

# Diplomprüfung zum/zur Webdesigner/-in

## Theorieteil

### Photoshop & Screendesign

#### 1. Fachbegriffe:

##### **Alphakanäle**

Der **Alphakanal** oder  **$\alpha$ -Kanal** ist ein zusätzlicher Kanal, der in Rastergrafiken zusätzlich zu den Farbinformationen die Transparenz (Durchsichtigkeit) der einzelnen Pixel (Bildpunkte) speichert; also ein zusätzlicher Kanal, in dem z.B. Transparenzen abgespeichert werden können

##### **Anti-Aliasing**

**Antialiasing**, auch *Anti-Aliasing*, bezeichnet in der Computergrafik Techniken zur Verminderung von unerwünschten, durch das begrenzt aufgelöste Pixelraster verursachten Effekten wie dem Alias-Effekt oder dem Treppeneffekt, die bei der Erzeugung einer Computergrafik entstehen können. Beim Antialiasing werden die Bildinhalte nicht nur ausschließlich am Pixel, sondern auch an anderen Positionen ausgewertet und in die Berechnung der Pixelfarbe mit einbezogen. Wird Antialiasing vorrangig dazu eingesetzt, um das durch den Treppeneffekt verursachte kantige Erscheinungsbild abzumildern, so spricht man auch von *Kantenglättung*.

##### **Vektormaske**

Vektormasken dienen dazu, auf einer Ebene scharfkantige Masken zu erstellen, wie Sie für Grafiken, und weniger für Rasterbilder typisch sind.

Stärke der Vektormaske: verknüpft Pfade eng mit den Ebenen.

Die Maske folgt allen Transformationen und Bewegungen der Ebene durch die Zuordnung der Vektormaske zur Ebene. Dennoch lässt sie sich weiterhin unter Sichtkontrolle mit den Pfadwerkzeugen bearbeiten. Natürlich kann man die Vektormaske auch ausblenden oder in eine klassische Ebenenmaske rastern lassen. Last but not least eignet sich die Vektormaske ganz wunderbar, um zusammen mit einer Ebenenmaske harte und weiche Freisteller zu kombinieren.

#### 2. Dateiformate:

##### **.jpg**

Bildformat, komprimiert, für detailreiche Fotos, behält Mill. Farben bei, darf jedoch nicht zu oft verändert/umgespeichert werden, geht auf Kosten der Auflösung.

##### **.gif**

Reduziert Farben, daher ergibt sich eine kleine Datenmenge. Kann für Animationen verwendet werden, Webformat. Kann Transparenzen darstellen, jedoch nicht „perfekt“.

##### **.png**

Mill. Farben, Webformat, 24 bit speichert verlustfrei und in größerer Datenmenge, stellt Transparenzen nahezu perfekt dar.

##### **.pdf**

Lassen sich auf jeder Plattform anzeigen und ausdrucken. Sie behalten das Erscheinungsbild der Originaldokumente bei und bewahren alle Informationen des Ausgangsdokuments. Durch Komprimierung und Codeoptimierung sind die Daten recht klein und können problemlos z.B. per FTP

übertragen werden. Unterschiedlichster Verwendungszweck z.B. als Austauschdokumente, zur Bildschirmpräsentation, als interaktive elektronische Publikationen oder als Grundlage für den Offsetdruck.

#### **.eps**

Vektorformat, beliebig vergrößerbar, lässt sich nicht überall öffnen, lässt sich jederzeit bearbeiten, unkomprimiert, in der Regel kleine Dateimenge.

#### **.tiff**

Bildformat, große Dateimenge, unkomprimiert, lässt sich nicht ohne ein Photoprogramm öffnen.

### 3. Farbprofile

Ein **ICC-Profil** (synonymer Begriff: Farbprofil) ist ein genormter Datensatz, der den Farbraum eines Farbeingabe- oder Farbwiedergabegeräts, z. B. Monitor, Drucker, Scanner etc. beschreibt. Ziel eines konsequent eingesetzten Farbmanagements ist, dass eine Vorlage, die mit irgendeinem Eingabegerät erfasst wurde, an einem beliebigen Ausgabegerät möglichst ähnlich wiedergegeben wird.

Colormanagement-Systeme können Geräte wie Scanner, Digitalkameras, Monitore, Drucker sowie Film- und Plattenbelichter aufeinander abstimmen. Die Farbe wird z. B. den Druckbedingungen entsprechend angezeigt.

ICC ist hier die Abkürzung für das International Color Consortium, einen 1993 gegründeten Zusammenschluss vieler Hersteller von Grafik-, Bildbearbeitungs- und Layoutprogrammen, mit der Absicht eine Vereinheitlichung von Farbmanagementsystemen zu erzielen.

### 4. Harmonische Proportionen

Eine Proportion von 4 gleichartigen Größen, wenn sich der Unterschied der 1. u. 2. zum Unterschied der 3. u. 4. verhält wie die 1 zur 4; also wenn  $a : d = b - a : d - c$ .

### 5. Was ist ein Gestaltungsraster, wie wird er konstruiert

**Gestaltungsraster:** Ordnungs- und Konstruktionssystem für Medienprodukte. Schema für ein durchgängiges Design von Büchern, Prospekten, Drucksachen, Benutzeroberflächen interaktiver Systeme. Ein solches Raster basiert auf einem horizontalen und vertikalen Koordinatensystem mit dem Ziel, Schrift, Bild, Farbe, Fläche und Raum systematisch, zweckgerichtet und lesefreundlich aufzubereiten.

Das Gestaltungsraster bei Webseiten ist eine maßstabsgerechte Einteilung des Bildschirms, mittels Hilfslinien in Bereiche für Navigation, Orientierung, Emotion, Inhalt, Layout etc. Hierbei wird das Format festgelegt z.B. 900 x 600 px.

### 6. Kernbereich einer Webseite

Eine **Website**, (deutsch der „Webauftritt“, „Webpräsenz“ oder „Webangebot“), ist ein ganzes Projekt im World Wide Web, das meist aus mehreren Webseiten oder Dokumenten (Dateien) und anderen Ressourcen besteht. Zudem wird ein Webangebot gewöhnlich durch eine einheitliche Navigation (Hypertext-Verfahren) zusammengefasst und verknüpft.

Jede Präsentationsform, also auch die Gestaltung einer homepage im internet, muß angemessen sein

- dem/den **Zweck(en)** und **Ziel(en)**
- dem/den **Adressaten**
- dem **Inhalt**,
- den medialen **Möglichkeiten** und
- **Ihren Fähigkeiten**.

inhaltlicher Kernbereich der Homepage *ist die Präsentation des Unternehmens* (Startseite, Wir über uns, Unsere Angebote, ...)

## 7. Fixe oder flexible Designs, Vorteile und Nachteile

Der entscheidende Unterschied von **flexiblen Layouts** (%-basiert zu **fixen Layouts** (px-basiert) ist, dass Sie sich im Rahmen der vom Designer festgelegten Grenzen automatisch der Breite des Browserfensters anpassen und sich dadurch ohne Zutun des Anwenders perfekt an unterschiedliche Fensterbreiten und Ausgabemedien anpassen können. Fixe Layouts können dies nicht. Der Seitenzoom ermöglicht dem Anwender lediglich, die vom Browser gerenderte Webseite *nachträglich* in Eigeninitiative seinen Bedürfnissen anzupassen. (->Heute noch hat der IE6 eine Verbreitung von ca. 40%, womit fast jeder zweite Anwender nicht einmal diese Möglichkeit hat).

**Nachteil flexibler Designs:** bei unzureichender CSS-Formatierung kann es passieren, dass der Text beim Aufziehen des Browserfensters unkontrolliert in die Breite geht, was ihn letztlich unansehnlich und schwer lesbar macht. Ein korrektes CSS löst aber diesen Nachteil:

- **Minimale Breite:** Wert in der Einheit Pixel (z.B.: 760px). Auf diese Untergrenze sollten alle pixelbasierten Inhalte abstimmt werden, sodass sie auch bei minimaler Layoutbreite noch fehlerfrei positioniert werden können.
- **Maximale Breite:** Wert in der Einheit EM (z.B. 90em). Dadurch hat man den Textfluss immer unter Kontrolle, d.h. Absätze werden nicht übermäßig breit. Gleichzeitig skaliert der Maximalwert mit der Schriftgröße.

-> Webdesigner haben nicht wirklich die Kontrolle über das Layout. Über die Darstellung entscheidet maßgeblich der Anwender durch Browserwahl, Schriftgrößeneinstellungen (gern auch direkt über das Betriebssystem), alternative Nutzer-Stylesheets und so weiter.

Ein fixes Layout lässt sich für den Rechner und Lieblingsbrowser des eigenen Chefs oder des Auftraggebers optimieren, aber eben nicht für die grenzenlose Vielfalt der Nutzeranforderungen im Internet.

## 8. Farbharmonie

### **Eine Farbharmonie erstellen**

Aus dem Menü für Harmonieregeln kann man verschiedene Optionen treffen. Nach ihnen wird die Farbgruppe zur Grundfarbe gebildet. Mit einem Klick auf das Rastersymbol kann die Auswahl auf die Farbbibliothek beschränkt werden. Die Farben können weiter bearbeitet- und im Farbfeld-Bedienfeld gespeichert werden. Es besteht die Möglichkeit, ein Variationsschema auszuwählen. Dabei stehen die Optionen "Farbtöne/Schattierungen", "Warm/Kalt" und "Strahlend/Gedeckt" zur Verfügung.

### **Farbharmonie-Regeln**

Man kann bei den Farbharmonien zwischen mehreren Regeln wählen. "Analog" ermittelt Farben, die im Farbkreis dicht nebeneinander liegen. "Monochromatisch" variiert eine Farbe in der Sättigung während "Schattierungen" die Farbe mit Schwarz abmischt. "Zusammengesetzt" kombiniert erdige Töne, die Anteile aller Primärfarben enthalten. "Komplementär" wählt Farben aus, die sich auf dem Farbkreis gegenüber liegen. "Teilkomplementär" kombiniert eine Farbe mit den beiden Nachbarfarben ihrer

Komplementärfarbe. Möchte man kontrastreiche Zusammenstellungen erzielen, wählt man "Hoher Kontrast". "Triade" kombiniert dagegen drei Farben, die auf dem Farbkreis gleich weit voneinander entfernt sind. "Tetrade" macht dasselbe mit vier Farben und "Pentagramm" mit fünf.

## 9. Wirkung von Formen

Farben und Formen sind Elemente, die Schwere oder Leichtigkeit, Gemütlichkeit oder Sterilität, Traurigkeit oder Fröhlichkeit, Sachlichkeit oder Unbeschwertheit... vermitteln. Diese Aufzählung ließe sich um ein Mehrfaches erweitern.

Farben und auch Formen rufen psychische und physikalische Assoziationen hervor, die bei der Gestaltung von Websites oder einer "optischen Identität" beachtet werden sollten. z.B. eine Eisdiele wirbt mit dem Logo einer Eistüte, die mit blutroten Eiskugeln gefüllt ist, Oder ein Bestattungsinstitut hat ein grünes Kreuz an seiner Tür. Das wäre absolut unpassend; Eis soll erfrischen, und die Farbe müsste eine kühlende oder erfrischende Assoziation hervorrufen. Bei dem Bestattungsinstitut verstößt die Farbe Grün – die Leben und Gesundheit signalisiert – gegen das Pietätsempfinden.

### Beispiele

Gelb: fröhlich, intelligent, sonnig

Lila: depressiv, extravagant

Rot: dynamisch, aggressiv, gefährlich, signalisierend

Grau: technisch, neutral, modern, sachlich

Die physikalische Wirkung von Farben wurde in Untersuchungen nachgewiesen: Lagerarbeiter transportieren einen Tag lang schwarze Kisten und am darauffolgenden Tag helle Kisten. Trotz des gleichen Gewichts wurden die dunklen Kisten als schwerer empfunden. Diese Wirkungen müssen bei jeder gestalterischen Maßnahme beachtet und abgewägt werden, um Ihrem Produkt ein angemessenes Image zu verschaffen.

# Web-Design

## 10. Fachbegriffe

### **Javascript**

JavaScript ist eine clientseitige, objektbasierte Scriptsprache und gilt als "Actionebene", d.h. sie ermöglicht die Darstellung und Einbindung spezieller (interaktiver und somit veränderbaren) Funktionen auf der Webseite.

### **HTML**

XHTML gilt als Strukturebene und enthält Inhalte/Content, während CSS (Cascading Syle Sheet) die Präsentationsebene darstellt und sämtliche Informationen zu Layout und Formatierung enthält.

### **CSS**

CSS (Cascading Syle Sheet) stellt die Präsentationsebene darstellt und enthält sämtliche Informationen zu Layout und Formatierung.

## 11. Unterschiede Javascript HTML, CSS (was dient wofür)

JavaScript ist eine clientseitige, objektbasierte Scriptsprache und gilt als "Actionebene", d.h. sie ermöglicht die Darstellung und Einbindung spezieller (interaktiver und somit veränderbaren) Funktionen auf der Webseite.

XHTML gilt als Strukturebene und enthält Inhalte/Content, während CSS (Cascading Syle Sheet) die Präsentationsebene darstellt und sämtliche Informationen zu Layout und Formatierung enthält.

CSS (Cascading Syle Sheet) stellt die Präsentationsebene darstellt und enthält sämtliche Informationen zu Layout und Formatierung.

## 12. Was ist das DOM

Das **Document Object Model (DOM)** ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Sie wird vom World Wide Web Consortium definiert.

Eine Implementierung, die dieser Spezifikation genügt, besteht im Sinne der objektorientierten Programmierung aus einem Satz von Klassen zusammen mit deren Methoden und Attributen. Sie erlaubt Computerprogrammen, dynamisch den Inhalt, die Struktur und das Layout eines Dokuments zu verändern. Dokumente werden logisch wie ein Stammbaum dargestellt. Knoten (*nodes*) stehen über "Verwandtschaftsbeziehungen" zueinander in Verbindung.

## 13. Wozu dienen im Javascript Variablen, Bedingungen und Schleifen

### **Variablen**

Speicherbereiche, in denen Daten, die im Laufe von Programmprozeduren benötigt werden, gespeichert werden können. Der Inhalt, der in einer Variablen gespeichert ist, wird als "Wert" bezeichnet. Den Wert einer Variablen kann man jederzeit ändern. Um mit Variablen arbeiten zu können, müssen die benötigten Variablen zuerst definiert werden.

### **Bedingungen** (=Verzweigungen):

Bedingungen sind etwas sehr wichtiges in der Programmierung, so auch bei JavaScript. Mit ihnen kann man Code-Teile, die nur unter bestimmten Bedingungen ausgeführt werden sollen, realisieren. Es gibt zwei sehr verschiedene Arten von solchen Kontrollstrukturen. Die erste Art ist die *if-Struktur*. Man kann sie am besten verstehen, wenn man sich ihre deutsche Übersetzung anschaut: *wenn*. Bei dieser Verzweigung wird einfach der Code in dem Anweisungsblock nach der if-Klausel ausgeführt.

#### **Hier nochmal die allgemeine Form:**

1. if (Bedingung){
2. //Anweisungen
3. }

### **Schleifen**

Mit Schleifen kann man eine beliebige Befehlsfolge mehrmals wiederholen. Wenn schon vorher feststeht, wie oft eine Schleife durchlaufen werden muss, benutzt man meist die for - Schleife. Ansonsten nimmt man eine while - Schleife.

Die Syntax der **while - Schleife** erlaubt es aber auch, Schleifen wie eine for-Schleife zu programmieren. Deshalb benutzen manche Programmierer nur diesen einen Schleifentyp.

## 14. Zusammenspiel HTML - Javascript (wo stehen die Funktionen, wie werden sie gestartet,)




**JavaScript** ist kein direkter Bestandteil von **HTML**, sondern eine eigene Programmiersprache, um Webseiten zu optimieren.

JavaScripts werden wahlweise direkt in der HTML-Datei oder in separaten Dateien notiert. Sie werden zur Laufzeit vom Web-Browser interpretiert. Dazu besitzen moderne Web-Browser entsprechende Interpreter-Software. Es kann WWW-Seiten, den Browser sowie den Inhalt und das Verhalten von HTML-Seiten (Dokumenten) steuern. Es arbeitet plattformunabhängig, sodass es auf allen Computersystemen grundsätzlich funktioniert. Allerdings bestehen diverse Unterschiede zwischen den Browsern, die JavaScript unterstützen. Dies sind vor allem der Microsoft Internet Explorer und verschiedene Netscape-Browser.

Es gibt keine festen Vorschriften dafür, an welcher Stelle einer HTML-Datei ein JavaScript-Bereich definiert werden muss. Es ist unter JavaScript-Programmierern zur Gewohnheit geworden, einen solchen Bereich im Kopf der HTML-Datei, also zwischen `<head>` und `</head>` zu definieren. Dadurch ist am ehesten sichergestellt, dass der Code vom WWW-Browser bereits eingelesen ist und zur Verfügung steht, wenn er ausgeführt werden soll.


JavaScript-Code kann automatisch beim Einlesen der HTML-Datei ausgeführt werden. Das ist dann der Fall, wenn JavaScript-Befehle in einem JavaScript-Bereich außerhalb jeder selbst definierten Funktion stehen. In solchen Fällen ist es manchmal auch erforderlich, den Script-Bereich innerhalb des HTML-Dateikörpers, also innerhalb von `<body>...</body>`, zu notieren. Zum Beispiel, wenn man am Ende der Datei mit JavaScript dynamisch Datum und Uhrzeit der letzten Änderung an der Datei schreiben will. Gewisse besonders alte Browser, zum Beispiel Netscape 1 und Internet Explorer 2, wissen nichts von JavaScript und kennen daher das script-Element nicht. Wenn diese Browser auf einen JavaScript-Bereich treffen, stellen sie irrtümlicherweise dessen Inhalt so dar, als handelte es sich um normalen Text in der HTML-Datei. Dadurch bekommt der Anwender den Code zu Gesicht. Um dies zu verhindern, sollte der JavaScript-Code zwischen `<script type="text/javascript">` und `</script>` in einen HTML-Kommentar eingeschlossen sein. In der Zeile nach dem Start-Tag wird der Kommentar mit `<!--` geöffnet. In der Zeile vor dem End-Tag des JavaScript-Bereichs notiert man einen einzeiligen JavaScript-Kommentar `//`, gefolgt vom schließenden HTML-Kommentar `-->`.

**Das document-Objekt** bezieht sich auf den Inhalt, der in einem Browser-Fenster angezeigt wird. In der Objekthierarchie von JavaScript liegt es unterhalb des window-Objekts.

Im Document Object Model (DOM) des W3-Konsortiums ist das document-Objekt das Ausgangsobjekt für den Elementenbaum. Die Elemente eines HTML-Dokuments stellen dem DOM zufolge also Unterobjekte des document-Objekts dar. Die einzelnen HTML-Elemente können dabei eigene Eigenschaften und Methoden haben. Diese werden im Abschnitt über  HTML-Elementobjekte beschrieben. Entscheidend für den Zugriff auf den HTML-Elementenbaum sind beim document-Objekt die DOM-Methoden  getElementById und  getElementsByName.

Darüber hinaus enthält das document-Objekt selber eine Reihe wichtiger Eigenschaften und Methoden, die dokumentglobale Eigenschaften speichern oder Funktionen ausführen. Die meisten davon werden auch vom DOM unterstützt.

Eigenschaften des document-Objekts, wie etwa der Titel der angezeigten HTML-Datei (<title>...</title>), kann man mit document ansprechen, den Titel der Datei also document.title. Entsprechend werden Methoden des document-Objekts angesprochen, zum Beispiel die write-Methode mit document.write(). Man kann auch window.document.title oder window.document.write() notieren. Da es sich dabei aber um den Inhalt des aktuellen Fensters handelt, ist die Angabe des Fensterobjekts nicht zwingend erforderlich. Anders beim Zugriff auf andere Fenster oder Frame-Fenster. In diesen Fällen müsste man das Fenster korrekt adressieren. Für zusätzliche Fenster wird der Abschnitt zur Verwendung des window-Objekts und für Frame-Fenster den Abschnitt über die Verwendung des frames-Objekts gelesen.

Mit dem HTML-Element <noscript> können Bereiche definiert werden, die angezeigt werden, wenn JavaScript deaktiviert ist. Die Beschreibung dazu befindet sich im HTML-Bereich:  Noscript-Bereich definieren

#### 15. Anwendungsmöglichkeiten von Javascript (Beispiele nennen)

- Taschenrechner
- Monatskalender
- Persönliche Seitenbesuche zählen mit Cookies
- Formulareingaben überprüfen
- Dynamische grafische Buttons
- Aufenthaltsdauer auf Web-Seite anzeigen
- Verweisliste als Auswahlliste

#### 16. HTML Tags und die wichtigsten Attribute zu Grundstruktur

**body:** beinhaltet alle direkt "sichtbaren" Inhalte der Webseite (Text, Bilder, usw.)

**table:** erzeugt einen Tabellen-Container

**td:** bezeichnet eine mit Content befüllbare Zelle innerhalb einer Tabelle

**div:** div: Blockelement "Box" gehört zur Gruppe der Blockelemente und steht für division

**p:** steht für "paragraph" und erzeugt einen Absatz

**form:** erzeugt ein Formular

**input:** definiert ein Formularelement

**img:** steht für "image" und dient zur Einbindung eines Bildes auf der Webseite

**a:** steht für "Anker" und ermöglicht die Einbindung von Links

**ol:** steht für "ordered list" und erzeugt eine geordnete Liste mit Ranking-Darstellung der Inhalte

**ul:** steht für "unordered list" und erzeugt eine ungeordnete Liste ohne Hierarchien

**li:** steht für "list" und erzeugt Listenpunkte in Aufzählungsform

## 17. Headinhalt

Im Head (oder Kopf-Bereich) befinden sich sämtliche technische und dokumentarische Informationen wie z.B. Stylesheets; Text, der im Head steht, ist im Browser nicht sichtbar.

## 18. Link, Bild, Tabelle, Formulare

### **Link**

=Verweis; alle Verweise in HTML haben den gleichen Aufbau. Nachteile: es gibt keine Verweise für bestimmte Aufgaben. Es ist Aufgabe des Webdesigners, Verweise so gestalten, dass dem Anwender klar wird, ob es sich um einen Rückverweis zur Einstiegsseite, um einen Querverweis zu einem assoziativ verwandten Thema, oder etwa um einen Verweis zu einer ganz anderen Web-Adresse handelt.

Aufbau: Mittels dem XHTML\_Attribut „href:“ das ist sowas wie „src“, nur bezogen auf Links; href bezeichnet einen Quellanker und enthält die URL

### **Bild**

Um Grafiken in eine HTML-Dateien einzubinden, muss die Grafikdateien an gewünschten Stellen im HTML-Quelltext angegeben werden, z.B. ``.

Geeignete Dateiformate für Web-gerechte Grafiken sind vor allem GIF und JPEGoder PNG.

Grafiken fürs Web sollten nicht zu groß sind, denn aufwendige Grafiken verursachen lange Ladezeiten und Missmut beim Anwender. (-> Reduktion der Anzahl der Farben, verringerte Bildgröße, nicht zu viele Grafik-Referenzen in einer einzigen HTML-Datei). Breite und Höhe mit angeben.

Grafiken sind aus HTML-Sicht Inline-Elemente. In der "Strict"-Variante von HTML müssen solche Elemente innerhalb von Block-Elementen vorkommen, etwa in einem Textabsatz oder einem allgemeinen Bereich oder auch einer Tabellenzelle.

### **Tabelle**

`<table>` leitet eine Tabelle ein (*table = Tabelle*). Wenn die Tabelle sichtbare Gitternetzlinien enthalten soll, müssen im einleitenden `<table>`-Tag das Attribut `border` notiert und ihm ein Wert größer 0 zugewiesen werden. (Breite des Rahmens in Pixeln). Für eine blinde Tabelle ohne sichtbaren Rahmen und Gitternetzlinien, `border="0"` notieren.

`<tr>` leitet eine neue Tabellenzeile ein (*tr = table row = Tabellenzeile*). Im Anschluss daran werden die Zellen (Spalten) der betreffenden Reihe definiert. Am Ende einer Tabellenzeile wird ein abschließendes Tag `</tr>` notiert.

`<th>` leitet eine Kopfzelle ein, (Text in Kopfzellen wird hervorgehoben )

`<td>` eine normale Datenzelle (*th = table header, td = table data*). Der Inhalt einer Zelle wird jeweils hinter dem Tag notiert. Obwohl die zugehörigen End-Tags `</th>` bzw. `</td>` offiziell optional sind, ist dringend zu empfehlen, sie immer und in jedem Fall zu notieren.

In einer Tabellenzelle können beliebige Elemente stehen, d.h. außer normalem Text z.B. auch andere Block- und Inline-Elemente. Sogar eine weitere Tabelle können Sie innerhalb einer Zelle definieren.

Die Anzahl der Zellen sollte bei jeder Zeile gleich sein, sodass die Tabelle durchweg die gleiche Anzahl Spalten pro Zeile hat. In der ersten Zeile legt man deshalb durch die Anzahl der dort definierten Zellen die Anzahl der Spalten Ihrer Tabelle fest.

Tabellenzellen dürfen auch leer sein. Wenn Sie in einer Zeile für eine Spalte keine Daten eingeben werden sollen, notiert man `<td></td>`. Viele Web-Browser stellen die Zelle in diesem Fall als "nicht vorhanden" dar; deshalb die Notation `<td>&#160;</td>` für leere Tabellenzellen.

Wirkung der HTML-Elemente, die eine Tabelle erzeugen:

### Formulare

Mit `<form>...</form>` definiert man ein Formular (*form = Formular*). Alles, was zwischen dem einleitenden `<form>`-Tag und dem abschließenden Tag `</form>` steht, gehört zum Formular. Das sind hauptsächlich Elemente des Formulars wie Eingabefelder, Auswahllisten oder Buttons.

Um die Formularelemente zu platzieren und zu beschriften, können dazwischen aber auch andere HTML-Elemente notiert werden. Dabei muss folgendes beachtet werden: Gemäß der HTML-Variante Strict darf man innerhalb eines Formulars nur Block-Elemente (und Script-Bereiche) notieren, also etwa Überschriften, Textabsätze, allgemeine Bereiche oder Tabellen. In der HTML-Variante Transitional ist es dagegen außerdem erlaubt, zwischen `<form>` und `</form>` auch gemischten Inhalt aus Text und Inline-Elementen zu notieren. Im einleitenden `<form>`-Tag gibt das Pflichtattribut `action` an, an welchen URI die Formulardaten beim Absenden des Formular übertragen werden sollen. Der URI sollte die Adresse eines Programmes auf dem Server-Rechner sein, das die Formulardaten weiterverarbeitet. Üblicherweise handelt es sich dabei um ein CGI-Script, das z.B. in Perl und PHP geschrieben wurde.

Für den Wert des `action`-Attributes gelten die Regeln zum Referenzieren in HTML. Das Referenzieren mit relativen oder absoluten Pfadangaben ist also ebenso möglich

## 19. Die wichtigsten CSS Eigenschaften zur Formatierung von Text und Absätzen

### **position** (Positionsart):

*static* = normaler Elementfluss

*relative* = relative Positionierung (Verschiebung), gemessen an der Normalposition oder Anfangsposition des Elements selbst,

*absolute* = absolute Positionierung, gemessen am Rand des nächsthöheren Vorfahrenelements, das **nicht** die Normaleinstellung `position:static` aufweist. Scrollt mit,

*fixed* = absolute Positionierung, bleibt beim Scrollen stehen.

**width** (Breite): erlaubt ist eine numerische Angabe oder der Wert `auto` für automatische Breite.

**min-width** (Mindestbreite)/**max-width** (Maximalbreite): erlaubt ist eine numerische Angabe

**overflow** (Elementbereich mit übergroßem Inhalt): *visible, hidden, scroll, auto*

**float** (Textumfluss)-> nachfolgende Elemente umfließen das aktuelle Element bzw. den aktuellen Bereich. Folgende Angaben sind möglich: *left, right, none*

**clear** (Fortsetzung bei Textumfluss)

einen Umfluss abbrechen und die Fortsetzung unterhalb des umflossenen Elements oder Bereichs erzwingen. Folgende Angaben sind möglich: *left, right, both* (= Erzwingt Fortsetzung unterhalb), *none*

**z-index** (Schichtposition bei Überlappung)->bestimmt die Reihenfolge von überlappenden Elementen

**display** (Anzeigeart bzw. Nichtanzeige ohne Platzhalter)-> unterdrückt die Anzeige von Elementen oder Festlegung der Art der Anzeige; *block* (= Erzwingt einen Block), *inline* (= Anzeige im laufenden Textfluss), *inline-block, list-item, run-in,*

**font-family** (Schriftart)

**font-family** (Schriftart)

Bei mehreren angegebenen Schriftarten ist die Reihenfolge der Angabe entscheidend: ist die erste angegebene Schriftart verfügbar, wird diese verwendet. Ist sie nicht verfügbar, wird die zweite Schriftart verwendet, falls diese verfügbar ist usw.

Folgende **generische** Schriftfamilien sind fest vordefiniert - neben Schriftartnamen zu benutzen: serif , sans-serif , cursive, fantasy, monospace

**font-style** (Schriftstil): *italic* = kursiv, *oblique* = schrägestellt, *normal*

**font-variant** (Schriftvariante): small-caps = Kapitälchen, normal

**font-size** (Schriftgröße) erlaubt sind: numerische Angabe und Prozentangaben, absolute und relative Schlüsselworte

Prozentwerte beziehen sich auf die Schriftgröße des Elternelements.

Schlüsselworte: Absolut: xx-small , x-small , small, medium, large, x-large , xx-large

Relativ: smaller = kleiner als im Elternelement, larger = größer als im Elternelement.

**font-weight** (Schriftgewicht)

*bold* , *bolder* , *lighter* , 100,200,300,400,500,600,700,800,900 = extra-dünn (100) bis extrafett (900).  
*normal*

**font** (Schrift allgemein)

Kombinierung verschiedener Schriftformatierungen. Erlaubt sind die üblichen Wertangaben zu den erlaubten Eigenschaften, die in font zusammengefasst sind; font-size und font-family sind verpflichtend. Eine vorgegebene Reihenfolge ist aber einzuhalten; font-family wäre also beispielsweise als letzte Eigenschaft zu notieren.

**word-spacing** (Wortabstand); erlaubt sind numerische Angaben, jedoch keine Prozentangaben.

**letter-spacing** (Zeichenabstand); erlaubt sind numerische Angaben, jedoch keine Prozentangaben.

**text-decoration** (Textdekoration): *underline* , *line-through* , *blink* = blinkend, *none*

**text-transform** (Text-Transformation): *capitalize* , *uppercase* , *lowercase* , *none*

**color** (Textfarbe): Farbname (z.B. blue) oder #Nummer

**text-shadow** (Textschatten): erlaubt sind Farbangaben oder der Wert none für "keinen Textschatten" sowie numerische Angaben, jedoch keine Prozentangaben.

**font-stretch** (Schriftlaufweite): *wider* = weiter als normal, *narrower* , *condensed* , (*semi-extra-ultra*-) *condensed* , (*semi-extra-ultra*) *expanded* , normal

**line-height** (Zeilenhöhe)

**vertical-align** (Vertikale Ausrichtung)

**text-align** (Horizontale Ausrichtung) white-space (Textumbruch)

**margin** (Außenrand/Abstand allgemein)

**margin-top / margin-right / margin-bottom / margin-left**

**padding** (Innenabstand allgemein)

**padding-top / padding-right / padding-bottom / padding-left**

**border** (Rahmen allgemein)

**border[-top, -right, -bottom, -left]-width** (Rahmendicke)

**border[-top, -right, -bottom, -left]-color** (Rahmenfarbe)

**border[-top, -right, -bottom, -left]-style** (Rahmentyp)

**background** (Hintergrund allgemein)

**background-color / background-image / background-repeat / background-attachment**  
(Wasserzeichen-Effekt) / **background-position**

Möglichkeiten des Zusammenspiels HTML und CSS

XHTML gilt als Strukturebene und enthält Inhalte/Content, während CSS (Cascading Style Sheet) die Präsentationsebene darstellt und sämtliche Informationen zu Layout und Formatierung enthält.

## 20. CSS Fachbegriffe:

### **Vererbung**

Ein XHTML-Dokument ist in einer Baumstruktur aufgebaut. Jedes Element hat Vor- und evtl. Nachfahren, die man als Eltern- und Kind-Elemente bezeichnet. Formatiert man ein Element mit Stylesheets, so übernehmen nachfolgende Kindelemente einige dieser Formatierungen. Diesen Vorgang bezeichnet man als Vererbung.

### **Arten von Selektoren**

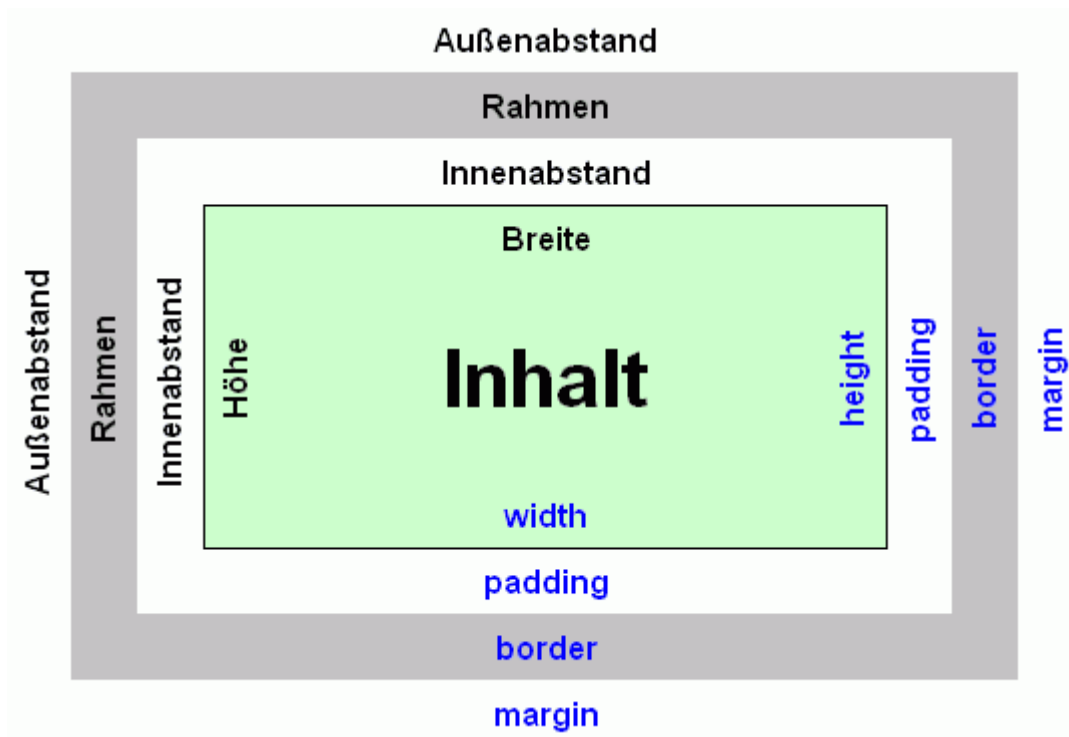
Selektoren stellen die Verbindung zwischen den zentralen Formaten im <head>-Bereich oder der externen Datei und den Elementen innerhalb der Seite dar, die formatiert werden sollen. Der Aufbau ist immer gleich: Selektor, geschweifte Klammer auf, Stylesheet-Angaben, geschweifte Klammer zu.

\* **Universal-Selektor** : betrifft alle Element der HTML-Seite

**Typ-Selektor** : Alle E-Tags der HTML-Seite, können öfter verwendet werden

**ID-Selektor** : Alle HTML-Tags mit dem Attribut id=" "

## 21. CSS Boxmodell und seine CSS Eigenschaften



Ein Blockelement erzeugt immer eine neue Zeile und nimmt immer die gesamte Breite seines Elternelementes ein, und kann durch die Eigenschaften `width`, `height`, `padding` und `margin` definiert werden. Ein Blockelement darf ausserdem jede Art von Inhalt beinhalten.

## 22. Möglichkeiten CSS-basierter Layouts

Mit CSS werden die Möglichkeiten des Webdesign nahezu grenzenlos. Durch die Trennung von Struktur und Design wird man flexibel. Änderungen können schnell durchgeführt- und eine HTML-Seite für mehrere Plattformen (z.B. Handy, Drucker, PDA) gleichzeitig zur Verfügung gestellt werden.

Das Grundprinzip von CSS : Über Styleanweisungen werden Strukturelementen in der HTML-Datei designtechnische Eigenschaften zugeteilt. Dazu zählen z.B. Abstand, Größe, Farbe und Position. Man kann entweder alle Tags eines Typs (z.B. das Tag <p>) gleichzeitig formatieren oder durch Vergabe von ID's oder Klassen ganz spezielle Bereiche bearbeiten.

# Flash

## 23. Fachbegriffe:

### **Timeline**

Das Timeline-Objekt entspricht der Flash-Zeitleiste. Sie können über `fl.getDocumentDOM().getTimeline()` auf die Flash-Zeitleiste des aktuellen Dokuments zugreifen. Diese Methode gibt die Zeitleiste der aktuellen Szene oder des zurzeit bearbeiteten Symbols zurück.

Wenn Sie mit Szenen arbeiten, hat die Zeitleiste jeder Szene einen Indexwert. Sie können mit `fl.getDocumentDOM().timelines[i]` auf die Zeitleiste einer Szene des aktuellen Dokuments zugreifen. (In diesem Beispiel ist `i` der Index des Zeitleistenwerts.)

Wenn Sie die Methoden und Eigenschaften des Timeline-Objekts bei Bildern einsetzen, sollten Sie daran denken, dass dieser Bildwert einen auf Null basierenden Index darstellt und nicht die tatsächliche Bildnummer in der Bildsequenz der Zeitleiste ist. Dies bedeutet, dass das erste Bild den Bildindex 0 hat.

### **Keyframe**

Die Keyframe-Klasse definiert den visuellen Zustand zu einer bestimmten Zeit in einem Bewegungstween. Die folgenden Eigenschaften sind die primären Animationseigenschaften: `position`, `scale`, `rotation`, `skew` und `color`. Ein Schlüsselbild kann eine oder mehrere dieser Eigenschaften definieren (optional). So kann sich beispielsweise ein Schlüsselbild nur auf die Position auswirken, während sich ein anderes Schlüsselbild zu einem anderen Zeitpunkt nur auf die Skalierung auswirkt. Dabei kann ein weiteres Schlüsselbild gleichzeitig Auswirkungen auf alle Eigenschaften haben. Bei einem Bewegungstween kann jede Zeitindexposition nur ein Schlüsselbild haben. Ein Schlüsselbild verfügt noch über weitere Eigenschaften, wie z. B. `blend mode`, `filters` und `cacheAsBitmap`. Diese Eigenschaften sind immer verfügbar. So verfügt ein Schlüsselbild zum Beispiel immer über einen Mischmodus.

### **Tweening**

Adobe Flash CS4 optimiert den Ablauf zur Erstellung von Schlüsselbildern und verfügt über einen neuen Bewegungs-Editor. In früheren Flash-Versionen mussten Sie Bewegungs-Tweenings von Hand in einer Reihe von zeitleistenbasierten Schritten erstellen. Mit dem neuen objektbasierten Bewegungsmodell hat sich die Erstellung von Schlüsselbildern und Bewegungs-Tweenings in einen automatisierten und rationellen Prozess gewandelt.

### **Symbole**

Man unterscheidet in Flash grundsätzlich drei Arten von Symbolen: Grafiken, Movieclips und Schaltflächen. Sie unterscheiden sich in ihrem Verhalten und ihrer Einsetzbarkeit, verfügen jedoch alle über eine eigene Zeitleiste und Bühne.

### **Instanzen**

Um eine neue Instanz eines Symbols zu erzeugen, muss man dieses in der Bibliothek auswählen und per Drag & Drop auf die Bühne des gewünschten (Schlüssel-)Bildes ziehen. Im Eigenschaftsinspektor kann man anschließend den Instanznamen definieren und die Eigenschaften der Instanz (nicht des Symbols!) wie etwa Größe sowie Instanzeffekte (Helligkeit, Deckkraft, Färbung, etc.) festlegen. Natürlich lässt sich auch die Position der Instanz verändern, indem man sie auf der Bühne mithilfe des Pfeilwerkzeugs bewegt.

## MovieClip

Movieclipsymbole dienen vorzüglich dem Erstellen von wieder verwendbaren Animationen mit von der Zeitleiste des Hauptfilms völlig unabhängiger Zeitleiste. Aufgrund dieser Unabhängigkeit sind Movieclips natürlich um weiten variabler einsetzbar als Grafiksymbole, da ihr Ablauf interaktiv steuerbar ist. Die einzelnen Instanzen sind via ActionScript über ihrer Namen ansprechbar und können so beispielsweise beliebig gestartet oder sogar einige Bilder ihrer Animation übersprungen oder wiederholt werden. Auch ist in Bezug auf das oben genannte Beispiel die Länge des Hauptfilms unerheblich. Das vollständige Abspielen der Animation, d.h. der Movieclipinstanz, bleibt stets gewährleistet, solange sie nicht durch interaktive Steuerungsanweisungen, d.h. mehr oder weniger gewollt, beeinflusst wird. Movieclipinstanzen können selbst wieder eine oder mehrere Movieclipinstanzen beinhalten, so dass sich eine Filmhierarchie in Organisation einer Baumstruktur ergibt. Zum Einbinden von animierten Symbolen sind Movieclips aufgrund der genannten Eigenschaften also besser geeignet. Beispielsweise ist das Erstellen von Schaltflächen, welche im Hoverzustand (Cursor steht über Schaltfläche) animiert sein sollen, nur mithilfe von Movieclip-, nicht aber von Grafiksymbolen möglich, da deren Zeitleiste nicht variabel abspielbar ist.

### 24. Was ist ActionScript und was kann damit bezweckt werden

Action Script ist die Schnittstelle HTML/Flash.

**ActionScript** (kurz: AS) ist eine Programmiersprache des US-amerikanischen Softwareunternehmens Adobe Systems und kann in einer Reihe von Adobe-Produkten eingesetzt werden, so zum Beispiel Adobe Flash, Flex und Air. Somit kann man von einfachen Flash-Animationen bis hin zu komplexen Webanwendungen, sogenannten Rich Internet Application, aber auch neuerdings Desktop-Anwendungen (Adobe Air) erstellen. Während Flash selbst anfangs zur Darstellung von Medieninhalten, wie Animationen und Filmen, diente, kam mit ActionScript die Möglichkeit interaktiver Anwendungen und Datenverarbeitung hinzu.

### 25. Sinnhaftigkeit von Flash auf Webseiten (Anwendungsmöglichkeiten)

Flash ist eine Autoren-Software zum Erstellen skalierbarer, interaktiver Animationen für das World Wide Web. Es dient als Grundlage für animierte Logos, Navigationselemente für Websites, lange Animationen oder komplette Flash-Websites.

### 26. In welchen Bereichen neben dem Web wird Flash noch eingesetzt Für Desktop-Anwendungen (Adobe Air)

## 27. Was muss beachtet werden, um die Dateigröße von Flash Anwendungen in Grenzen zu halten

- ▶ möglichst keine Pixel-Bilder im Film
- ▶ den Sound möglichst klein halten  
(8 Bit/11 kHz für Effekte, max. 8 Bit/22 kHz für längere Musik-Stücke, möglichst Mono)
- ▶ jeden Darsteller, der mehr als einmal im Film vorkommt, in ein Symbol umwandeln  
(und dann jedes Mal auch als solches verwenden)
- ▶ möglichst wenig verschiedene Schriften und Schriftschnitte (z. B. **Fett**, *Kursiv*) verwenden
- ▶ besser Tweening statt Bild-für-Bild-Animationen
- ▶ besser Linien statt Pinselstriche
- ▶ möglichst wenig verschiedene Linientypen
- ▶ Animationen in separaten Ebenen anlegen (Trennung von statischen Elementen)
- ▶ bei 3-D-Vektor-Objekten nicht sichtbare Elemente löschen
- ▶ komplexe Formen minimieren durch *Modifizieren:Kurven:Optimieren*
- ▶ längeren Text nicht »Teilen« (*Modifizieren:Teilen*) oder als Vektoren importieren
- ▶ Farbverläufe vermeiden
- ▶ wenige verschiedene Farben verwenden
- ▶ evtl. mehrere kleine Filme statt einen großen erstellen  
(bei Bedarf mit LoadMovie nachladen)

Am besten lässt man sich beim Veröffentlichen einen Größenbericht erstellen (Einstellungen für Veröffentlichen).

## Joomla!

### 29. Was ist ein CMS und welche Alternativen gibt es zu Joomla!

Mit einem **CMS** oder Content Management Systemen können Webinhalte wie Bilder und Texte sehr schnell geändert werden, auch wenn der User über keine HTML-Kenntnisse verfügt.

**Alternativen zu Joomla!:** d.n.a CMS, Typo3 CMS , plone CMS , Wordpress

### 30. Was bedeutet Open Source

**Open Source** bzw. **quelloffen** ist eine Palette von Lizenzen für Software, deren Quelltext öffentlich zugänglich ist und durch die Lizenz Weiterentwicklungen fördert.

Open-Source-Software steht unter einer von der Open Source Initiative (OSI) anerkannten Lizenz. Diese Organisation stützt sich bei ihrer Bewertung auf die Kriterien der Open Source Definition, die weit über die Verfügbarkeit des Quelltexts hinausgeht. Sie ist fast deckungsgleich mit der Definition Freier Software.

### 31. Welche Aufgaben haben PHP bzw. MySQL im Zusammenhang mit Joomla!

Der **MySQL** Server ist ein Relationales Datenbankverwaltungssystem. Es ist als Open-Source-Software für verschiedene Betriebssysteme verfügbar und bildet die Grundlage für viele dynamische Webauftritte.

**PHP** (PHP Hytertext Preprocessor) ist eine Scriptsprache die zur Erstellung von dynamischen Webseiten oder Webanwendungen verwendet wird.

PHP (rekursives Backronym für „PHP: Hypertext Preprocessor“, ursprünglich „Personal Home Page Tools“) ist eine Skriptsprache mit einer an C angelehnten Syntax, die hauptsächlich zur Erstellung von dynamischen Webseiten oder Webanwendungen verwendet wird. PHP ist Open-Source-Software. Datenbank Anwendungen werden hauptsächlich mit PHP realisiert, der Code wird auf dem Server verarbeitet.

### 32. Wann sind CMS basierte Web Sites sinnvoll

Durch die Trennung von Inhalt und Design entstehen folgende Vorteile:

- Uploaden und updaten von Texten oder Fotos ohne Programmier- und HTML-Kenntnisse seitens des Users
- Direkte und einfache Bearbeitung der Inhalte
- Gleichzeitige Verwendung durch mehrere Anwender
- Arbeitsteilung bis zum Workflow
- Einfache Erweiterbarkeit
- Trennung von Inhalt, Layout und Technik
- Einheitliche Darstellung
- Automatisierung zahlreicher Funktionen

Eine Präsentation mit Content Management System ist für die ersten Seiten aufwendiger und damit teurer. Dies gilt auch für Open Source Systeme!

Ein System benötigt eine lauffähige Umgebung, die in der Regel hohe Anforderungen an den Server stellt. Die monatlichen Kosten sind damit deutlich höher.

Ein Server, auf dem ein CMS betrieben wird, ist Änderungen unterworfen. Manche davon betreffen die Sicherheit und sind daher notwendig. Die geänderte Umgebung wird auf Dauer Änderungen des darauf laufenden Systems notwendig machen. Updates des Systems sind damit irgendwann zwingend erforderlich. Diese sind auch bei Open Source Systemen nicht kostenlos!

->Besonderen Sinn machen CMS wenn

Wesentliche Inhalte der Site sich in kurzen Zeitabständen aktualisieren und laufend erneuert werden müssen

Mehrere Redakteure am Entstehen der Site beteiligt sind:

Die wichtigsten Vorteile eines CMS dabei im Überblick:

- **Einfache Bedienung**  
Texte, Bilder und herunterladbare Dateien können ohne Vorkenntnisse auf vorhandene oder neu erstellte Seiten hinzugefügt, bearbeitet und gelöscht werden. Erfahrenen Anwendern kann darüber hinaus die Möglichkeit eingeräumt werden, mit Hilfe von HTML, Java Script, Flash o.Ä. fortgeschrittene Gestaltungselemente wie Animationen, Tabellen, etc. einzubinden.
- **Einheitliches Layout in der firmeneigenen CI (Corporate Identity)**  
Alle Seiten basieren auf einer oder mehrer Layoutvorlagen, die die Elemente, die allen Seiten gemeinsam sind, enthalten und die vor dem unbeabsichtigtem Zugriff durch den Benutzer geschützt sind.
- **Effiziente Neugestaltung bei Bedarf (z.B. geänderte CI)**  
Durch die Trennung von Inhalt und Layout kann die Gestaltung aller Seiten an zentraler Stelle erfolgen, ohne dass bereits bestehende Inhalte von Seiten geändert werden müssen. Der Aufwand beschränkt sich lediglich auf die Änderung der zentralen Vorlage und ist somit minimal.
- **Benutzung des Systems von mehreren Arbeitsplätzen**  
Die entwickelten CMS sind serverbasiert und werden über einen Webbrowser bedient; d.h. die Anwendung muss nicht auf den Arbeitsplatzrechnern installiert werden, sondern kann von jedem Computer mit Internetzugang bedient werden - entsprechende Zugangsberechtigung vorausgesetzt. Das System kann somit auch gleichzeitig von mehreren Anwendern genutzt werden.
- **Wer darf was? Die Benutzerverwaltung**  
Die Benutzerverwaltung des CMS ermöglicht es, die Berechtigungen von Anwendern detailliert zu justieren. Beispielsweise erhält ein Abteilungsleiter die Berechtigung, nur Seiten seiner Abteilung zu editieren, die PR-Abteilung hingegen kann alle Texte ändern, jedoch keine Bilder löschen oder einstellen.
- **Integrierter Workflow**  
Definieren Sie Ihren eigenen Workflow. Ein Mitarbeiter erstellt z.B. eine Pressemeldung und gibt diese zur Kontrolle frei. Der Abteilungsleiter wird automatisch hierüber informiert und schaltet die Pressemeldung auf "veröffentlichen" und macht sie damit im Internet sichtbar oder er sendet den Text mit Änderungswünschen an den Mitarbeiter zurück.
- **Zeitgesteuerte Veröffentlichung von Inhalten**  
Die von Ihnen erstellten Seiten können zeitgesteuert ins Internet gestellt und wieder herausgenommen werden. Sie können beispielsweise im Vorfeld einer Veranstaltung Pressemeldungen vorbereiten, die automatisch zu einer bestimmten Uhrzeit im Internet veröffentlicht werden. Oder ein Mitarbeiter kann Seiten vorbereiten, die während seiner Abwesenheit automatisch auf Ihrer Webseite eingestellt oder herausgenommen werden.
- **Automatisierte erstellte zentrale Seite**  
Das CMS ermöglicht die automatische Erstellung von Inhaltsverzeichnissen (Sitemaps), Übersichten

neuer oder aktualisierter Seiten, etc. Die so generierten Seiten reagieren ohne Zutun des Anwenders. Beispielsweise erscheint nach einer Änderung an der Seite "*unsere Produkte*" diese automatisch auf der Startseite unter der Rubrik "*Neu auf diesem Server*".

- **Interaktive Benutzerseiten**

Das CMS erleichtert durch seine integrierten Funktionen das Einbinden von Suchfunktionen (Volltextsuche), flexiblen E-Mail-Formularen, etc.

- **Passwortgeschützte Bereiche**

Einzelne Bereiche können vor unberechtigtem Zugriff geschützt werden. So kann beispielsweise ein Lieferant oder Außendienstmitarbeiter nur mit dem entsprechenden Passwort den aktuellen Lagerbestand überprüfen oder ein Endkunde gelangt mittels seines Passwortes auf eine spezielle Seite, welche ihm den aktuellen Stand seiner in Auftrag gegebenen Bestellung anzeigt.

- **Mehrsprachigkeit**

Sie können einen Teil oder alle Seiten in verschiedenen Sprachen anbieten. Das CMS kann so eingerichtet werden, dass dem Besucher automatisch die richtige Sprachversion der Seite angezeigt wird. Natürlich kann jederzeit zwischen den Sprachen hin- und hergeschaltet werden.

Darüber hinaus kann auch der Inhalt der Seite auf den Standort des Besuchers angepasst werden.

### 33. Nennen Sie Standard Joomla Module

Ein Modul ist im Gegensatz zu einer Komponente einfacher aufgebaut. Es ist ein Code-Fragment, das von einem anderen Programmteil eingebaut und interpretiert wird.

Module dienen der Anzeige spezieller Inhalte im Front- und im Backend. Sie können im Template positioniert werden und sind nicht Bestandteil des normalen Inhaltsbereichs. Sie werden im Modul-Manager verwaltet. Beispiele sind das Login- und das Suchformular.

#### Site-Module:

**Breadcrumbs** (Brotkrümel) : die Anzeige der Hierarchie von Beiträgen. Die einzelnen Oberbegriffe des Beitrages, der Bereich News und die Kategorie Latest sind verlinkt und bieten dadurch dem Besucher eine Orientierungs- und Navigationsmöglichkeit.

**Footer:** Dieses Modul zeigt die Joomla!-Lizenzinformationen in der Fußzeile an.

**Banner:** Dieses Modul steuert die Anzeige der Banner.

**Statistik:** ist standardmäßig deaktiviert. Um es zu aktivieren, muss man die Seiten auswählen, auf denen es angezeigt werden soll

#### Administrator-Module:

**Login Form:** Das Login-Modul stellt zwei Sichten zur Verfügung. Ist man noch nicht angemeldet, erhält man ein Login-Formular, in das man seinen Benutzernamen und sein Passwort eingeben kann. Abhängig von den Einstellungen in Site - Konfiguration - Site ist es auch möglich, sich neu zu registrieren oder nicht

**Logged in Users:** zeigt eine Liste der momentan eingeloggten Benutzer an.

**Popular:** präsentiert die Liste der am meisten besuchten Inhalte als Karteireiter im Kontrollzentrum

**Recent added Articles:** Dieses Modul zeigt eine Liste der zuletzt veröffentlichten Beiträge als Karteireiter im Kontrollzentrum an.

## Praxisteil

- Fehler in einem HTML Dokument samt eingebettetem Stylesheet finden
- Bild für eine Webseite nach Vorgabe aufbereiten
- Ausschnitt verändern
- Drehen
- Tonwertkorrektur
- Farbton/Sättigung
- Reperaturpinsel
- Einfärben von Bildteilen
- Kleinere Fotomontagen
- Einbau von Bild, Text, Link und Table auf Webseiten inkl. Formatierung mit CSS (mit Dreamweaver)

## Projekt

Das Prüfungsprojekt muss vor der Prüfungskommission präsentiert werden. Die Präsentation entspricht in Vorbereitung und Ablauf der vor realen KundInnen.

Wie in der Praxis auch gilt es die Kommission von dem Projekt zu überzeugen, die Gestaltung bzw. technische Umsetzung zu argumentieren und auf konkrete Kundenfragen zu antworten.

Die Präsentation ist Teil der Beurteilung. Darüber hinaus sind folgende Kriterien zu beachten.  
Reales Projekt - keine Selbstpräsentationen.

Zumindest ein Teil der Site MUSS mit einem CMS realisiert werden

Ziel und Zielgruppe sind klar erkennbar

Zielgerechter, effizienter und standardkonformer Einsatz der technischen Möglichkeiten (HTML, CSS, DHTML, Flash)

Klare, intuitiv bedienbare Navigation

Usergerechte Content-Aufbereitung

Nachvollziehbare Begründungen für Layout, Farbwahl, Einsatz von Bildern

Die Seiten funktionieren unter Windows auf IE und Firefox ohne Einschränkungen für die definierten Zielgruppen

Weboptimierte Bilder (Dateigröße)

Aufbereitung für Suchmaschinen (Titel, Meta-Tags)

## QUELLEN:

<http://www.wikipedia.org>

### FLASH

[http://www.on-design.de/tutor/flash\\_mx.htm](http://www.on-design.de/tutor/flash_mx.htm)

### FLASH2

[http://www.html-world.de/program/flash\\_4.php](http://www.html-world.de/program/flash_4.php)

### Fragen und Antworten zu Joomla!

<http://www.joomla-blog.net/>

### Joomla! Tutorial

<http://joomla.cocoate.com>

### Ein Joomla! Handbuch. Super!!

[http://openbook.galileocomputing.de/joomla/joomla\\_07\\_000.htm#Xxxobjoomla07000040027e51f0491c5](http://openbook.galileocomputing.de/joomla/joomla_07_000.htm#Xxxobjoomla07000040027e51f0491c5)

### HTML-Grundkurs

<http://webdesign.bhak-bludenz.ac.at/>

<http://de.selfhtml.org>

### HTML/CSS/Java-Tutorials“

<http://de.html.net/>

### Photoshop\_Tutorials:

<http://www.psd-tutorials.de/>

[http://www.designerinaction.de/tipps\\_tricks/show.php?art=photoshop](http://www.designerinaction.de/tipps_tricks/show.php?art=photoshop)

### Einfärben:

<http://www.youtube.com/watch?v=COaYKE0gcpw&NR=1>

<http://www.youtube.com/watch?v=t36Qc4Vgrlc>

<http://www.youtube.com/watch?v=93u1PLxhMNo>

<http://www.der-webdesigner.net/tutorials/photoshop/postwork/101-bilder-faerben.html>

<http://www.youtube.com/watch?v=3-T-nS6Ztys>

<http://www.youtube.com/watch?v=Ka2HE15CyhQ>

[http://www.youtube.com/watch?v=3wURbGx2soY&feature=PlayList&p=9D012A27A0CE6F05&playnext=1&playnext\\_from=PL&index=8](http://www.youtube.com/watch?v=3wURbGx2soY&feature=PlayList&p=9D012A27A0CE6F05&playnext=1&playnext_from=PL&index=8)

### Speedeffekt:

[http://www.youtube.com/watch?v=hx5TLEx1TkM&feature=PlayList&p=9D012A27A0CE6F05&index=14&playnext=7&playnext\\_from=PL](http://www.youtube.com/watch?v=hx5TLEx1TkM&feature=PlayList&p=9D012A27A0CE6F05&index=14&playnext=7&playnext_from=PL)