

# WEBT FS 2020

Victor Fernández

Februar 2020

## Inhaltsverzeichnis

<b>I</b>	<b>SW01 - Einführung</b>	<b>2</b>
<b>1</b>	<b>Inhalt</b>	<b>2</b>
1.1	Verstehen der verschiedenen Varianten von Web Applikationen . . . . .	2
1.2	Verstehen von dynamischen Webseiten . . . . .	2
1.3	Kennen der Arbeitsweise sowie der Vor- und Nachteile von nativen Applikationen, hybriden Applikationen und Web Applikationen . . . . .	2
1.4	Kennen der grundlegenden Adressierungsmechanismen . . . . .	2
<b>II</b>	<b>SW02 - HTML</b>	<b>2</b>
<b>2</b>	<b>HTML</b>	<b>3</b>
2.1	Verstehen wie HTML Informationen strukturiert und wie der Aufbau von HTML Dokumenten ist . . . . .	3
2.1.1	HTML tags . . . . .	3
2.2	Wissen wie der <b>syntaktische Aufbau</b> von HTML ist . . . . .	3
2.2.1	Struktur von Webseiten - Elemente . . . . .	6
2.3	Kennen von <b>geeigneten Werkzeugen</b> für das Erstellen, Bearbeiten, Darstellen, Validieren etc. von HTML Dokumenten . . . . .	6
2.4	Wissen um geeignete <b>Quellen und Referenzen</b> im Internet . . . . .	6

## Teil I

# SW01 - Einführung

## 1 Inhalt

- Klassifikation von Web-Applikationen
  - Statische Webseiten
  - Dynamische Webseiten
    - \* Serverseitige Erzeugung: Datenbank-orientiert
    - \* Serverseitige Erzeugung: mit Webservices
    - \* Clientseitige Erzeugung
- Formen mobiler Applikationen
- Grundlegende Adressierungskonzepte

### 1.1 Verstehen der verschiedenen Varianten von Web Applikationen

**Statische Websites** Einfache statische Webauftritte, z.B. Homepage

- Vorteile
  - Einfache Basistechnologien: HTML und CSS
  - Einfache Infrastruktur notwendig: Webserver
  - Automatische Erfassung durch Suchmaschinen
- Nachteile
  - Aktualisierungs- und Konsistenzprobleme
  - Keine Anwendungsfunktionalität

**Dynamische Websites (Inhalt dynamisch generiert)** Verschiedene Ansätze:

- Datenbankorientierte Web-Applikation
- Verwendung von Webservices
- Clientseitiges Erzeugen der dynamischen Inhalte

### 1.2 Verstehen von dynamischen Webseiten

**Konzept** Das Problem mit statischen Websites ist, dass man mit der Seite selber nicht interagieren kann. Das Ziel von dynamischen Websites ist es, eine **Benutzer-System-Interaktion** über das Web zu ermöglichen. Gewünschte Interaktionen beinhalten (nicht abschliessend):

- Suche - Ergebnis
- Eingabe - Speicherung
- Bestellung - Lieferung
- Profil Angabe - Personalisierung

**Datenbankorientiert**

- Server erstellt ein **Formular** zur Verfügung, das vom **Web-Browser angezeigt** wird
- Benutzer füllt Formular aus und drückt auf Submit Button
- Browser entnimmt die vom Benutzer eingegebenen Daten aus den entsprechenden Feldern und schickt sie an den Server zusammen mit dem Namen

### 1.3 Kennen der Arbeitsweise sowie der Vor- und Nachteile von nativen Applikationen, hybriden Applikationen und Web Applikationen

### 1.4 Kennen der grundlegenden Adressierungsmechanismen

## Teil II

# SW02 - HTML

## 2 HTML

- Das W3C (World-Wide-Web Consortium) ist für Standardisierung zuständig
- Aktuellster fertiger Standard ist Version 5.2 (Dez. 2017)
- Zudem wurden verschiedene Themen rund um HTML5 in eigenen Spezifikationen verabschiedet, bzw. sind in der Entwicklung (AAM, HTML Extension Specifications, ARIA,...)

### 2.1 Verstehen wie HTML Informationen strukturiert und wie der Aufbau von HTML Dokumenten ist

#### 2.1.1 HTML tags

### 2.2 Wissen wie der syntaktische Aufbau von HTML ist

#### <head>

- Enthält Kopfdaten wie Metainformation, Titel, Stil, Scriptdefinitionen, Adress- und Zielfensterbasis
- Ist in jedem HTML Dokument zu finden
  - Metainformationen werden durch Metatags repräsentiert

```
1 <meta charset="utf-8">
2 <meta name="author" content="Hans Wurst">
```

- Stildefinitionen (CSS - Cascaded Style Sheet) legen Darstellungen fest

```
1 <style>
2   h1 { color: white; }
3   p  { font-weight: bold; }
4 </style>
```

#### <body>

- HTML <body> kennzeichnet den Anfang und das Ende des sichtbaren Inhalts der Webseite
- Browser zeigen nur den Inhalt zwischen dem öffnenden und schliessenden body-Tag im Browserfenster
- Enthält weitere HTML Tags welche die Information strukturieren, aber auch **Scripts**, welche an der aufgeführten Stelle ausgeführt werden
- Ein HTML-Dokument darf nur einen body-Tag haben

### Text- und Informations-Strukturierung

- Absatz
- Zeilenumbruch
- Vorformatierung
- Überschriften
- Waagrechte Linien
- Container

```
1 <p>...</p>
2 <br />
3 <pre>...</pre>
4 <h1>...</h1> bis <h6>...</h6>
5 <hr />
6 <div>...</div> oder <span>...</span>
```

- Sind alles **Blockelemente**, das heisst, ein neuer Absatz (Zeilenumbruch) wird eingeleitet

### Verfügungen (Hypertext-Referenzen)

- Link

```
1 <a href="pfad/datei">Linktext</a>
```

- Sowohl lokal als auch ins Internet möglich

```
1 <a href="/index.html">Home</a>
2 <a href="http://www.hslu.ch">Gehe zu HSLU</a>
```

- Mail-Links

```
1 <a href="mailto:hans@muster.ch">Mail schreiben</a>
```

– Sollte vermieden werden, da Spambots diese automatisiert auslesen

- Interne Verknüpfung mittels Anker

```
1 <a href="#Kapitel1">Kapitel 1</a>
```

- als Ziel dieses Links

```
1 <p id="Kapitel1">Kapitel 1</p>
```

- Öffnen im neuen Fenster

```
1 <a href="adresse" target="_blank">Adresse</a>
```

## Grafiken

- Grafikformate gif, jpg, png, ...
- Einbinden mit

```
1 
```

- alt-Attribut verwenden spezielle Browser (Barrierefreiheit) oder Suchmaschinen (z.B. Google Bildersuche), unbedingt angeben
- Anzeigegröße veränderbar mit Attributen `width` und `height`
- Rahmen: `border="1px"`
- Als Hintergrund der Seite

```
1 <body background="bildname">
```

## Klickbare Grafiken: Imagemaps

### 1. Definition des Bildes

```
1 <map name="karte">
2   <area shape="circle" coords="50,50,45" href="Ziel.html" alt="Reiseziel" />
3 </map>
```

- Wird häufig zu Navigationszwecken verwendet

### Beispiel Imagemap-Code

```
1 <body>
2   ...
3   
4   <map name="transmap">
5     <area shape="rect" coords="59,390,172,441" href="#ACACIA" />
6     <area shape="rect" coords="280,21,390,63" href="#ALMOND" />
7     <area shape="rect" coords="135,52,243,124" href="#APPLE" />
8     <area shape="rect" coords="141,235,189,284" href="#ASH" />
9     <area shape="rect" coords="110,336,205,388" href="#BEECH" />
10    <area shape="rect" coords="152,289,236,334" href="#BIRCH" />
11    <area shape="rect" coords="330,231,402,288" href="#CHERRY" />
12    ...
13  </map>
14  ...
15  <a name="ACACIA">Acacia</a>
16  ...
17  <a name="ALMOND">Almond</a>
18  ...
19 </body>
```

## Metatags

- Werden nicht zur Gestaltung sondern **zur Beschreibung des Inhalts** verwendet (daher der Name: Meta-Information)
- Werden im `head` Bereich eingefügt
- Aufbau:

```
1 <meta name="Schlüsselwort" content="Inhalt">
```

- Die wichtigsten Metatags

- keywords
- description
- language
- page-topic (Thema für Suchmaschinen und Kataloge)
- audience (Zielgruppe in Suchmaschinen und Katalogen)
- robots (zur Linkverfolgung)
- refresh (und expires → Ablaufdatum)
- copyright

### Sonderzeichen

- Damit Sonderzeichen korrekt dargestellt werden, muss das charset metatag korrekt gesetzt sein

```
1 <meta charset="utf8">
2 <meta charset="iso-8859-1">
```

- Eine Alternative ist die Zeichen speziell zu kodieren:  
Beispiel: aus **ü** wird **&uuml;**;
- Dies ist auch notwendig bei Zeichen, welche mit dem HTML-Markup kollidieren:  
& zu &amp;, < zu &lt;, > zu &gt;;

### Masseinheiten

- Verwendet in Attributen zur Bestimmung der Dimensionen verschiedenerer Elemente wie Bilder, Schriften, Ränder, Abstände
- Die wichtigsten Einheiten sind:
  - pt: Punkt, **absolute** Angabe, 1 Punkt entspricht 1/72 Inches
  - in: Inch, **absolute** Angabe, 1 Inch entspricht 2.54cm
  - mm: Millimeter, **absolute** Angabe
  - px: Pixel, **absolute/relative** Angabe Abhängig von der **Pixeldichte** des Ausgabegeräts
  - em: M, **relative** Angabe, auf die Schriftgrösse des Elements bezogen (mit Ausnahmen)
  - %: Prozent, **relative** Angabe, je nach CSS-Eigenschaft relativ zur elementeigenen Grösse, oder zu der des Elternelements, oder zu einem allgemeineren Kontext

### Farben

- Farben werden aus den RGB-Wertangaben gebildet
- Beispiel: #FF0000 ist rot, #00FF00 ist grün, #0000FF ist blau
- Werte werden in hexadezimaler Form angegeben
- Werte von #00 bis #FF (255) sind möglich
- Einige Farben sind per Namen in der DTD (Document-Type-Definition) definiert:

Black	= #000000	Green	= #008000
Silver	= #C0C0C0	Lime	= #00FF00
Gray	= #808080	Olive	= #808000
White	= #FFFFFF	Yellow	= #FFFF00
Maroon	= #800000	Navy	= #000080
Red	= #FF0000	Blue	= #0000FF
Purple	= #800080	Teal	= #008080
Fuchsia	= #FF00FF	Aqua	= #00FFFF

### 2.2.1 Struktur von Webseiten - Elemente

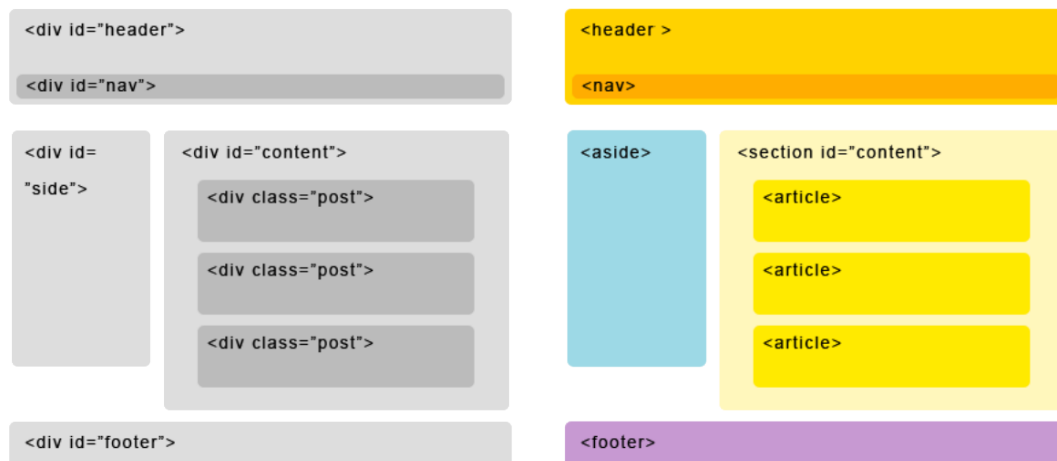


Abbildung 1: Strukturbeispiel von Webseiten

#### <header>

- enthält sichtbaren Kopfbereich einer Webseite
- Gruppierung einleitender Inhalte (Firmenlogos, Motto, Navigationslinks)

#### <footer>

- enthält Informationen, die in Webseiten am Ende stehen: Autor, Hinweise zum Urheberrecht, ein Link zum Impressum
- Position ist nicht notwendigerweise am unteren Rand  
→ bei Blogbeiträgen steht der footer oft neben dem Text

#### <article>

- stellt in sich geschlossene Abschnitte eines Dokuments dar  
→ vergleichbar mit einem Zeitungsartikel  
→ innerhalb von article-Elementen weitere strukturierende Elemente wie header, section oder footer

#### s<section>

- enthält eine thematische Gruppierung von Inhalten typischerweise mit einer Überschrift
- dient dazu, den Inhalt oder auch einen article in semantische Abschnitte zu gliedern

#### <nav>

- umschließt insbesondere Navigationsleisten
- kann neben einer ungeordneten Liste mit den Verweisen auch eine Überschrift oder ähnliches enthalten

#### <aside>

- umschließt Abschnitte einer Seite, deren Inhalt nur in einem indirekten Zusammenhang mit dem umgebenden Inhalt stehen
- Beispiele: Randbemerkungen, Fussnoten oder Links zu weitergehenden Webseiten

## 2.3 Kennen von geeigneten Werkzeugen für das Erstellen, Bearbeiten, Darstellen, Validieren etc. von HTML Dokumenten

## 2.4 Wissen um geeignete Quellen und Referenzen im Internet