

Einführung in Web Development und Shopware Entwicklung

Block 1 / Teil 1

Olga Khorkova

Sportspar GmbH

13. August 2019

Über den Kurs

- Anfänger-Niveau
- Aufgeteilt in 2/(?)3 Blöcke
- Jeder Block besteht aus 5 Teilen
- Jeder Block sowie jeder Teil werden mit einem Projekt abgeschlossen
- Projekte sowie die Vertiefungsthemen werden gemeinsam abgestimmt
- Projekte werden reviewed.

Über den 1. Block

- Web-Architektur
- HTML 5
- CSS 3
- PHP/MySQL
- JavaScript
- Responsivness
- SEO, Accessibility, valider Code
- Architektur von Shopware 5
- GIT
- **Abschluss-Projekt B1: Ein einfaches Shopware-Theme erstellen**

Über den 2. Block

- Vertiefung in CSS
- Einführung in Preprozessoren (SASS)
- Einführung in CSS-Frameworks (Bootstrap)
- Vertiefung in JS
- JS-Frameworks (jQuery)
- Web-Animationen
- Databases
- Vertiefung in PHP
- Objekt-Orientierende Programmierung
- **Abschluss-Projekt B2: ?**

Generelle Empfehlungen

- Es gibt keine doofen Fragen.
- Google is your friend!
- Macht den Kurs bei Udemy: *Einführung in WebDev*
- Practice, practice, and practice

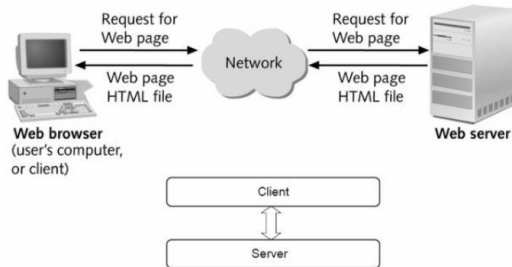
Inhalte des 1. Teils

- 1 Web-Architektur
- 2 HTML
- 3 CSS
- 4 Web Page
- 5 Zusammenfassung
- 6 Aufgabe
- 7 Nützliche Links

Wie funktioniert Web?

Generell: web besteht aus zwei Schichten (*tiers*)

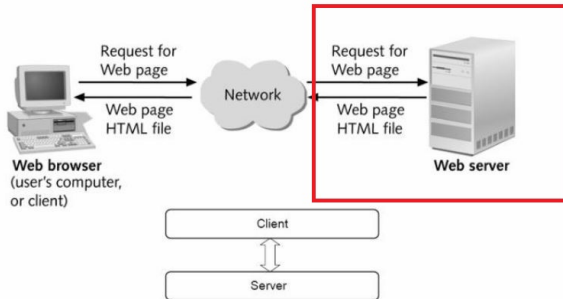
- Web Browser: präsentiert die Informationen → **Client**
- Web Server: liefert die Informationen dem Client → **Server**



Web Server

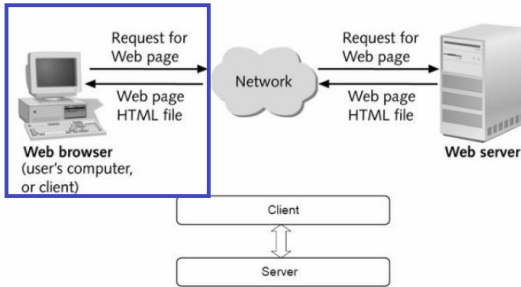
Unter dem **Web Server** wird folgendes verstanden:

- Ein Programm, das HTTP *Requests* akzeptiert und HTTP *Responses* (200, 400 etc.) zurückgibt, wobei *data content* optional ist
- Ein Rechner, der das oben beschriebene Programm ausführt



Client-Server-Modell

- Protokoll: **http**
- URL: **www.sportspar.de**
- Repräsentation: **HTML und CSS** ← Thema des 1. Blocks



HTML

HTML

HTML Basics 1

- es ist keine Programmierspache!!!
- Hyper Text Markup Language

HTML Basics 2

- Markup: Darstellen und Positionieren der Elemente auf der Seite. Damit werden Dokumente erzeugt. → **Darstellung**
- Mit Programmiersprachen werden Scripts oder Programme erstellt, also das, was ausführbar ist. → **Ausführung**

index.html

index.html ist Root-File einer Webseite

Tags Syntax 3 Regeln

1. `<irgendeintag>Inhalt</irgendeintag>`

Der Tag hat immer ein Anfang - *öffnendes tag* und ein Ende - *schliessendes tag*

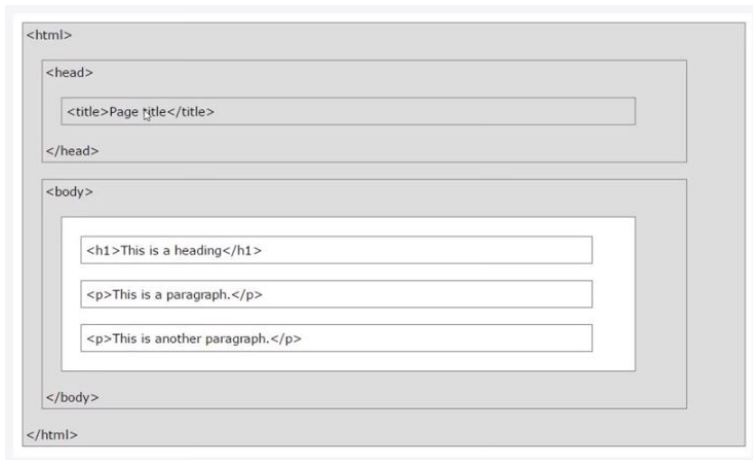
2. `<u>sehr wichtiger Inhalt</u>`

Die Tags dürfen ineinadergeschachtelt werden.

3. `<u>sehr wichtiger</u> Inhalt `

Die Tags sollen nicht über Kreuz geschrieben werden.

visuelle Struktur HTML-Seite



Struktur HTML-Seite

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```


Tags Attribute 1

```
<irgendeintag attributename="attributevalue">  
Inhalt</irgendeintag>
```

- Alle Tags können ein Attribut haben
- Attribute liefern die Informationen über das Element
- Sie sind in dem öffnenden Tag platziert

Tags Attribute 2

```
<p class="paragraph">Inhalt</p>
```

Der gleiche *Class* kann mehreren Elementen zugewiesen werden.

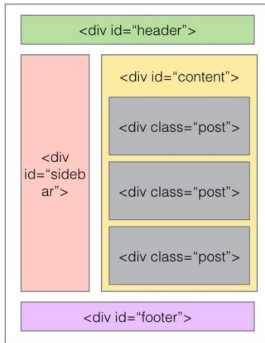
```
<p id="paragraph-special">Inhalt</p>
```

ID darf nur ein Element haben

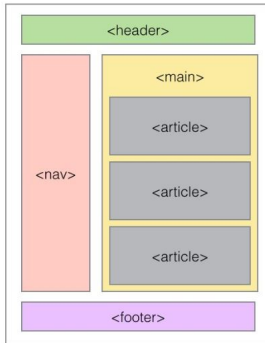
Semantik

Semantische HTML Tags liefern eindeutige Bedeutung der Blöcke sowohl dem Browser als auch dem Developer. Außerdem vereinfacht das das Crawlen der Seite → **SEO**

HTML4: Lots of Classes/IDs



HTML5: Semantic Tags/Sections



Semantik

```
<header></header>  
<nav></nav>  
<main></main>  
<section></section>  
<article></article>  
<aside></aside>  
<footer></footer>
```

Images

img ist ein non-binary Tag, d.h. es gibt keinen schliessenden Tag

```

```

Links

```
<a href="https://www.sportspar.de"></a>
```

CSS

CSS

Add CSS

- Inline CSS: direkt in dem HTML-Dokument → **NO**
- Internal CSS: `<style>` Tag direkt in dem Document
- External CSS: link zu einem css File

Basics

```
body {  
  margin-left: 100px;  
  margin-right: 100px;  
  background-color: grey;  
  font-family: "Raleway";  
  font-size: 16px;  
}
```

Selectoren

Selectoren: verweisen auf Element, das zu stylen ist.

- tag `h`, `p`, `body`
- class `.test-class`
- id `.test-id`

Manche Eigenschaften

- margin
- padding
- color
- font-size

Kurzschreibweise der Eigenschaften

Anstatt einzelne Werte für `margin-top`, `margin-right`, `margin-bottom` oder `margin-left` (bzw. `padding-...`) anzugeben kann man alle Werte mit in der einen CSS-Eigenschaft `margin` (bzw. `padding`) zusammen.

Die Reihenfolge erfolgt immer im Uhrzeigersinn, angefangen wird beim ...-top Wert.

Die folgenden Schreibweisen sind identisch:

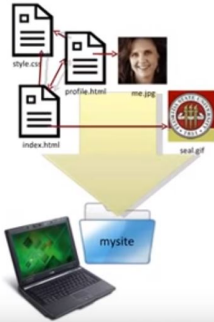
```
1 margin-top: 30px;  
2 margin-right: 10px;  
3 margin-bottom: 40px;  
4 margin-left: 20px;
```

und

```
1 margin: 30px 10px 40px 20px;
```

Web Page 1

- bestehen aus mehreren Files .html, .css, .jpg etc.
- sind in einem Ordner organisiert



Web Page 2

- miteinander verbunden mittels HTML → links zu den relativen Positionen
- Aufgabe von Browser: alle diese Files miteinander zu verbinden und die Instruktionen aus dem HTML/CSS zu entnehmen, um zu determinieren, wie man die Inhalte richtig presenteren muss



Zusammenfassung

- Client-Server-Model
- HTML: strukturelle Blöcke des Webs
- CSS: styling der Blöcke
- Web Page besteht aus mehreren Files (.html, .css, .jpeg), die auf dem Server liegen und dem Client nach dem Request zur Verfügung gestellt werden.

Aufgabe

Es ist eine Aufgabe zu wählen (Struktur und ein wenig CSS Wireframe):

- Personal Page
- Layout von Netflix Landing Page nachzubilden

Tutorials

- Brad Traversy HTML5 <https://www.youtube.com/watch?reload=9&v=UB1030fR-EE>
- Brad Traversy CSS3 <https://www.youtube.com/watch?v=yfoY53QXEnI>
- w3schools HTML5 <https://www.w3schools.com/html/default.asp>
- w3schools CSS3 <https://www.w3schools.com/css/default.asp>
- HTML Handbook <https://www.freecodecamp.org/news/the-html-handbook/>

Links

- Validator: <https://www.w3.org/>
- Editor: VS Code <https://code.visualstudio.com/>