

Version 2023c

# Grundlagen der Web-Programmierung

# Cascading Style

# Sheets Level 3

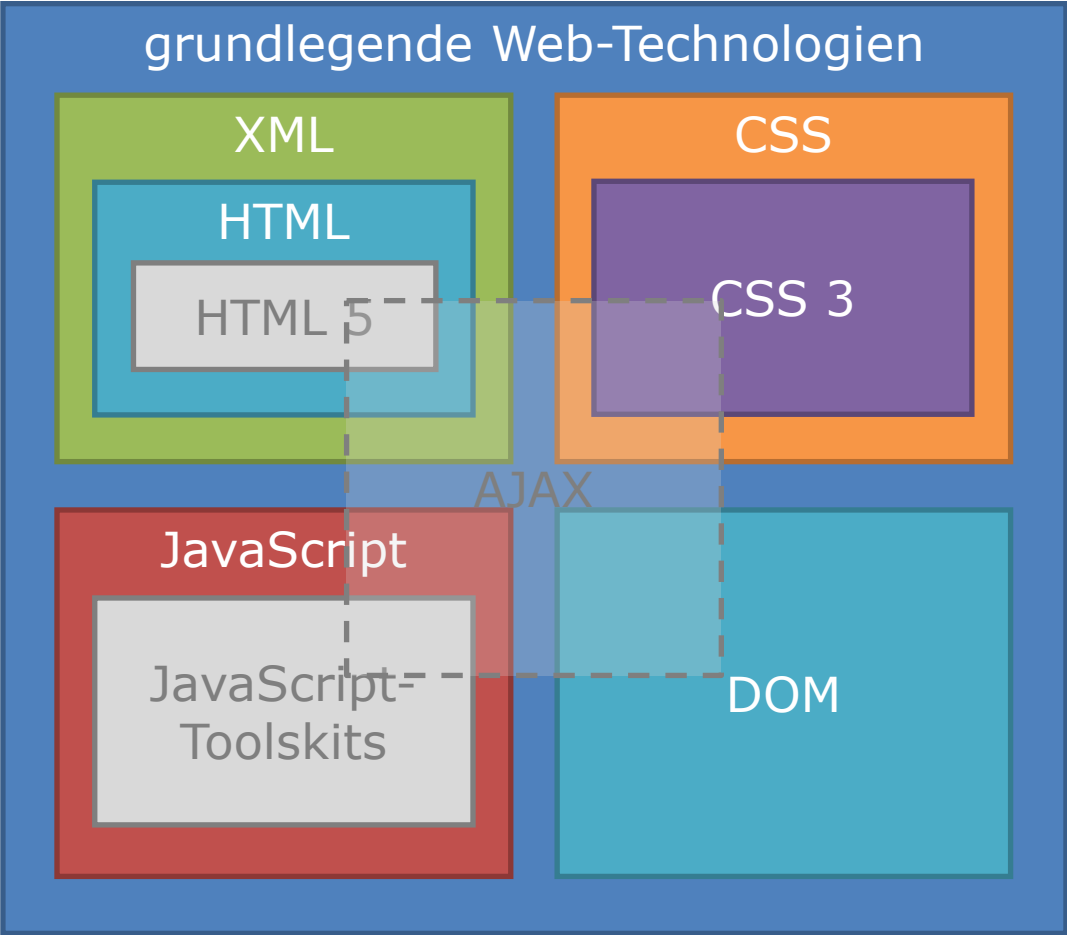
**Prof. Dr.-Ing. Tenshi Hara**  
tenshi.hara@ba-sachsen.de



# AUFBAU DER LEHRVERANSTALTUNG

Progressive Web Apps

Search Engine Optimization



Proprietäre Standards

# VERSIONIERUNG UND STANDARDISIERUNG

- aktuell standardisierte Version von CSS: Level 2 Revision 1 (CSS 2.1)
- CSS 3 liegt als Entwurf vor, der aber nicht standardisiert wurde
  - wird ständig weiter entwickelt
  - keine klaren Inkremente
  - einzelne Aspekte von CSS (3) sind oder werden ggf. standardisiert
    - CSS Color Level 3
    - CSS Namespaces
    - CSS Selectors Level 3
    - CSS Media Queries
- nach aktueller Planung wird es kein CSS 4 geben [<http://www.xanthir.com/b4Ko0>]

# BEDARF UND ANWENDUNGSGEBIET

- **Heterogenitätsproblem**

Web-Anwendungen sind ohne **Adaptivität** nicht adäquat auf unterschiedlichen Zielgeräten darstellbar

- **Bedienbarkeitsproblem**

unterschiedliche Bedienkonzepte benötigen unterschiedliche Darstellungs- und Interaktionsformen, bspw. Touch vs. Click

- **Anpassungsproblem**

Web-Anwendungen ohne **Responsivität** können nicht auf sich ändernde Umgebungsbedingungen adaptieren, bspw. nach Drehung des Geräts

- **Individualisierungsproblem**

zur stärkeren Nutzerbindung müssen sich Web-Anwendungen an ihre Nutzer anpassen oder Individualisierungsmöglichkeiten geben

# Neuerungen in CSS 3

# LÄNGENANGABEN

- rem** Größe relativ zur Schriftgröße der obersten DOM-Ebene  
(z.B.: Schriftgröße 12px → 1.5rem entsprechen dann 18px)
- cap** Größe relativ zur Höhe des Majuskel-O (Kapitalhöhe) der  
ersten verfügbaren CSS3-Webfont
- ch** Größe relativ zur Breite der 0 in der aktuellen Schriftart  
(falls der Wert nicht berechnet werden kann: 1ch = 0.5em)
- ic** Größe relativ zur Breite des CJK-Symbols 水
- lh** Größe relativ zur Zeilenhöhe des aktuellen Elementes
- rlh** Größe relativ zur Zeilenhöhe der obersten DOM-Ebene

# LÄNGENANGABEN FÜR MOBILE BROWSER

- vh** (**vw**) relativ zu 1% der vertikalen Höhe (horizontalen Breite) des Inhaltsbereichs
- svh** (**svw**) bezogen auf den reduzierten Inhaltsbereich sobald Adress-, Status- und Bildlaufleisten eingeblendet sind
- lvh** (**lvw**) bezogen auf den maximalen Inhaltsbereich sobald Adress-, Status- und Bildlaufleisten ausgeblendet sind
- dvh** (**dvw**) dynamisch auf die tatsächliche Größe des Inhaltsbereichs bezogen (z.B. wenn nur einzelne Leiste eingeblendet)

# SCHRIFTLINIEN

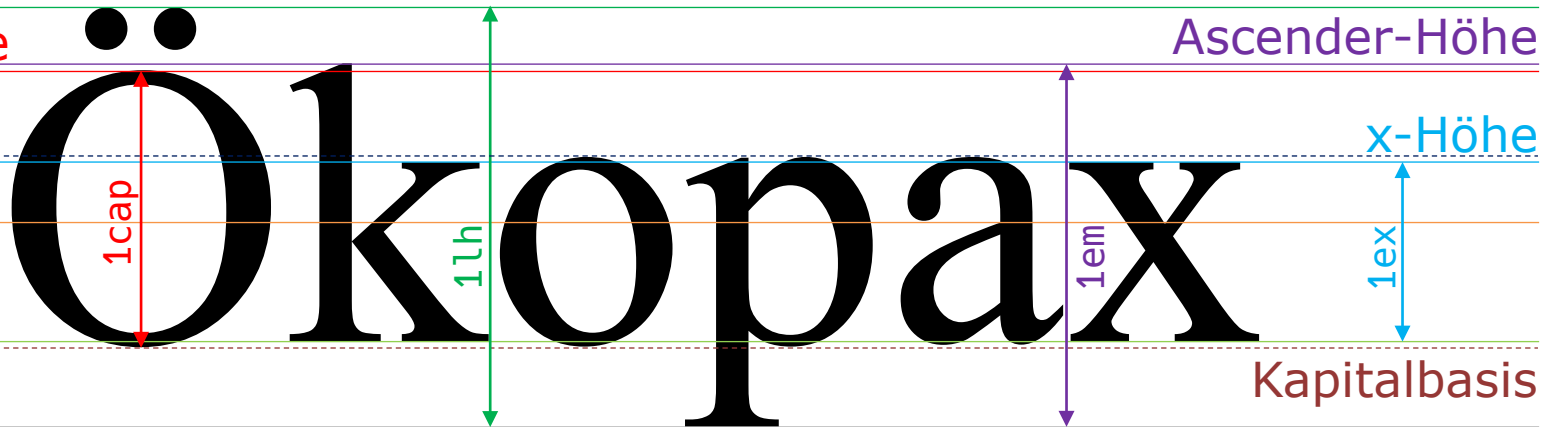
Akzenthöhe

Kapitalhöhe

Minuskel-  
höhe

Median

Grundlinie



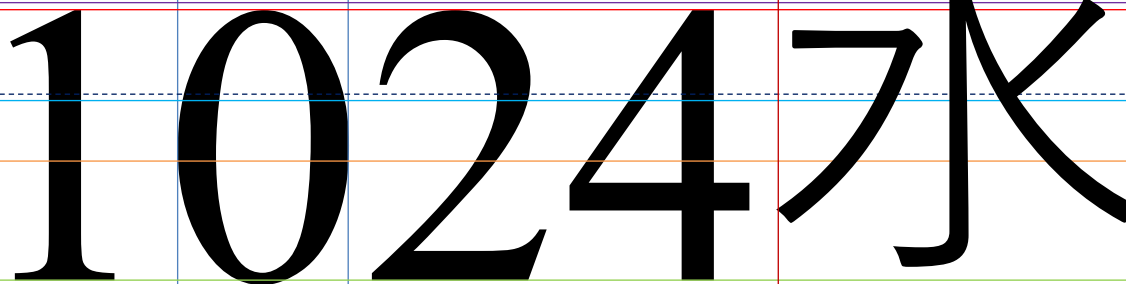
Descender-Linie

0-Breite

1ch

水-Breite

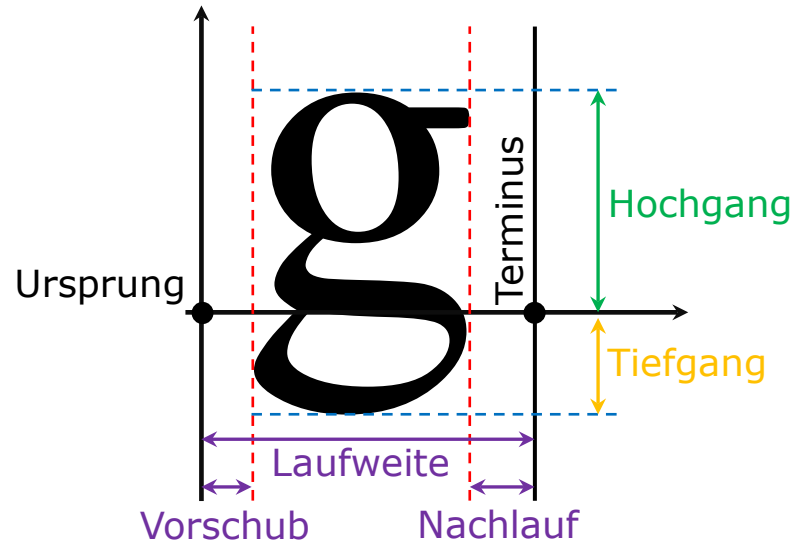
1ic



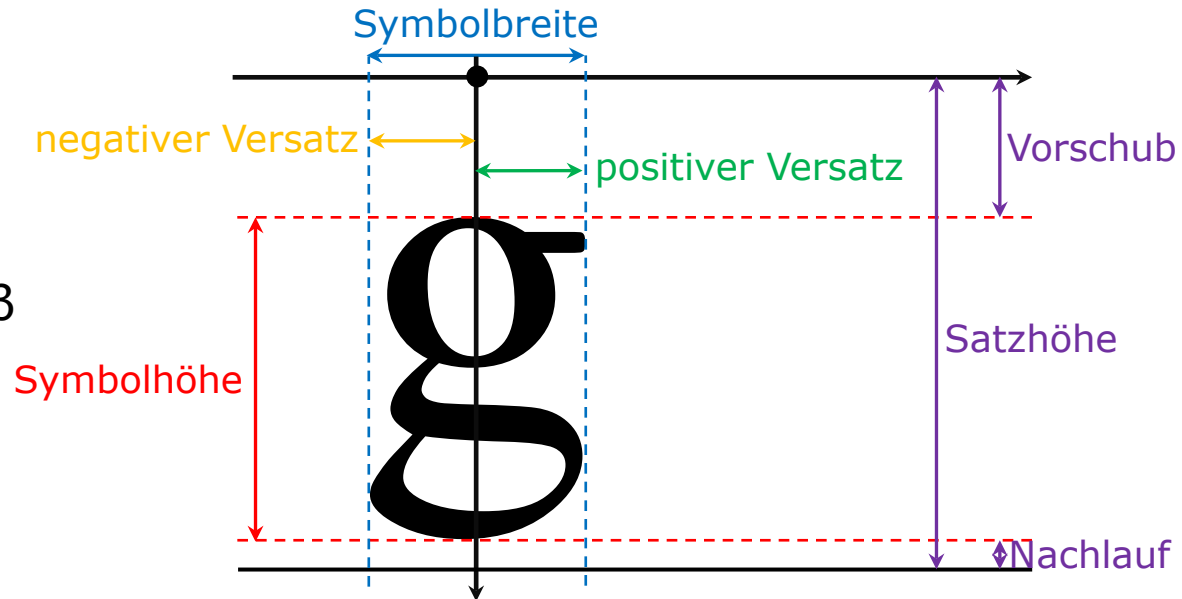


# BERECHNUNGSGRUNDLAGE: SCHRIFT- UND SYMBOLMETRIK

Schriftmaß



Symbolmaß



# SELEKTOREN, KOMBINATOREN UND PSEUDOKLASSEN

| Selektor     | Auswahl  |
|--------------|--|
| [attr^=val]  | Elemente, deren Attribut attr den Wertpräfix val hat |
| [attr\$=val] | Elemente, deren Attribut attr den Wertsuffix val hat |
| [attr*=val]  | Elemente, deren Attribut attr den Wertinfix val hat  |

| Kombinator | Auswahl  |
|------------|--|
| P ~ Q      | Selektor Q auf gleicher DOM-Ebene wie Selektor P |

| Pseudoklasse | Auswahl   |
|--------------|---|
| :target      | Element, welches Ziel des gerade Geklickten ist |
| :enabled     | auswählbare Elemente der GUI                    |
| :disabled    | gesperrte Elemente der GUI                      |
| :checked     | ausgewählte/angehakte Elemente der GUI          |
| :not(Q)      | Elemente, für die Selektor Q nicht gilt         |

# STRUKTURELLE PSEUDOKLASSEN

| Pseudoklasse         | Auswahl  |
|----------------------|--|
| :first-child         | erstes Kind eines Elements   |
| :last-child          | letztes Kind eines Elements  |
| :nth-child(Q)        | Q-tes Kind eines Elements  |
| :nth-last-child(Q)   | Qt-letztes Kind eines Elements                                     |
| :only-child          | Elemente, die Einzelkind eines Elements sind                       |
| :first-of-type       | erster Nachfahre eines Typs eines Elements                         |
| :last-of-type        | letzter Nachfahre eines Typs eines Elements                        |
| :nth-of-type(Q)      | Q-ter Nachfahre eines Typs eines Elements                          |
| :nth-last-of-type(Q) | Qt-letzter Nachfahre eines Typs eines Elementes                    |
| :only-of-type        | Elemente, die der einzige Nachfahre eines Typs eines Elements sind |
| :root                | Wurzelelement  |
| :empty               | kinderlose Elemente  |

## CSS COLOR LEVEL 3

neben hexadezimalen RGB-Farben sind jetzt auch andere Formate möglich

- **RGBA** – mit Alpha-Kanal zur Unterstützung von Transparenz
  - Angabe der Transparenz in Prozent (50%) oder einsgenormt (.5)
- **HSL** – Farbton, Sättigung, Helligkeit
  - Farbton: aus dem Farbwinkelraum; Integer aus dem Bereich (0, 359)
    - 0 Rot
    - 60 Gelb
    - 120 Grün
    - 180 Cyan
    - 240 Blau
    - 300 Magenta
  - Sättigung und Helligkeit als Prozentwert oder einsgenormt
- **HSLA** – wie HSL, aber mit Transparenz
- **Opacity** – Deckkraft; ähnlich Transparenz, aber beeinflusst alle Darstellungseigenschaften sowie alle Kindelemente

# CONTAINER DESIGN (1/2)

- abgerundete Ecken mittels `border-radius` und genannten Maßeinheiten
  - `border-radius: 25px;`  
alle Ecken mit Rundungen von 25-Pixel Radius
  - `border-radius: 5px .2cm 2pt 3em;`  
unterschiedliche Radien für die einzelnen Ecken (RULO; RBLT)
- mittels `-shadow` können Elemente Schattenwurf gestalten
  - `text-shadow: top0ffset left0ffset blurRadius color;`
  - `box-shadow: 5pt 10pt darkgreen;`
  - ...

## CONTAINER DESIGN (2/2)

- Stapeln von Hintergrundbildern (visuell vorne nach hinten)
  - `background-image:`  
`url(near.gif), url(mid.png), url(far.jpg);`
  - Transparenz und Animationen der Bilder werden berücksichtigt
  - Deklaration weiterer Eigenschaften in der gleichen Reihenfolge
    - `background-position:` `right bottom`, `absmid`, `left top`;
    - `background-repeat:` `no-repeat`, `no-repeat`, `repeat`;
- Gestaltung von Gradienten
  - `background:` `linear-gradient(to right, red, green);`
  - `background:` `radial-gradient(circle, blue, cyan, green);`

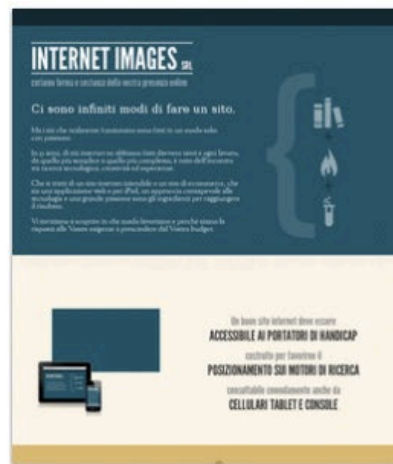
## CSS MEDIA QUERIES (1/3)

Abfrage von Medien-Eigenschaften wie Breite/Höhe des Browserfensters, Breite/Höhe des Bildschirms, Quer-/Hochformat, Bildschirmauflösung, ...

```
#content {  
    width: 800px;  
}  
  
@media screen and (max-width: 1024px) {  
    #content {  
        width: 600px;  
    }  
}  
  
@media screen and (max-width: 720px) {  
    #content {  
        width: 540px;  
    }  
}
```

## CSS MEDIA QUERIES (2/3)

- basierend auf Browser-Attributen werden unterschiedliche CSS-Anweisungen interpretiert → multiple CSS für das gleiche Element
- hauptsächlich: Attribute in Abhängigkeit zur Bildschirmgröße
  - Definition multipler CSS-Layouts auf dem selben HTML-Dokument
  - Definition von Umbrüchen für unterschiedliche Layouts
  - Zeigen/Verbergen/Skalieren/Verschieben von Elementen in Abhängigkeit vom ausgewählten Layout



[Internet Images SRL, <https://interim.it>]  
weitere Beispiele: <https://mediaqueri.es>



# CSS MEDIA QUERIES (3/3)

- Medientypen
  - all|aural|braille|handheld|**print**|projection|**screen**|tv
  - schränken die Anwendbarkeit von CSS-Eigenschaften ein
    - `<link rel="stylesheet" media="only screen and (color)" href="example.css" />` → Laden der CSS
    - `@media screen and (max-width: 600px) { ... }` → innerhalb der CSS
- beliebige Kombination aus einzelnen Medien, verketteten Medien und Operatoren
  - `@media screen and (min-device-width: 480px) and (orientation: landscape)`
  - `@media screen and (max-width: 1200px) and (min-resolution: 260dpi) and (aspect-ratio: 1/1)`
- Eigenschaften physischer und logischer Natur
  - width|min-width|max-width|height|min-height|...
  - device-width|min-device-width|max-device-width|device-height|...
  - aspect-ratio|min-aspect-ratio|max-aspect-ratio
  - resolution|min-resolution|max-resolution
  - orientation|...

# Anwendungen

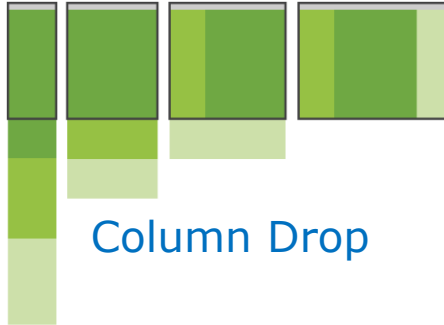
# MEHR-GERÄTE-LAYOUT-MUSTER

Erzeugung auf Basis der Prinzipien des responsiven Web-Designs

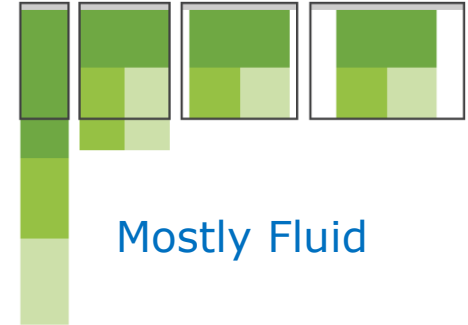
[Wroblewski; <http://www.lukew.com/ff/entry.asp?1514>]



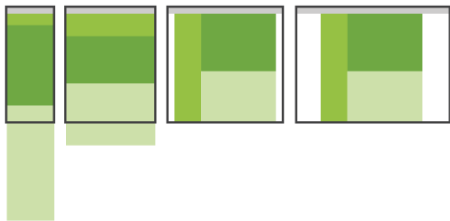
Tiny Tweaks



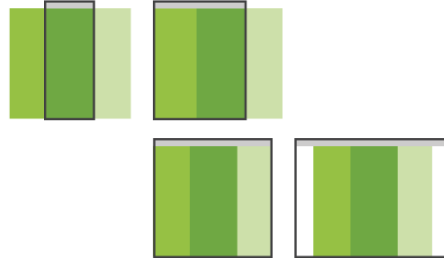
Column Drop



Mostly Fluid



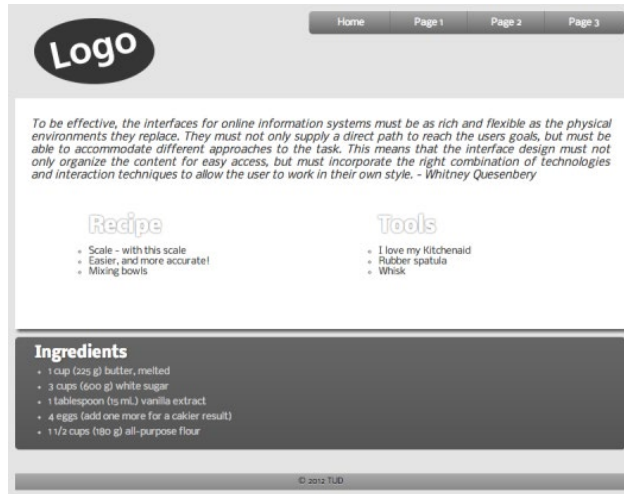
Layout Shifter



Off Canvas



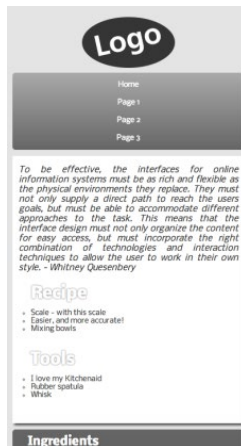
# BEISPIEL – MOSTLY FLUID



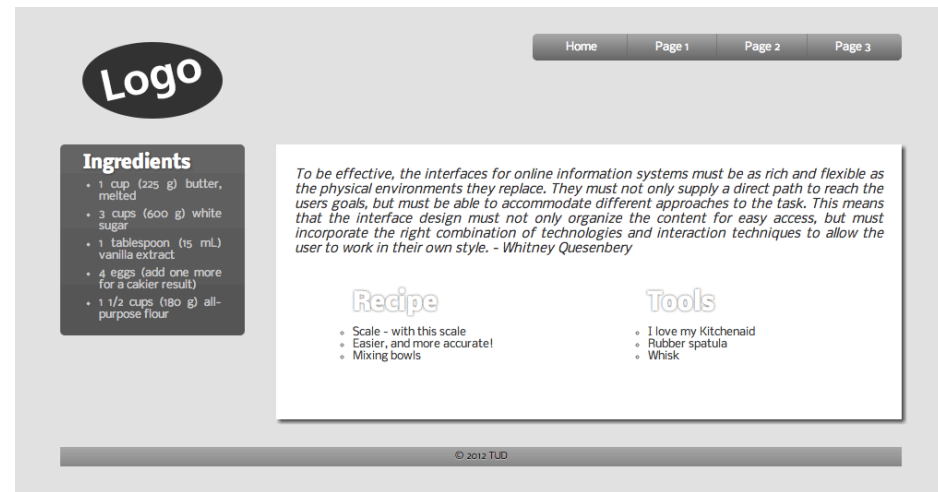
1024px – alter PC-Bildschirm



480px – Low-Budget-Smartphone



1280px – Smartphone (Portrait-Modus)



1920px – HD-Desktop

# BEISPIEL – MOSTLY FLUID

- Mobile first – Basis-CSS (480px)

```
body {margin: 10px 0;}
```

```
header, nav, article, footer {
  width: 100%;
}
```

```
nav ul li {
  width: 25%; /* 4 Elemente */
}
```

```
article div.list div {
  float: left; width: 100%;
}
```



- 2. Media Query (Smartphone, 720px+)

```
@media (handheld and min-width: 720px) {
  nav ul li {
    float: left;
    width: 100%;
  }
}
```

```
article div.list div {
  float: left;
  width: 100%;
}
```



- 1. Media Query (1024px)

```
@media (min-width: 1024px) {
  body { margin: 20px 0; }
```

```
header {
  float: left;
}
```

```
nav {
  float: right;
  width: auto;
}
```

```
}
```



- 3. Media Query (1920px+)

```
@media (min-width: 1920px) {
  article {
    float: right;
    width: 75%;
  }
}
```

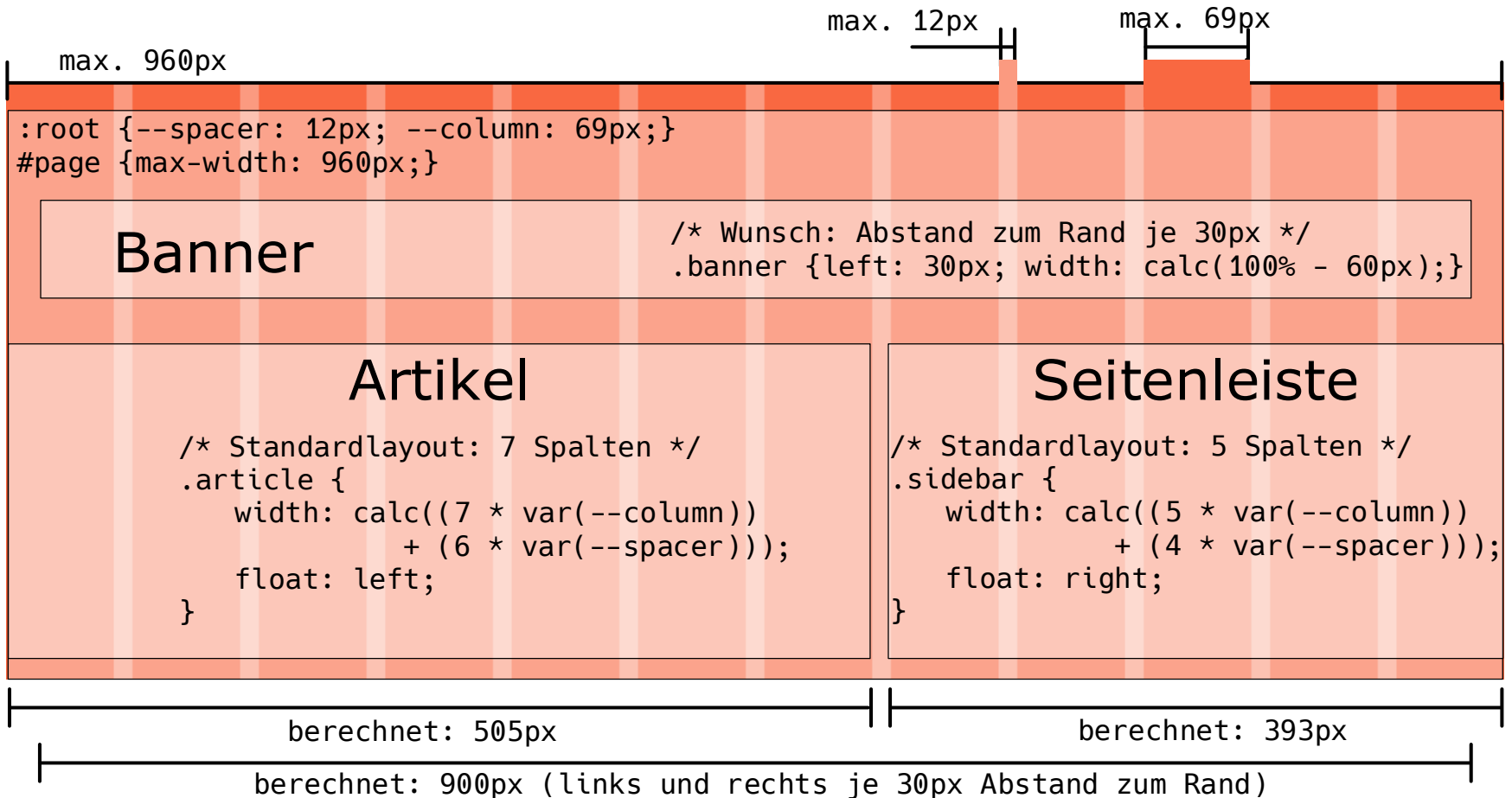
```
aside {
  float: left;
  width: 25%;
}
```

```
}
```



# GRID-BASED

- Seiten-Layout basierend auf Raster
  - Raster mit Spalten fester Breite und festen Abständen
  - Elemente mit berechneter Breite auf dem Raster



# WEITERE NEUERUNGEN

- Übergänge und Animationen
  - zeitlich abgepasste Übergänge zwischen CSS-Styles
  - Animationen: komplexe Übergänge zwischen multiplen Styles
- Web Fonts
  - Einbettung von Schriftarten aus dem Web via @font-face
  - Embedded Open Type (.eot), TrueType (.ttf), OpenType (.otf)

# CSS-BASIERTE INFORMATIONSOBFUSKATION (CSSIO)

- verwendet, um Bots aus dem Tritt zu bringen
- Ziel: für Menschen sichtbare/verborgene/angepasste Information
- gehen in der Regel zu Lasten der Barrierefreiheit
- einfach zu kompilieren, visuelles Ergebnis sehr schwer semantisch dekompilierbar
- Beispiel: Obfuskation der E-Mail-Adresse

HTML-Kode:

```
<span class="mail">te</class><span class="rev mail">h.ihsn</span><span class="mail">a</class><span class="mailer mail">&nbsp;</span>a<span class="rev-mb rev mail">mail@</span><span class="rev mail">d-ab</span><span class="rev-dom rev mail"></span><span class="mailer mail">&nbsp;</span><span class="rev mail">ed.nedse</span>
```

CSS-Kode:

```
span.mail{font-style:italic;}span.mailer{margin-left:-.6ch;}span.mailer:after{content:"\0072";}span.rev{unicode-bidi:bidirectional-override;direction:rtl;}span.rev-dom:before{content:"\0040";}span.rev-mb{visibility:hidden;display:none;}
```

Ergebnis: