

Templates - Bauanleitung für ein Seitenlayout

Über den Aufbau eines Layouts mit HTML und CSS.

Tem
plat

Einrichten der Head Sektion

Die Sprachangabe

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      lang="de-DE">
  <head>
    <meta charset="utf-8">
```

Das lang-Attribut stellt die Sprache für Inhalte und Attribute-Werte des Dokumentes ein. Die Sprache kann für einzelne Elemente überschrieben werden.

| | |
|---------------------------|-------------------|
| <p>erster Absatz</p> | -> deutsch |
| <p lang="fr">Deuxième</p> | -> französisch |
| <p>dritter Absatz</p> | -> wieder deutsch |

Die Zeichensatzangabe

```
<!DOCTYPE html>  
<html lang="de-DE">  
  <head>  
    <meta charset="utf-8">
```

UTF - UCS Transformation Format

UCS - Universal Character Set

[Wikipedia:] UTF-8 ist in den ersten 128 Zeichen (Indizes 0–127) deckungsgleich mit ASCII und eignet sich mit i.d.R. nur einem Byte Speicherbedarf für Zeichen vieler westlicher Sprachen besonders für die Kodierung englischsprachiger Texte, die sich im Regelfall ohne Modifikation daher sogar mit nicht-UTF-8-fähigen Texteditoren ohne Beeinträchtigung bearbeiten lassen, was einen der Gründe für den Status als De-facto-Standard-Zeichenkodierung des Internets und damit verbundener Dokumenttypen darstellt. ...

<title> - Der Dokumententitel

```
<!DOCTYPE html>
<html lang="de-DE">
  <head>
    <meta charset="utf-8">
    <title>Titel des Dokumentes - Herkunft</
title>
```

Kürzeste Identifikation eines Dokuments für
Suchmaschinenlistings, Bookmarks, Fenstertitel;
SEO-relevant!

Den Viewport für mobile Geräte einrichten

```
<head>  
  <meta charset="utf-8">  
  <title>Titel des Dokumentes</title>  
  <meta name="viewport"  
    content="width=device-width,  
            initial-scale=1.0">
```

Die obige Zeile stellt die Darstellung auf Bildschirmbreite ein, dabei wird die Darstellungsgröße um einen Gerätefaktor vergrößert (iPhone = 2,0), der für eine bessere Darstellungsgröße und Lesbarkeit sorgt. Weitere Werte: min-scale, max-scale, user-scalability.

Dies ist derzeit ausschließlich nur für mobile Geräten wirksam und wird von Desktopbrowsern ignoriert.

Den Viewport für Featurephones und Windowsphones einrichten

```
<meta name="HandheldFriendly" content="true">
```

```
<meta name="MobileOptimized" content="320">
```

Um bei Featurephones anzuzeigen, dass die Seite für 320px optimiert wurde, wird "HandheldFriendly", für Windowsphones das das "MobileOptimized" verwendet.

Icons für den Browser und mobile Geräte

```
<link rel="apple-touch-icon-precomposed" sizes="144x144"  
      href="src/ico/apple-touch-icon-144-precomposed.png" />  
  
<link rel="apple-touch-icon-precomposed" sizes="114x114"  
      href="src/ico/apple-touch-icon-114-precomposed.png" />  
  
<link rel="apple-touch-icon-precomposed" sizes="72x72"  
      href="src/ico/apple-touch-icon-72-precomposed.png" />  
  
<link rel="apple-touch-icon-precomposed"  
      href="src/ico/apple-touch-icon-57-precomposed.png" />  
  
<link rel="shortcut icon" href="src/ico/favicon.png" />
```

apple-touch-icons werden von iOS verwendet, um ein Icon auf der iPhoneoberfläche als Bookmark zur Seite/App anzulegen. Die verschiedenen Auflösungen berücksichtigen die Displayvarianten.

Metaelemente als HTTP-Kopfdaten-Äquivalent

```
<meta http-equiv="Content-Type"
      content="text/html; charset=utf-8" />
```

Sie enthalten Informationen, die in Kopfzeilen der HTTP-Kommunikation vom Server zum Client übertragen werden. Die Daten können bereits vor dem Erzeugen der HTTP-Antwort vom Webserver gelesen und in die Kopfzeilen eingebaut werden. Eine Information im HTTP-Header wird einem Meta-Element vorgezogen. Angabe des MIME-Type (im Beispiel text/html) und der verwendeten Zeichenkodierung (im Beispiel UTF-8) des Dokuments.

Metaelemente als HTTP-Kopfdaten-Äquivalent

```
<meta http-equiv="refresh"
      content="5;
      URL=http://www.example.org/" />
```

Weiterleitung durch „refresh“

Leitet beim Aufruf einer Seite zu einer anderen URL um. Das content-Attribut enthält eine Zeitangabe für die Weiterleitung, z. B. um den Anwender noch auf den Grund für die Weiterleitung hinzuweisen.

Metaelemente für Webcrawler I

```
<meta name="robots" content="index, follow" />  
oder <meta name="robots" content="all" />
```

Der Webcrawler (Suchroboter) soll die Webseite ausdrücklich in den Suchmaschinen-Index aufnehmen, damit sie auffindbar ist, und den Hyperlinks auf der Seite folgen. Diese Angabe ist im Allgemeinen unnötig, da sie dem normalen Verhalten der meisten Webcrawler entspricht.

Metaelemente für Webcrawler I

```
<meta name="robots" content="noindex" />
```

Der Suchroboter soll die Webseite nicht in den Index aufnehmen. Die Webseite soll also nicht über Suchmaschinen auffindbar sein. Siehe auch: Robots Exclusion Standard.

```
<meta name="robots" content="nofollow" />
```

Der Suchroboter darf die Webseite zwar aufnehmen, aber soll den Hyperlinks auf der Seite nicht folgen. Die Wirkung ist so, als wären die Links auf der betreffenden Seite gar nicht vorhanden. Derselbe Effekt lässt sich auch für einzelne Links erreichen, indem diese um das Attribut `rel="nofollow"` ergänzt werden. (siehe `nofollow`)

Metaelemente für Webcrawler II

```
<meta name="robots" content="noodp" />
```

Die Suchmaschine soll nicht die Beschreibungstexte des Open Directory Projects verwenden – sofern die Seite dort eingetragen ist –, sondern einen Ausschnitt der Seite selbst. Diese Anweisung wird von Google und MSN Search unterstützt.

```
<meta name="robots" content="noydir" />
```

Die Suchmaschine soll anstelle der Beschreibungstexte aus dem Yahoo-Verzeichnis einen Ausschnitt der Seite selbst. Diese Anweisung wird nur von Yahoo (Robot: Slurp) unterstützt.

Metaelemente für Suchmaschinen (oder auch nicht mehr)

Kurze Beschreibung des Inhalts der Webseite

```
<meta name="description"
      content="Lorem ipsum dolor sit amet,
              consetetur sadipscing elitr." />
```

Stichwörter

Nennt Stichwörter oder Themen, die auf der Webseite behandelt werden. Komma-getrennte Liste.

```
<meta name="keywords"
      content="Stichwort,Stichwort2,Stichwort3" />
```

Autor: Name des Autors und inhaltlich Verantwortlichen.

```
<meta name="author"
      content="Autorenname" />
```

Verlinken externer Stylesheets

```
<link href="src/css/default.css"  
      rel="stylesheet" />
```

```
</head>
```

Externe Stylesheets bieten der Vorteil der Wiederverwendbarkeit und Modularität.

Sie werden im Kopfteil der HTML eingebunden, damit sie möglichst frühzeitig zur Verfügung stehen.

Der Kopfbereich eines HTML Dokumentes

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
    <title>Titel des Dokumentes</title>
    <meta name="viewport"
      content="width=device-width,
        initial-scale=1.0">

    <meta name="HandheldFriendly" content="true">
    <meta name="MobileOptimized" content="320">
    <link rel="apple-touch-icon-precomposed"
      sizes="144x144" href="src/ico/apple-touch-
        icon-144-precomposed.png" />

    <link rel="icon" href="src/ico/favicon.png" />

    <meta name="description" content="...">
    <meta name="author" content="...">
    <meta name="robots" content="index, follow">
    ...
    <!-- 2 - Einbinden der Stylesheets -->
    <link href="src/css/default.css"
      rel="stylesheet">
  </head>
```


Browserkompatibilität herstellen

Codesnippets, die helfen, ein HTML für alle Browser zu schreiben.

Gerätespezifische Einstellungen

```
<meta name="apple-mobile-web-app-capable"  
      content="yes">
```

Adressleiste und Toolbar in Safari verschwinden.

```
<meta name="apple-mobile-web-status-bar-style"  
      content="black">
```

Farbe der Statuszeile wird auf Schwarz gesetzt.

```
<link rel="apple-touch-start-image"  
      href="img/splash.png">
```

Ein Bild als Startbild laden und anzeigen.

```
<meta http-equiv="cleartype"  
      content="on">
```

Sorgt für eine bessere Schriftdarstellung
im mobilen Internet Explorer.

Javascript zum Fixen von IE Fehlern von Version 5.5 - 8

```
<!--[if lte IE 8]>  
    <script type=„text/javascript“ charset=„utf-8“  
        src=„http://ie7-js.googlecode.com/svn/  
            version/2.0(beta3)/IE8.js“></script>  
<![endif]-->
```

Conditional Comments!

Das bedingte HTML ist eine Internet Explorer eigene Angelegenheit. Sie wird von alle anderen Browsern ignoriert.

IE spezifische Anpassungen für Versionen < 9

```
<!-- HTML5 shim, for IE6-8 support of HTML5 elements -->  
<!--[if lt IE 9]>  
    <script src="lib/js/html5shiv.js"></script>  
<![endif]-->
```

vorher: <section></section> -> section {display:inline-block}

-> passiert nix!

-> createElement('section'); // Jetzt geht es!

Hier kommt zunächst das eigene CSS ...

```
<!-- CSS for IE6-8 -->  
<!--[if lt IE 9]>  
    <link rel="stylesheet" href="ie-color.css" />  
<![endif]-->
```

Reset setzt das Aussehen von HTML zurück auf Null.

Reset.css

Setzt alle browser-eigenen Eigenschaften der HTML Elemente auf null.

Damit ist es möglich, ein für alle Browser kompatibles CSS-basiertes Grundlayout zu gestalten.

```
<link rel="stylesheet" type="text/css"
      media="all"
      href="lib/css/reset.css" />
```

Normalize egalisiert das Aussehen von HTML

Normalize.css

Erzeugt ein einheitliches Erscheinungsbild für alle Browser.

Normalize wurde von Nicolas Gallagher geschrieben und beständig weiter entwickelt.

```
<link rel="stylesheet" type="text/css"
      media="all"
      href="lib/css/normalize.css" />
```

Die Body Sektion

Templateebene - Layoutbereiche

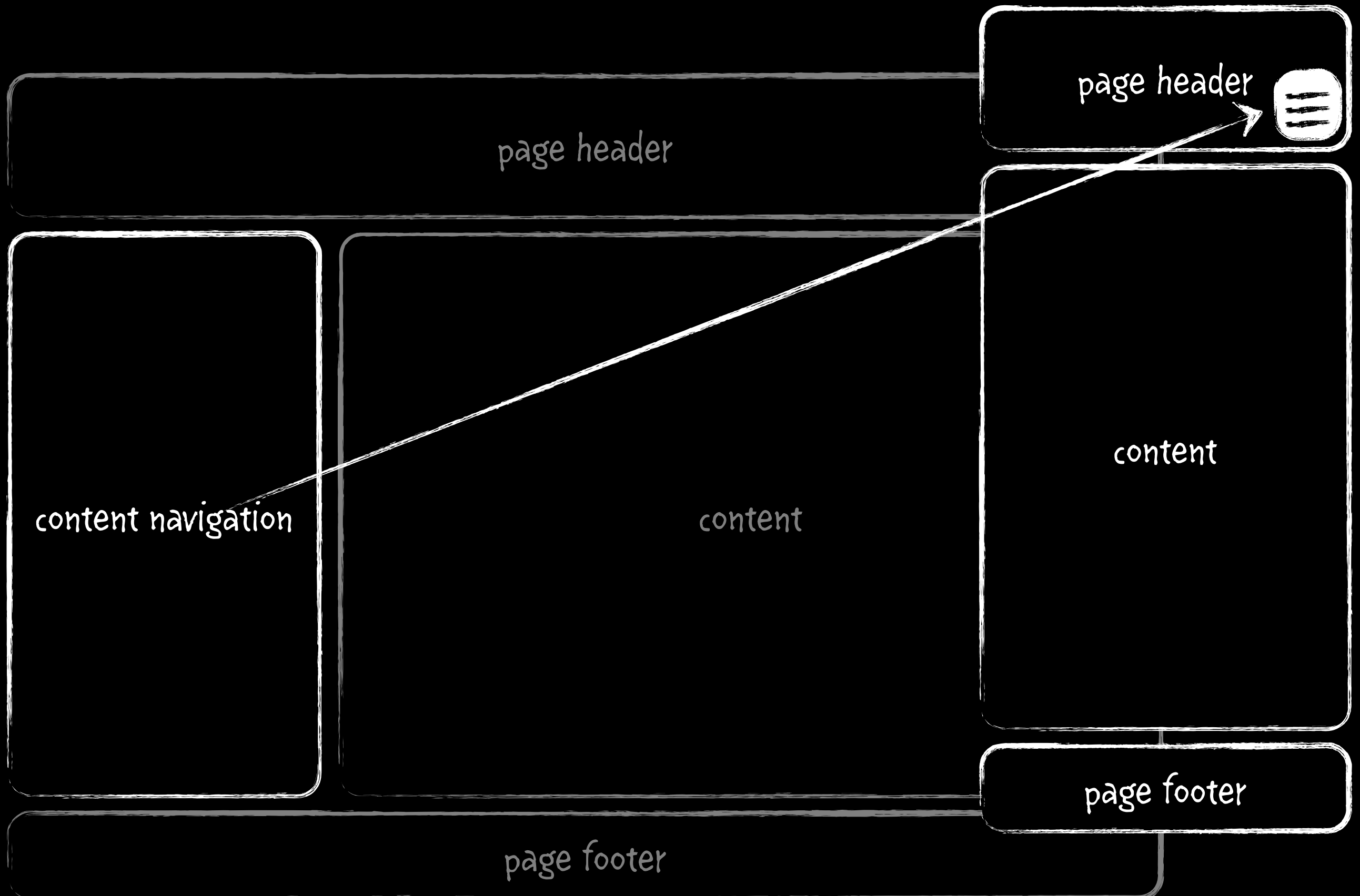
page header

content navigation

content

page footer

Für mobile Geräte ändert sich oft nur das Aussehen der Elemente



Aufbau von Layoutelementen

```
<body>
```

```
    <header class="row"> ... </header>
```

```
    <div class="row">
```

```
        <div class="span3">...</div>
```

```
        <div class="span9">...</div>
```

```
    </div>
```

```
    <footer class="row">...</footer>
```

```
</body>
```

```
</html>
```

Id's einsetzen

```
<div id="wrapper"> ... </div>
```

Ein Id-Attribut gibt einem HTML Element eine Identität. Ein Id-Wert muss innerhalb einer Seite unique bleiben.

Id's funktionieren als 'Hooks' in das Dokument und werden von Selektoren in Javascript oder CSS verwendet.

Javascript:

```
document.getElementById( 'wrapper' );
```

CSS:

```
div#wrapper {  
    some styles ...  
}
```

Klassenattribute verwenden

```
<div class="span3"> ... </div>
```

Ein Klassen-Attribut verbindet ein HTML Element mit einer Style-Anweisung in einem Stylesheet. In Gegensatz zur ID ist eine Klasse mehrfach verwendbar.

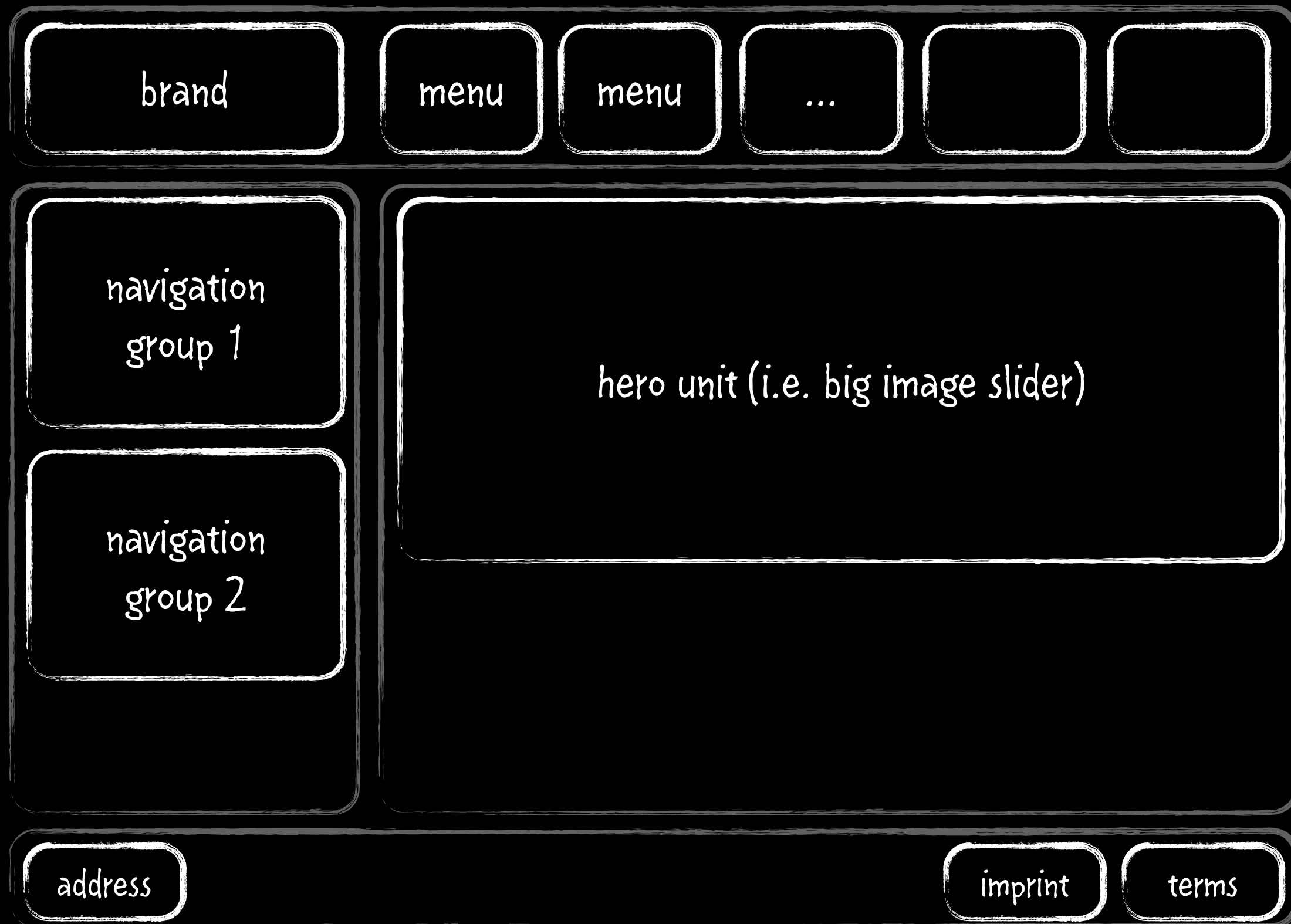
Klassen werden in der Stylesheet-Datei mit Klassenselektoren geschrieben.

Javascript:

```
document.getElementsByClassName( 'span3' );
```

CSS: .span3 { some styles ... }

Templateebene - Bausteine und Gruppen der Standardinhalte



Aufbau von Layoutelementen

```
<body>
  <header>
    <div class="brand">...</div>
    <nav>...</nav>
  </header>

  <div class="content-wrapper">
    <div class="span3">
      <nav>...</nav>
    </div>

    <div class="span9">
      <section id="hero">...</section>
      <section id="content"> ...</section>
    </div>
  </div>

  <footer>
    <ul>...</ul>
  </footer>
</body>
</html>
```

Den Viewport einstellen

**View
port**

Viewport

Breite und Höhe eines Layouts

Die Breite und Höhe eines Layouts hängt von der Bildschirm Auflösung, der Fenstergröße und den Anzeigen Bereich ab. Diese sind je nach Gerät sehr unterschiedlich.

Als Viewport wird der Bereich eines Layouts bezeichnet, der ohne Scrollen sofort sichtbar ist.

Im Desktopbrowser ist das der Weißbereich, auf einem mobilen Gerät meist der ganze Bildschirm.

Einstellung des Viewports

Für mobile Geräte wurde zusätzlich eine Möglichkeit entwickelt, den Viewport, also die Grundeigenschaften des Gerätefensters zu beeinflussen. Hierbei geht es zunächst um eine Skalierung des Browserinhaltes auf eine der Bildschirmgröße angemessene Vergrößerung;

Mobile Geräte besitzen eine Pixel Density, die die Anzahl der Pixel pro Inch definiert, Retina Display haben zum Beispiel eine doppelt so hohe Pixeldichte, daher muss der Skalierfaktor um einiges höher sein.

```
<meta name="viewport" content="width=device-width" />
```

```
<meta name="viewport"  
      content="width=device-width,  
              initial-scale=1,  
              minimum-scale=1,  
              maximum-scale=1,  
              user-scalable=no" />
```

Simulation eines Viewports

Steht, wie bei Blackberry 6 Browsern der Viewport Tag nicht zur Verfügung, so kann per CSS mit der Vergrößerung der Basisschriftgröße und relativen Layoutelementen dasselbe erreicht werden. In Verbindung mit MediaQueries durchaus sinnvoll.

```
/* LAYOUT CSS */

body, body * {
    font-size : 32px; /* 16px wären Standard ... */
}

header, section#content, footer {
    width      : 100%;
    padding    : 0.5em; /* em sind abhängig
                        von der Basisschriftgröße*/
}
```

Für Featurephones und Windowsphones

Um bei Featurephones anzuzeigen, dass die Seite für 320px optimiert wurde, wird "HandheldFriendly", für Windowsphones das das "MobileOptimized" verwendet.

```
<meta name="HandheldFriendly"
      content="true">
<meta name="MobileOptimized"
      content="320">
```

HTML Komponenten

Mo
dule

Der Komponentengedanke

Schreiben Sie definiertes, wiederverwendbares HTML. Viele Seitenelemente lassen sich mehrfach verwenden.

Jedes Komponente besitzt eine "Root"-Gruppe. Diese besitzt eine ID oder eine Basisklasse, über die sie selektiert werden kann.

Trennen Sie semantische Gruppen von Layoutgruppen. Das ist flexibler.

Führen Sie einen Komponentenkatalog, aus dem Codes kopiert werden können.

Beispiel Navigationskomponente

```
<div>
  <nav id="nav-main" class="nav nav-block">
    <ul>
      <li><a href="..">Linktext</a></li>
      <li><a href="..">Linktext</a></li>
      <li><a href="..">Linktext</a></li>
    </ul>
  </nav>
</div>
```

```
<div>
<nav id="nav-main" class="nav nav-block">
  <ul>
    <li><a href="..">Linktext</a></li>
    <li><a href="..">Linktext</a></li>
    <li><a href="..">Linktext</a></li>
  </ul>
</nav>
</div>
```

Layoutgruppe

Semantische Rootgruppe

Komponentencode

```
<div id="nav-main" class="nav nav-block">
<nav>
  <header>Hauptnavigation</header>
  <ul>
    <li><a href="#">Linktext</a></li>
    <li><a href="#">Linktext</a></li>
    <li><a href="#">Linktext</a></li>
  </ul>
</nav>
</div>
```

Optionale Komponenten-
ergänzung

Beispiel Formularelement

```
<div class="page-main">
<form id="login" class="form form-columnar">

    <div class="form-element-group">
        <label for="input-email">E-Mail</label>
        <input id="input-email" type="email" ... >
        <p class="help">Hilfetext für das Element</p>
    </div>

    ...

</form>
</div>
```

Formularelement als Liste

```
<div class="page-main">
<form id="login" class="form form-columnar">

    <ul>
        <li class="form-element-group">
            <label for="input-email">E-Mail</label>
            <input id="input-email" type="email" ... >
            <p class="help">Hilfetext für das Element</p>
        </li>

        ...
    </ul>

</form>
</div>
```