist, und erweitert das Übertragungsprotokoll für Faxdaten so, daß alternativ zu den Rasterdaten auch PostScript-Daten übertragen werden können. Ist das empfangende Gerät ebenfalls PostScript-fähig, so überträgt der Sender PostScript- statt der Fax-Daten. Das Übertragungsprotokoll muß dafür in der Lage sein, den Gerätetyp des Empfängers zu erkennen.

## 6.4 PostScript-Treiber

### Ins Betriebssystem integrierte PostScript-Treiber

Bei den Windows-Betriebssystemen sind die Druckertreiber Bestandteil des Betriebssystems und stehen daher allen Anwendungen gleichermaßen zur Verfügung, nämlich über die GDI-Schnittstelle (GDI=Graphical Device-Interface). Da das GDI sowohl die Bildschirm- als auch die Druckerausgabe steuert, genügt es in Anwendungsprogrammen im wesentlichen, die Bildschirmausgabe auf den Druckertreiber umzulenken, um zu drucken. Vorteilhaft dabei ist:

- dass ein neuer Druckertreiber nach seiner Installation sofort allen Anwendungen zur Verfügung steht und
- dass die Anwendungs-Entwickler sich um die Details einzelner Druckertypen nicht kümmern müssen.

Diese standardisierte Druckerschnittstelle läßt sich auch verwenden, um beliebige Daten aus Anwendungen zu exportieren, z.B. PDF-Daten über einen PDF-Treiber oder Fax-Daten über eine Fax-Treiber.

### Programme mit integriertem PostScript-Treiber

Bei den Betriebssystemen MS DOS und Unix müssen Programme eigene Treiber für die unterstützten Geräte mitbringen. Unter Unix hat sich zwar mit dem XWindow System ein Standard für die Graphikausgabe etabliert, dieser ist aber nicht in allen Unix-Installationen vorhanden und unterstützt nicht das Drucken. Daher müssen Programme unter MS-DOS und Unix ihre PostScript-Ausgabe selbst erzeugen und können keinen vorinstallierten Treiber aufrufen.

Manche Software-Pakete für Windows umgehen den systemweiten Druckertreiber weitgehend. Sie benutzen den Treiber im sogenannten Pass-Through-Modus, in dem er die vom Programm erzeugten PostScript-Anweisungen unverändert weiterleitet. Dies kann z.B. sinnvoll sein:

- wenn die Anwendung bereits intern mit einer PostScript-Darstellung arbeitet, wie z.B. Adobe Illustrator, der EPS als Speicherungsformat für Graphikdateien verwendet und sie fast unverändert zum Drucker schicken kann.
- wenn die Anwendung effizienteren PostScript-Code erzeugen kann als der PostScript-Treiber wie z.B. Acrobat
- wenn die Anwendung Anforderungen, etwa bzgl. Farbtreue hat, die der Druckertreiber nicht erfüllen kann.

### PostScript-Konverter

stellen eine letzte Möglichkeit zum Erzeugen von PostScript-Ausgabe dar. Es handelt sich dabei um Filter, die ein bestimmtes Datenformat einlesen und daraus PostScript-Format erzeugen.

### PPD-Dateien

PostScript-Operatoren für die Layout-Beschreibung sind zwar geräteunabhängig und werden auf allen Geräten gleich interpretiert. Trotzdem sollte ein PostScript-Treiber in der Lage sein, die Eigenschaften eines bestimmten Druckers optimal zu nutzen, z.B. die Möglichkeit aus verschiedenen Papierkassetten unterschiedliche Papiersorten zu laden.

PostScript-Treiber bestehen daher aus zwei Teilen: der geräteunabhängige Teil ist für alle Drucker identisch, ein zweiter Teil erzeugt gerätespezifischen Code, damit spezielle Fähigkeiten eines Druckers auch genutzt werden können. Die geräteabhängigen Parameter werden vom Hersteller in einer PPD-Datei (PPD=PostScript Printer Definiton) bereitgestellt, deren Format von Adobe festgelegt wurde.

PPD-Dateien sind ASCII-Dateien und können auch vom Anwender modifiziert werden. Ihre Inhalte beschreiben u.a.

- die Drucker-Hardware (Papierformat, Auflösung, Speicherausbau, Papierschächte, .... )
- den PostScript-Interpreter (PostScript-Level, Farbmodell, .... )
- die Farbfähigkeiten des Druckers (Farbmodelle, Rasterfunktionen, Rasterfrequenzen, ... )
- die zur Verfügung stehenden Fonts
- Papier-Nachbearbeitung (sortieren, lochen, heften, stapeln, .... )

Ausschnitt aus einer PPD-Datei für einen Lexmark Optra-PostScript-Drucker:

```
*PPD-Adobe: "4.2"
*% Adobe PostScript(R) Printer Description File
*% For Lexmark Optra C LaserPrinter
*% For use with Windows NT
*% Produced by Lexmark International, Inc.
*% .....

*FormatVersion: "4.2"
*FileVersion: "1.2"
*LanguageVersion: English
*PCFileName: "LOPTRAC.PPD"
*Product: "(Lexmark Optra C LaserPrinter)"
*PSVersion: "( )"
*ModelName: "Lexmark Optra C LaserPrinter"
*ShortNickName: "Lexmark Optra C PS"
.....
*LanguageLevel: "2"
*Protocols: PjL
*TTRasterizer: Type42
*FreeVM: "6269952"
*ColorDevice: True
*DefaultColorSpace: RGB
*VariablePaperSize: True
*Throughput: "3"
*Password: "0"

*% === Paper Handling =====
*DefaultPageSize: Letter
.....
*PageSize A4: "
  1 dict dup /ManualFeed false put setpagedevice
  1 dict dup /Policies 1 dict dup /PageSize 2 put put setpagede-
vice
  2 dict dup /PageSize [595 842] put dup /ImagingBBox null put
setpagedevice"
*End
.....
```

```

*% === Font Information
=====
*DefaultFont: Courier
*Font Courier: Standard "(001.000)" Standard ROM
*Font Courier-Bold: Standard "(001.000)" Standard ROM
.....
*Font Helvetica: Standard "(001.000)" Standard ROM
*Font Helvetica-Bold: Standard "(001.000)" Standard ROM
*Font Helvetica-Oblique: Standard "(001.000)" Standard ROM
.....

*% === Color Separation Information =====
*DefaultColorSep: ProcessBlack.85lpi.600dpi/85 lpi / 600 dpi
*InkName: ProcessBlack/Process Black
*InkName: CustomColor/Custom Color
*InkName: ProcessCyan/Process Cyan
*InkName: ProcessMagenta/Process Magenta
.....
*% End of PPD file for Lexmark Optra C LaserPrinter

```

**6.5 DSC-Kommentare**

DSC-Kommentare wurden von Adobe eingeführt, um PostScript-Dateien zu strukturieren und Meta-Informationen über den eigentlichen PostScript-Code einzufügen. Sie sind als Kommentarzeilen in den PostScript-Code eingefügt (beginnend mit %%) und werden vom PostScript-Interpreter ignoriert.

Der Aufbau einer DSC-kompatiblen PostScript-Datei ist wie folgt:

Prolog	<pre> %!PS-Adobe-3.0       DSC-Kommentare, z.B. Seitenzahl, PostScript-Level, ..... %%EndComments </pre>
	<pre> %%BeginProlog       .... %%EndProlog </pre>
Script	<pre> %%BeginSetUp       PostScript-Code und Kommentare, die das ganze       Dokument betreffen, z.B. Papierschachtauswahl %%EndSetUp </pre>
	<pre> %%Page: 1 1       Beschreibung von Seite 1 </pre>
	<pre> %%Page: 2 2       Beschreibung von Seite 1 </pre>
	<pre> ..... </pre>
	<pre> %%Trailer       PostScript-Code und Kommentare </pre>

Wichtige DSC-Kommentare sind z.B.



`%!PS-Adobe-3.0`

Steht am Anfang einer PostScript-Datei, die den DSC-Konventionen entsprechend aufgebaut ist. Ist der einzige DSC-Kommentar, der nicht mit `%%` beginnt.

`%%Page: 2 2`

Damit wird der Beginn einer Seitenbeschreibung angezeigt. die erste Zahl gibt die logische, die zweite die physikalische Seitennummer an.

`%%BeginDocument . . . . . %%EndDocument`

Dient zur Klammerung von eingebetteten Dokumenten.

DSC-kompatible PostScript-Dateien ermöglichen eine wesentlich vielfältigere Verwendung von PostScript-Dateien:

- Mit Hilfe der Kommentare `%%Page:x y` kann man eine beliebige Seite eines PostScript-Dokuments anzeigen, ohne das Dokument linear vom Anfang bis zu ihr durchzublättern.
- Man kann einzelne Seiten, z.B. die ungeraden zum Drucken extrahieren
- Man kann mehrere eiten auf ein Blatt drucken

## 6.6 Übertragungsprotokolle

Außer den eigentlichen PostScript-Daten müssen bei der Übertragung weitere Dateien übertragen werden:

- zur Synchronisation der Kommunikation zwischen Rechner und Drucker
- zur Kennzeichnung von Anfang und Ende eines Druck-Job
- zur Protokollumschaltung, wenn ein Drucker verschiedene Druckersprachen versteht, etwa PostScript und PCL.

Die wichtigsten Protokolle sind

### Das Standard-Protokoll:

Dieses kommt bei der seriellen und parallelen Schnittstellen zum Einsatz. Dabei sind einige Zeichen als Steuercodes reserviert und dürfen daher nicht in den PostScript-Daten enthalten sein. Aus diesem Grund können keine beliebigen Binärdaten übertragen werden, sondern nur ASCH-Zeichen im Bereich von 32 bis 127 selbst dann, wenn die Binärübertragung physikalisch möglich wäre! Das wichtigste Steuerzeichen ist Ctrl-D (Bytewert 4), der sogenannte end-of-job-marker, das das Ende eines Druckauftrags bzw. den Start eines neuen angibt.

Da Ctrl-D in den PostScript-Daten dann nicht vorkommen darf, ist mit diesem Protokoll die Übertragung von Binärdaten, etwa von Pixelbildern nicht möglich. Diese müssen hexadezimal verpackt werden.

### Das Binary Control Protokoll (BCP)-Protokoll: `*Protocols: BCP`

Dieses Protokoll ermöglicht die Übertragung beliebiger Binärdaten über die serielle oder parallele Schnittstelle und wird von einigen Level-1-Druckern und vielen Level-2-Geräten unterstützt. Voraussetzung für BCP ist ein 8-Bit-fähiger Übertragungskanal, so daß z.B. eine auf 7-Bit-übertragung konfigurierte serielle Schnittstelle nicht in Frage kommt. Der

## Das Tagged Binary Control Protocol (TBCP).

\*Protocols: TBCP

Dieses Protokoll entspricht weitgehend BCP, enthält allerdings zusätzliche Steuersequenzen, die Anfang und Ende einer Übertragungsfolge kennzeichnen. Im Gegensatz zum Steuerzeichen Ctrl-D, das die Grenzen eines einzelnen Druckauftrags markiert, kann eine solche Übertragungsfolge aus mehreren PostScript-Druckaufträgen bestehen. Ist die Übertragungsfolge beendet, können Druckaufträge anderer Emulationen folgen, z.B. PCL.

## Die Printer Job Language (PJL):

\*Protocols: PIL

Dieses Protokoll wurde von Hewlett-Packard für die Laserjet-Familie definiert. Es steuert zusätzliche Druckerfunktionen, auch außerhalb der einzelnen Druckaufträge. Die folgenden Beispiele Anweisungen legen die Auflösung auf 600 dpi fest und stellen PostScript als Seitenbeschreibungssprache ein:

```
@PIL JOB
@PIL SET RESOLUTION = 600
@'PIL ENTER LANGUAGE = POSTSCRIPT
```

## 6.7 PostScript-Fehler

Beim Drucken auf einem PostScript-Drucker können Fehler an verschiedenen Stellen des gesamten Vorgangs auftreten:

- Datenübertragungsfehler (falsches Übertragungsprotokoll, Leitungsstörungen, Binärdaten über 7-Bit-Kanal)
- Geräteabhängige Operatoren im PostScript-Code, die der Drucker nicht unterstützt
- zu wenig Speicher im Drucker
- fehlender PostScript-Prolog (muß entweder in jeder Datei stehen oder resident im Drucker)

Fehlermeldungen des RIP können an unterschiedlichen Stellen ausgegeben werden:

- direkt am Bildschirm erscheinen (z.B. bei einem Software-RIP wie GhostScript oder wenn der Drucker bidirektional angeschlossen ist).
- in einer Log-Datei (z.B. beim Distiller)
- auf Papier ausgedruckt (dafür muß der Druckertreiber eingestellt werden).

PostScript-Fehlermeldungen haben die folgende Form:

```
%% Error <Fehlername> : Offending Command: <Operator>
```

### Beispiel einer Distiller-Fehlermeldung:

```
%%[ ProductName: Distiller ]%%
%%[ Error: Helvetica,Bold cannot be embedded due to licensing restrictions.]%%
%%[ Error: invalidfont; OffendingCommand: xshow ]%%

Stack:
[61 56 28 28 28 0]
(□□□□□□)

%%[ Flushing: rest of job (to end-of-file) will be ignored ]%%
%%[ Warning: PostScript error. No PDF file produced. ] %%
```

Methoden zum Beseitigen von Fehlern in PostScript-Dateien behandeln wir in der Übung zu diesem Kapitel.

## 7. PDF-Dateien

Als Ergänzung zum Stoff des folgenden Kapitels dienen

- das Portable Document Format Reference Manual, Version 1.3 [6]
- das Adobe Acrobat Development Kit, Version 4.0 R3 [8]

Die Unterlagen finden Sie zum Download auf der Link-Seite der Vorlesung

### 7.1 Die PDF-Dateistruktur

Die PDF-Spezifikation legt vier Bereiche fest, die den Aufbau einer PDF-Datei beschreiben:

- **Objekte**  
Dies sind einfache oder strukturierte Datenobjekte, z.B. booleans, numbers, strings, streams, ... , welche die Basisobjekte einer PDF-Datei darstellen.
- **Filestruktur**  
Diese bestimmt, wie Objekte zu speichern sind, wie man auf sie zugreift usw., und zwar unabhängig von der Semantik der Objekte.
- **Dokumentenstruktur**  
Die Dokumentenstruktur bestimmt, wie komplexe Objekte aus Basisobjekten aufgebaut sind, z.B. Seiten, Links, Annotationen, ....
- **Seitenbeschreibung**  
Diese bestimmt den konkreten Aufbau einer einzelnen Seite in einer PDF-Datei.

### 7.2 PDF-Objekte

PDF-Objekte entsprechen den Postscript-Datentypen. Im folgenden eine kurze Übersicht mit exemplarischen Beispielen. Eine Vollständige Beschreibung gibt das PDF-Reference Manual im Kapitel 4.

- **Booleans**  
Objekte dieses Type haben die Werte `true` oder `false`.
- **Numbers**  
Objekte dieses Type können integer- oder Real-Zahlen in Dezimalschreibweise sein, z.B.  
`47.11`      `0.12345`      `12345`
- **Strings**  
Diese haben Zeichenfolgen als Werte. Strings sind in runde Klammern einzuschließen. Der Backslash „\“ hat die Funktion eines ESCAPE-Zeichens für Sonderzeichen und dient zur Unterdrückung des Zeilenwechsels im String. Beispiele:  

```
(Dies ist ein String)
(Dies ist \
  auch ein String)
(Dies ist eine geklammerte Summe: \ ( 123 + 456 \ ))
```
- **Namen**  
Sie sind Namen von Objekten und werden mit einem Slash „/“ eingeleitet. Das erste Zeichen muß ein Buchstabe sein. Beispiele:  

```
/alpha
/Type
```
- **Arrays**  
Ein Array ist eine Folge von Objekten beliebigen Typs, die in „[„ und „]“ eingeschlossen sind. Beispiele für Arrays sind  

```
[1.23 /beta true false ]
[ 100 100 3000 2000 ]
```

– Dictionaries

Dictionaries sind die wichtigsten Bausteine einer PDF-Datei. Es handelt sich dabei um Listen von Objekt-Paaren der Form <key> <value> .

Keys müssen Objektnamen sein,  
Values dürfen beliebige Objekte sein.

Ein Dictionary wird mit „<<„ und „>>„ geklammert. Beispiel eines Dictionary vom Typ Catalog. Ein solcher Catalog ist Bestandteil jeder PDF-Datei. Hier verweist er auf die Liste der Seiten (Objektreferenz 3) und die Liste der Bookmarks (Referenz 2):

```
<<
  /Type /Catalog
  /Pages 3 0 R
  /Outlines 2 0 R
>>
```

– Streams

Streams sind Zeichenfolgen, ähnlich wie Strings. Sie müssen aber im Gegensatz zu Strings nicht auf einen Schlag ganz gelesen werden. Sie können insbesondere komprimiert sein. Beispiel eines Stream-Objekts:

```
111 0 obj
<< /Length 348 /Filter /LZWDecode >>
stream
D•phÜd ••FÂ• øp
EÃ`î, äe•™•□" 4^FfÄŠ`j;È•a&•I`D•-α"m•È•S••D"c>•Á¥±A@^F••G|c, \Üc2^
|•qQÄ1•<F#!" (PA8••&Á•Ä1, ®Ä•Ö••Lr0••Í•••Ö<] ••••"qĐäo2•LgCI¼Ü) .•@
Ö, v••Š•••H`Èg•' Åã1, î••"' J$ð\HßŠA' fŠ^òÊ1`Úí¶ñ•Æçu»PowÛp••...•œA£8•
Ûm•••`ân, îO'+^ÖB86•i•AÉrfT~\5••ë²rh•¶] "™&f{ (, `3...ùil•4iÆøÉ+Â•2"û
=Ä] ô#æó•-ãöü••òh•¼•Ú•••@-kßi>İ«°^Á-Ûû•Ä/î) ••.rj••({Š†ε`Z-
" 'H' †Á³ -•1È„•ÇÅ@j•
endstream
endobj
```

### 7.3 PDF-Datei-Aufbau

Eine PDF-Datei besteht grundsätzlich aus den folgenden vier Blöcken:

Header	enthält ein Kennungs-Flag, z.B. %PDF-1.3
Body	enthält die Objekte, die das Dokument beschreiben
Cross Reference Table	für Direktzugriff zu den Objekten
Trailer	verweist auf die Cross Ref.-Tabelle und einige andere Objekte der PDF-Datei

## Die Cross Reference Tabelle

Die Cross Reference Tabelle enthält in der ersten Zeile die Objektnummer des ersten Objekts und die Gesamtzahl von Objekten

Der Rest der Cross Reference-Tabelle besteht aus Subsections, die zusammenhängende Folgen von Objekt-Referenzen durch „Cross Reference - Entries“ beschreiben:

- benutzte Entries verweisen auf Objekte und haben die Form:  
    <Byte-Offset des Objekts> <Generationsnummer des Objekts> n
- unbenutzte Entries sind zu einer Liste verkettet und haben die Form:  
    <Objektnummer des nächsten freien Objekts> <Generationsnummer> f

Beispiel eine Cross-Reference Table:

xref		
0 5		Index des 1. Eintrags=0 / Anzahl von Einträgen=5
0000000000 65535 f		Anker der Liste der freien Objekte (Liste ist leer)
0000000009 00000 n		Referenz zum 1. Objekt
0000000074 00000 n		Referenz zum 2. Objekt
0000000120 00000 n		Referenz zum 3. Objekt
0000000179 00000 n		Referenz zum 4. Objekt

## Beispiel einer "leeren" PDF-Datei

Diese PDF-Datei wurde erzeugt mit Acrobat Distiller 4.0: Die Einstellung Monitor-Optimiert wurde dazu folgendermaßen modifiziert und (untereinem anderen Namen gespeichert:

- Voreinstellungen/Einstellungen/Allgemein/ ASCII-Format
- Voreinstellungen/Einstellungen/Allgemein/ PDF optimiert nicht aktiviert
- Voreinstellungen/Einstellungen/Komprimierung/ Text nicht komprimiert
- Voreinstellungen/Einstellungen/Komprimierung/ Keine Schriften oder Untergruppen einbetten
- Voreinstellungen/Einstellungen/Farbe/ Farbe nicht ändern

Bei der Distanzberechnung ist zu beachten, dass in der PDF-Datei eine Zeile mit X'0A' also einem Byte beendet wird.

---

```
%PDF-1.2
%âãÿÓ
```

---

```
2 0 obj
<<
/Length 0
>>
stream

endstream
endobj
```

---

```
3 0 obj
<<
/ProcSet [ /PDF ]
>>
endobj
```

---

```
1 0 obj
<<
/Type /Page
/Parent 4 0 R
/Resources 3 0 R
/Contents 2 0 R
>>
endobj
```

---

```
4 0 obj
<<
/Type /Pages
```

Seiten-Inhalt  
Länge des Inhalts-Stream=0  
leerer Stream (kein Seiteninhalt)  
Ressourcen der Seite: leer  
Seitenobjekt: Beschreibung der (einzigen) Seite  
Referenz zum Seitenbaum  
Objektreferenz (R) auf die Seiten-Ressourcen  
Objektreferenz (R) auf den Seiten-Inhalt  
Baum-Liste aller Seiten des Dokuments:  
enthält hier nur eine einzige Seite

```

/Kids [1 0 R]
/Count 1
/MediaBox [0 0 595 842]
>>
endobj
-----
5 0 obj
Catalog-Objekt des Dokuments
<<
/Type /Catalog
/Pages 4 0 R
Objektreferenz (R) auf den Seiten-Baum
>>
endobj
-----
6 0 obj
<<
/CreationDate (D:20000618202148)
/Producer (Acrobat Distiller 4.0 for Windows)
>>
endobj
-----
xref
Cross Reference Tabelle
0 7
Index des 1. Eintrags=0, 7 Einträge
0000000000 65535 f
Anker der Freispeicherliste (leer)
0000000103 00000 n
Objekt #1 = Seitenobjekt
0000000016 00000 n
Objekt #2
0000000065 00000 n
Objekt #3
0000000183 00000 n
Objekt #4
0000000264 00000 n
Objekt #5
0000000313 00000 n
Objekt #6
-----
trailer
Trailer
<<
/Size 7
Größe der Cross Refernce Table
/Root 5 0 R
Objektreferenz (R) zum Catalog-Objekt
/Info 6 0 R
Objektreferenz zum Info Dictionary der Seite
/ID [<119d74ea01e1f966de05120c3e2ccc5e><119d74ea01e1f966de05120c3e2ccc5e>]
2 File-Id's
>>
-----
startxref
Dokumenten-Anker
413
Cross Ref. Table 413 Bytes vom Dateibeginn
%%EOF
-----

```

## Beispiel einer PDF-Datei mit dem Text "Hello World !"

Diese PDF-Datei wurde mit den selben Distiller-Einstellungen wie im vorigen Beispiel erzeugt.

```

%PDF-1.2
%âãÿÓ
-----
2 0 obj
Seiten-Inhalt
<<
/Length 118
Länge des Inhalts-Stream=118 Bytes
>>
stream
Stream „Hello World“
BT
Begin of Text
/F1 1 Tf
Font 1 (Courier)
11.04 0 0 11.04 70.8406 757.8803 Tm
0 0 0 rg
/GS1 gs
Generic Graphics State
-0.0022 Tc
Character Spacing
0 Tw
Word Spacing
[(Hello) -597.9(World) -597.9(!)]TJ
ET
End of Text
endstream
endobj
-----
3 0 obj
Seiten-Ressourcen
<<
/ProcSet [/PDF /Text ]
/Font <<

```

/F1 4 0 R	Verweis zur Fontbeschreibung
>>	
/ExtGState	Extended Graphics State
<<	
/GS1 5 0 R	Verweis z. Extended Graphics State dictionary
>>	
>>	
endobj	
5 0 obj	Extended Graphics State dictionary
<<	
/Type /ExtGState	
/SA false	Stroke Adjustment = false
/SM 0.02	Glättungsparameter
>>	
endobj	
4 0 obj	Fontbeschreibung
<<	
/Type /Font	
/Subtype /Type1	PostScript Type 1-Font (Outline-Font)
/Encoding /WinAnsiEncoding	Code-Tabelle für die Font = Windows/ANSI
/BaseFont /Courier	Basis-Font
>>	
endobj	
1 0 obj	Seitenobjekt: Beschreibung der (einzigen) Seite
<<	
/Type /Page	
/Parent 6 0 R	Referenz zum Seitenbaum
/Resources 3 0 R	Objektreferenz (R) auf die Seiten-Ressourcen
/Contents 2 0 R	Objektreferenz (R) auf den Seiten-Inhalt
>>	
endobj	
6 0 obj	Baum-Liste aller Seiten des Dokuments: enthält hier nur eine einzige Seite
<<	
/Type /Pages	
/Kids [1 0 R]	
/Count 1	
/MediaBox [0 0 595 842]	
>>	
endobj	
7 0 obj	Catalog-Objekt des Dokuments
<<	
/Type /Catalog	
/Pages 6 0 R	Objektreferenz (R) auf den Seiten-Baum
>>	
endobj	

```

8 0 obj
<<
/CreationDate (D:20000618211003)
/Producer (Acrobat Distiller 4.0 for Windows)
>>
endobj
-----
xref
0 9
0000000000 65535 f
0000000431 00000 n
0000000016 00000 n
0000000185 00000 n
0000000336 00000 n
0000000279 00000 n
0000000511 00000 n
0000000592 00000 n
0000000641 00000 n
-----
trailer
<<
/Size 9
/Root 7 0 R
/Info 8 0 R
/ID [

```

Dokumenten-Info

Cross Reference Tabele  
Index des 1. Eintrags=0, 9 Einträge  
Anker der Freispeicherliste (leer)  
Objekt #1 = Seitenobjekt  
Objekt #2 =  
Objekt #3  
Objekt #4  
Objekt #5  
Objekt #6  
Objekt #7=Catalog-Objekt  
Objekt #8

Trailer

Größe der Cross Refernce Table  
Objektreferenz zum Catalog-Objekt  
Objektreferenz zum Info Dictionary der Seite  
Dokumenten-Anker  
Cross Ref. Table 741 Bytes vom Dateibeginn



## 7.4. PDFmark-Operatoren

Als Ergänzung zum Stoff des folgenden Kapitels dienen

- das PDFmark Reference Manual, Technical Note #5150 [7]
- Kapitel 6 von [1]

Das PDF-Format ist ein Distributionsformat. Idealerweise sollte daher die Bearbeitung des Dokuments immer im Format der jeweiligen Anwendung (Textverarbeitung, Graphik-Anwendung, Konstruktions-Software, .... ) erfolgen und nach der Konvertierung keine wesentlichen Veränderungen der PDF-Datei mehr erfolgen.

Mit dem PDFmark-Operator hat Adobe daher eine Schnittstelle geschaffen, mit dem man aus einer (fast) beliebigen Anwendung heraus Einträge in eine PDF-Datei bringen kann. PDFmarks werden in die Quelldatei eingefügt, beim Erstellen der Postscript-Datei in den Postscript-Code eingeschleust und von normalen Postscript-Druckern ignoriert. Der Acrobat Distiller erzeugt daraus aber die entsprechenden Objekte im PDF-Code.

Diese Technik ist insbesondere für die automatische Generierung von Verweisen in Dokumenten wichtig, z.B. vom Inhaltsverzeichnis auf die Text-Abschnitte, vom Text auf Abbildungen, vom Stichwortverzeichnis auf den Text. Sie ist bisher nur in wenigen Text- und Publishing-Programmen integriert (Framemaker, Ventura Publisher).

Mit MS Word kann man über die Feldfunktion {DRUCK ... } PDFmarks erstellen. Es lassen sich so in der Textdatei fast beliebige Funktionalitäten für die daraus erzeugte PDF-Datei einbringen.

Die folgende Tabelle gibt einen Überblick über die wichtigsten Objekte, die mit dem Operator PDFmark definiert werden können:

1.	Notes
2.	Links innerhalb des Dokuments
3.	Links zu anderen Dokumenten
4.	WWW-Links
5.	Bookmarks
6.	Article Threads
7.	Named Destinations
8.	Beschneiden von Seiten
9.	Viewer-Default-Einstellungen
10.	Page Open-Aktionen
11.	Document Open-Aktionen
12.	Generierung, Füllen und Zurücksetzen von Formularen
13.	Ausgefüllte Formulare an eine URL schicken
14.	Starten von Anwendungen
15.	Abspielen von Audio- und Video-Clips
16.	Einfügen von Javascript-Anweisungen
17.	Menü-Operationen starten.

## Einfügen von PDFmark-Operatoren in Word-Dokumente

Die Benutzeroberfläche von MS Word unterstützt bisher das Einfügen von PDFmark-Operatoren nicht speziell. Sie können jedoch mit der Feldfunktion

```
{DRUCK ..... }
```

an beliebigen Stellen eines Dokuments eingefügt werden. Im folgenden werden einige typische Beispiele dazu gezeigt. Die Feldfunktionen selbst sind im Normalfall nicht sichtbar, daher werden sie zusätzlich als normaler Text gezeigt.

## Link auf eine bestimmte Seite innerhalb des Dokuments

Das Rechteck das den Link enthält wird hier direkt in Postskript-Koordinaten definiert. Der Link zielt auf die 3. Seite des PDF-Dokuments.

```
{ DRUCK  
"[ /Rect [ 2500 2200 3000 3000 ]  
/Page 3  
/View [ /XYZ -5 797 1.5]  
/LNK  
PDFmark"  
}
```

## Definition von "named destinations" im Dokument und von Links dorthin

*Definition der "named destination"*

```
{ DRUCK "[ /Dest Ziel_A /Page 11 /View [ /FitH 5 ] /DEST PDFmark" }
```

*Definition des Link*

```
{ DRUCK  
"[ /Rect [1000 1800 1400 2200]  
/Dest /Ziel_A  
/Subtype /Link  
/ANN  
PDFmark"  
}
```

## 8. PDF-Formulare – noch einmal betrachtet

In Kapitel 3 haben wir bereits einfache Techniken zum Erstellen von PDF- Formularen mit Acrobat betrachtet. In diesem Kapitel lernen wir zusätzliche Möglichkeiten kennen:

- vordefinierte Funktionen für Formularfelder
- JavaScript in Formularen
- Techniken des Datenaustauschs zwischen WEB-Server und Formularen auf dem Client.

### 8.1 Vordefinierte Funktionen für Formularfelder

Links in einer Acrobat-Datei können nicht nur als Verknüpfung zu anderen Dokumenten oder auf eine andere Stelle im gleichen Dokument benutzt werden, sondern auch Anwendungen starten, Audio- oder Video-Medien abspielen und vieles andere. Analog dazu können die Felder eines Formulars mit „Vorgängen“ verknüpft werden, die beim Eintreten bestimmter Ereignisse angestoßen werden.

#### Trigger-Ereignisse

Auslösende Ereignisse für vordefinierte Aktionen können sein:

- Maustaste loslassen / drücken:  
Das Drücken bzw. Loslassen einer Maustaste löst den Vorgang aus.
- Maus in Feld / Maus aus Feld:  
Der Vorgang wird gestartet, wenn der Mauszeiger sich in das Feld hinein oder aus ihm heraus bewegt.
- Keystroke  
Ein Vorgang wird gestartet, wenn in einem Textfeld oder eine Kombobox ein Zeichen eingegeben wird (mit Tastatur oder über Paste) oder wenn in einer Kombobox oder einer Listbox ein Item ausgewählt wird.
- Validate / Calculate  
Wenn der Benutzer ein Feld verlässt, tritt das Ereignis „validate“ ein, an das Überprüfungen des Feldinhaltes geknüpft werden können. Danach folgt das Ereignis „calculate“, bei dem die Feldinhalte neu berechnet werden, die vom geänderten Feldinhalt abhängen (gemäß einer Liste „calculation order array“, s.u.)

Ein Feld kann auch mit mehreren Trigger-Ereignissen verknüpft sein.

#### Vordefinierte Aktionen

Die für Formulare wesentlichen vordefinierten Aktionen sind:

- Formular zurücksetzen:  
Dabei hat man die Alternativen, „alle“, „alle ausser ...“, „nur folgende ...“, Felder eines Formulars zurücksetzen zu lassen.
- Formulardaten absenden:  
Dabei kann man außer einer URL, an die das Formular geschickt werden soll auch spezifizieren, in dem die Daten gesendet werden:
  - als FDF-Datenstrom oder
  - im gleichen Format wie bei HTML-Formularen.Für die Auswahl der Felder, die versandt werden, bestehen die gleichen Optionen wie bei „Formular zurücksetzen“. Zusätzlich kann man angeben, ob leere Felder versandt werden sollen.
- Formulardaten importieren:  
Damit können als FDF-Datei gespeicherte Feldinhalte in ein Formular importiert werden.

- Feld ein/ausblenden  
Damit kann man beim Eintreten eines Ereignisses die Sichtbarkeit eines Feldes ein/ausschalten.
- JavaScript:  
Damit kann man durch selbst erstellte JavaScript-Funktionen bestimmen, welche Aktion beim Eintreten eines Ereignisses stattfinden soll.

### Beispiel: Versenden von Formulardaten im FDF-Format

Wenn wir aus einem Formular mit den folgenden Feldern

Nachname:	<u>Karl</u>
Vorname:	<u>Bauer</u>
Hochschule:	<u>Fachhochschule Regensburg</u>
Studiengang:	<u>Technische Informatik</u>
Semester:	<u>IT 3</u>
Matrikel-Nr.:	<u>123456789</u>
eMail:	<u>karl.bauer@stud.fh-regensburg.de</u>

die Formulardaten im FDF-Format versenden, werden diese in der folgenden Form versandt:

```
%PDF-1.2
%âãÿÓ
1 0 obj
<< /FDF << /Fields
[
<< /V (Karl) /T (Nachname)>>
<< /V (Bauer) /T (Vorname)>>
<< /V (Fachhochschule Regensburg) /T (Hochschule)>>
<< /V (Technische Informatik ) /T (Studiengang)>>
<< /V (IT 3) /T (Semester)>>
<< /V (123456789) /T (Matrikel-Nr)>>
<< /V (karl.bauer@stud.fh-regensburg.de) /T (eMail)>>
]
/F (Anmeldung.pdf)>>
>>
endobj
trailer
<< /Root 1 0 R >>
%%EO
```

### Validieren und Formatieren von Feldinhalten

Das Format von Eingaben in Textfelder kann vorgeschrieben werden. U.a. kann festgelegt werden, dass nur Zahlen oder nur Datumsangaben möglich sind und dass diese nach der Eingabe in bestimmter Weise formatiert werden.

Die Inhalte von Textfeldern lassen sich nach der Eingabe überprüfen, z.B. ob eingegebene Zahlenwerte in einem vorgeschriebenen Bereich liegen.

Eingaben können auch durch eigene JavaScript-Funktionen überprüft werden.

### Vordefinierte Berechnungen

Für Textfelder, als deren Inhalte Zahlen vorgeschrieben sind, können einfache vordefinierte Berechnungen durchgeführt werden:

- Summe / Produkt von Feldern
- Maximum / Minimum von Feldern

Die Aktualisierung berechneter Feldinhalte wird immer dann durchgeführt, wenn sich die Aktivierung eines der Felder, die in die Berechnung eingehen, ändert.

#### **Wichtig:**

Wenn Felder über mehrere Stufen voneinander abhängen, dann wird die Reihenfolge der Neuberechnung durch ein internes „calculation order array“ gesteuert. Mit der Acrobat-Menüfunktion „Werkzeuge / Formulare / Feldberechnungsreihenfolge festlegen ...“ kann man diese Reihenfolge beeinflussen.

Für komplexere Fälle muss JavaScript zur Berechnung eingesetzt werden.

### Beispiel

Im folgenden Beispiel ist für die oberen Zahlenfelder eine Formatierung als Euro-Betrag vorgeschrieben mit einem Komma als Dezimal-Trennzeichen. Im Feld „Maximum“ das Maximum aus der oberen Zahlenreihe eingetragen. Diese Auswertung wird immer dann durchgeführt, wenn der Mauscursor aus einem der oberen Felder herausgenommen wird.



### Maximum:

### Beispiel

Im folgenden Beispiel werden im Feld „Summe ohne MWSt.“ die Positionen 1. bis 5. addiert und danach im Feld „Gesamtbetrag“ das Produkt dieser Summe mit der Mehrwertsteuer, die im (nicht veränderbaren) Feld MWSt. steht, eingetragen.

Für die korrekte Auswertung ist die Berechnungs-Reihenfolge von Bedeutung !

Position	Anzahl	Artikel	Einzelpreis	Gesamtpreis
1.				
2.				
3.				
4.				
5.				

Summe ohne MWSt.

MWSt.

Gesamtbetrag



## 8.2 JavaScript in Formularen

Mit der Möglichkeit, PDF-Formulare mit JavaScript-Routinen auszustatten, erweitert die Möglichkeiten zur Entwicklung von Formularen ganz erheblich, so kann man damit z.B.

- komplexe Berechnungen der Feldinhalte vornehmen,
- zahlreiche Eigenschaften von Feldern dynamisch verändern
- Formulare mit verschiedenen Methoden versenden, drucken, ....
- Formulardaten importieren / exportieren

Acrobat Forms JavaScript basiert auf der JavaScript-Version 1.2, stellt darüber hinaus aber spezielle Funktionen zur Verfügung, mit denen man auf Formular-Objekte zugreifen kann. Es besteht jedoch kein Zugriff auf die eigentlichen PDF-Seiten-Inhalte unter den Formular-Feldern.

JavaScript-Code innerhalb von PDF-Dateien kann an unterschiedlicher Stelle stehen:

- auf Dokumenten-Ebene:  
Dokumenten-Script können in Acrobat mit „Werkzeuge / Formulare / Document-JavaScripts“ in jede PDF-Datei eingefügt werden und werden dann stets bei Öffnen der Datei ausgeführt.
- bei jedem Formularfeld:  
In diesem Fall werden die Scripts ausgeführt, wenn ein zugeordnetes Ereignis eintritt.

Die Dokumentation der Acrobat-JavaScript-Erweiterung wird mit Acrobat 4 in installiert und ist im Menüpunkt „Hilfe / Form JavaScript-Handbuch ...“ zu öffnen.

### Die Objekte von Acrobat JavaScript

Die Acrobat-JavaScript-Erweiterung definiert die folgenden Objekte:

- das Doc-Object  
In den Eigenschaften des Doc-Object ist die Dokumenteninformation gespeichert, die auch mit der Acrobat-Menüfunktion „Datei / Dokumenteninfo ...“ interaktiv verändert werden kann. Die Methoden dienen zur Manipulation dieser Eigenschaften. Darüber hinaus aber auch zum Import und Export der Formulardaten im FDF-Format, zum Versenden des Formulars als eMail usw.
- Das Event-Object  
dient zur Reaktion auf Ereignisse, z.B. um festzustellen, welches Objekt ein Ereignis verursacht hat, zur Validierung usw.
- das Field-Object  
Mit dem Field-Object können alle Eigenschaften eines Feldes abgefragt und verändert werden.
- Das this-Object  
Dieses dient zum Zugriff auf das aktuelle Objekt. Bei Dokumenten-Scripts ist dies das Dokument selbst, bei Feld-Scripts ist es das Feld, dem das Script zugeordnet ist.
- weitere Objekte  
Neben diesen Objekten gibt es einige seltener gebrauchte, z.B. das App-Object (zur Steuerung des Viewer-Anzeigemodus, zur Navigation), das Global Object (zur Definition statischer Daten, die von anderen Dokumenten benutzt werden können), das util-Object (zur Formatierung von Feldinhalten), das Console-Object, (zur Ausgabe von Textmeldungen)

### Beispiel: Drucken eines Seitenbereichs

Im folgenden Beispiel kann man einen Seitenbereich angeben, der dann mit dem Print-Button gedruckt wird.

von Seite                      bis Seite

Im Druck-Button wird dazu der folgende JavaScript-Code eingefügt:

```
//Zugriff auf die Objekte "Anfang" und "Ende" ermöglichen
var first = this.getField("Anfang");
var last  = this.getField("Ende" );

// Drucken mit Parametern aus den Objekten "Anfang" und "Ende"
// Seitenzählung bei "print" läuft ab 0
this.print (false, first.value-1 , last.value-1);
```

### Beispiel: Versenden einer PDF-Datei als Mail-Attachment

Mit dem folgen Button wird das Dokument unter einem generierten Dateinamen gespeichert und diese Datei als Attachment einer eMail versandt. Die Zieladresse wird hier aus einem Textfeld übernommen, das je nach Anwendungsfall vom Benutzer auszufüllen ist oder durch Import von FDF.Daten automatisch belegt werden könnte und in diesem Fall garnicht sichtbar sein muss.

Zieladresse:

Send-Button:

Der JavaScript-Aufruf im Send-Button hat die Form:

```
// Das Objekt Zieladresse verfügbar machen
var f = this.getField("Zieladresse");

// eMail versenden
this.mailDoc( true, f.value, "", "",
              "<Text der als Subject erscheint>",
              "<Text, der im eMail-Body versandt wird>");
```

Der erste Parameter des Aufrufs der Methode mailDoc gibt an, wie das Absenden gesteuert wird:

- bei **true** wird zunächst ein eMail-Fenster zur weiteren Bearbeitung geöffnet, das dann mit „send“ abgeschickt werden muss.
- bei **false** wird die eMail ohne weitere Eingriffsmöglichkeit versandt.

Im Gegensatz zum Versenden mit Hilfe der serverseitigen formail-DLL, das wir in Übung 3 benutzt haben, ist bei der obigen Methode der Absender der eMail nicht anonym, weil der Client die versandte eMail generiert und nicht eine DLL auf dem Server.

### 8.3 Serverseitige Verarbeitung von Formulardaten

Formulare können ihre Feldinhalte an einen Server senden, der sie weiterverarbeitet und ggfs. dem Client auch eine Rückmeldung liefert.

#### Versenden von Formulardaten

Für das Versenden der Daten eines Formulars wird die Methode Post benutzt. Dabei kann man zwischen zwei Formaten zur Übertragung der Daten wählen:

- dem URL-Format, das identisch mit dem von HTML-Formularen ist und daher hier nicht näher beschrieben wird.
- dem FDF-Format, das aus dem PDF-Format abgeleitet aber wesentlich einfacher aufgebaut ist. Ein Beispiel eines FDF-Datenstroms wurde bereits in Abschnitt 8.1 gezeigt.

Das Versenden kann mit einer Schaltfläche bewirkt werden, welcher der Vorgang „Formular senden“ zugeordnet wird. Dabei ist auch die URL des Serverseitigen CGI-Script anzugeben.

Wenn der Server Daten an das Formular zurücksenden soll, dann muss an die URL der Zusatz #FDF angehängt werden.

#### Die Konfiguration des Servers

Der Server muß zur Verarbeitung von FDF-Daten, die auch als MIME-Type kennen. Dazu `application/vnd.fdf` bei den Mime Types einstellen.

#### Das Acrobat FDF-Toolkit

Adobe stellt zum Auswerten von Formulardaten auf dem Server das „Forms Development Toolkit“ zur Verfügung. Dieses ist eine Bibliothek von Funktionen, mit denen man einen Strom von Formulardaten vom Client übernehmen und analysieren, aber auch FDF-Daten an das Formular zurücksenden kann. Das FDF-Toolkit wird für unterschiedliche Plattformen geliefert:

- als C-Bibliotheken für Windows und verschiedene Unix-Betriebssysteme
- als ActiveX-Bibliothek
- als Java-Schnittstelle, die auf der C-Bibliothek aufsetzt
- als Pearl-Schnittstelle, die auf der C-Bibliothek aufsetzt
- als PHP-Schnittstelle

In den Übungen benutzen wir die C-Bibliothek für Windows-Plattformen.

#### Beispiel: Einlesen eines FDF-Datenstroms und zurücksenden an das Formular

Das folgende Beispiel-Programm übernimmt einzelne Datenfelder aus dem FDF-Datenstrom und schickt die Inhalte in andere Felder desselben Formulars zurück.

```
/* -----  
 * ping1:   FDF toolkit demo program 1:  
 *         Accepts and parses data from a PDF Form and sends  
 *         the extracted field values back to the client form.  
 *         Kp / 00.07.09  
 * -----  
*/  
#include <stdlib.h>  
#include <stdio.h>  
#include "fdftk.h"  
#include <string.h>  
  
#define   BUFLen   100
```



```

/* -----
 * Error handler:  Sends an error message to the client
 * -----
*/
void errorMsg (char *msg)
{
    printf("Content-type: text/plain\n\n");
    fprintf(stdout, "Form processing error: %s\n", msg);
    fflush(stdout);
    exit(1);
}

/* -----
   ping1 main function
   -----
*/
void main(int argc, char* argv[])
{
    FDFDoc  infile  = NULL;      /* file handle for FDF input stream */
    FDFErc  retCode;           /* returned error code */
    ASInt32 howMany = 0;       /* number of environment parameters */

    char    value[BUFLLEN];     /* parameter value */

    /* Get length of environment data */
    howMany = atoi(getenv("CONTENT_LENGTH"));
    if (howMany==0) errorMsg("No Form data transmitted !");

    /* Open FDFDoc object associated with stdin */
    retCode = FDFOpen("-", howMany, &infile);
    if(retCode!=FDFErcOK) errorMsg("input stream could not be opened!");

    /* Only for Unix platforms */
    /* FDFInitialize(); */

    /* Get values of fields Input1, Input2, Input3, Input4 and put them */
    /* into the fields Ping1, Ping2, Ping3, Ping4 of the FDFDoc object */
    retCode = FDFGetValue (infile,"Input1", value, sizeof(value), &howMany);
    retCode = FDFSetValue (infile,"Ping1" , value,true);

    retCode = FDFGetValue (infile,"Input2", value, sizeof(value), &howMany);
    retCode = FDFSetValue (infile,"Ping2" , value, true);

    retCode = FDFGetValue (infile,"Input3", value, sizeof(value), &howMany);
    retCode = FDFSetValue (infile, "Ping3", value, true);

    retCode = FDFGetValue (infile,"Input4", value, sizeof(value), &howMany);
    retCode = FDFSetValue (infile,"Ping4", value, true);

    /* Now send the values back to the Client */
    /* Emit the correct HTTP header for the MIME type to stdout */
    printf("Content-type: application/vnd.fdf\n\n");
    fflush(stdout);

    /* Emit the FDFDoc object to stdout and close infile */
    FDFSave (infile, "-");
    FDFClose(infile);

    /* Only for Unix platforms */
    /* FDFFinalize(); */
}

```

## Literatur

- [1] Thomas Merz:  
Web Publishing with Acrobat / PDF  
Springer-Verlag, Heidelberg, 1998
- [2] Thomas Merz:  
Die PostScript- & Acrobat - Bibel  
t/m-Verlag, 1996
- [3] Thomas Merz:  
Web Publishing with Acrobat / PDF  
Springer-Verlag, Heidelberg, 1996
- [4] Adobe:  
Acrobat-Dokumentation: acro\_doku.pdf  
Zum Ausdrucken formatierte Version des Acrobat-Hilfe-Handbuchs  
Download auch auf der Link-Seite der Vorlesung
- [5] Adobe:  
The official Adobe Electronic Publishing Guide  
Adobe Press, 1998
- [6] Adobe:  
Portable Document Format Reference Manual, Version 1. 3  
Ist Bestandteil des Adobe Acrobat Development Kit  
Download auf der Link-Seite der Vorlesung
- [7] Adobe:  
PDFmark Reference Manual, Technical Note #5150  
Ist Bestandteil des Adobe Acrobat Development Kit  
Download auf der Link-Seite der Vorlesung
- [8] Adobe Acrobat Development Kit , Version 4.0 R3  
Download auf der Link-Seite der Vorlesung