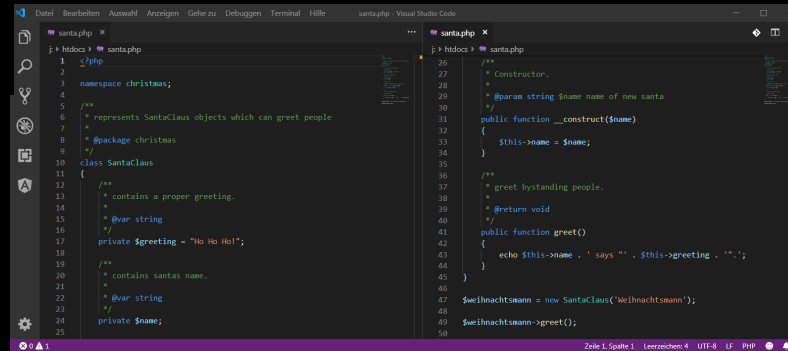


Crashkurs Webprogrammierung



The screenshot shows two editor windows in Visual Studio Code. The left window displays a PHP file named 'santaphp' with the following code:

```
1 <?php
2
3 namespace christmas;
4
5 /**
6  * represents SantaClaus objects which can greet people
7  */
8 @package christmas
9 */
10 class SantaClaus
11 {
12     /**
13      * contains a proper greeting.
14      */
15     @var string
16     private $greeting = "Ho Ho Ho!";
17
18     /**
19      * contains santas name.
20      */
21     @var string
22     private $name;
23
24     private $name;
25 }
```

The right window displays the same file with the following code:

```
26
27 /**
28  * Constructor.
29  */
30 @param string $name name of new santa
31 */
32 public function __construct($name)
33 {
34     $this->name = $name;
35 }
36
37 /**
38  * greet bystanding people.
39  */
40 @return void
41 */
42 public function greet()
43 {
44     echo $this->name . ' says ' . $this->greeting . ' ' . ' ';
45 }
46
47 $weihnachtsmann = new SantaClaus("Weihnachtsmann");
48 $weihnachtsmann->greet();
49
50
```

CSS: Cascading Style Sheets



Warum eine eigene Sprache statt weiteren HTML-Attributen?



Separation of Concerns

- HTML, CSS und JavaScript sind je für ihren eigenen Bereich zuständig, so überschneidungsfrei wie möglich
- Vorteile:
 - Quelltexte bleiben jeweils übersichtlich
 - So wenig Einschränkungen wie möglich durch spezialisierte Sprachen
 - Effiziente Umsetzung von Ideen
 - Strukturiertes Vorgehen
 - *progressive enhancement & graceful degradation*

Wie sieht CSS-Syntax aus?



CSS: Syntax – ein Beispiel

```
<h2 id="sprungmarke">Überschrift</h2>
<p>Hier steht wichtiger Inhalt, denn 4 < 5.</p>
<hr />
<p class="footnote">Dieser Inhalt ist eine Fußnote.</p>
```

```
/* Überschriften sollten unterstrichen sein. */
h1, h2, h3, h4, h5, h6 {
  text-decoration: underline;
}

p {
  font-family: 'Courier New', Courier, monospace;
}

hr {
  border: 2px dashed red;
}

.footnote {
  font-size: 0.8rem;
}

[id]::before {
  content: "\1f517 ";
  text-decoration: none;
}

/* nur beim drüberfahren mit der Maus */
[id]:hover::after {
  content: " #"attr(id);
  color: red;
  text-decoration: none;
}
```

Überschrift

Hier steht wichtiger Inhalt, denn 4 < 5.

Dieser Inhalt ist eine Fußnote.

Überschrift

Hier steht wichtiger Inhalt, denn 4 < 5.

Dieser Inhalt ist eine Fußnote.

CSS: Syntax – Selektoren



<https://wiki.selfhtml.org/wiki/CSS/Selektoren>

```
/* Überschriften sollten unterstrichen sein. */
h1, h2, h3, h4, h5, h6 {
    text-decoration: underline;
}

p {
    font-family: 'Courier New', Courier, monospace;
}

hr {
    border: 2px dashed red;
}

.footnote {
    font-size: 0.8rem;
}

[id]::before {
    content: "\1f517 ";
    text-decoration: none;
}

/* nur beim drüberfahren mit der Maus */
[id]:hover::after {
    content: " #"attr(id);
    color: red;
    text-decoration: none;
}
```

Überschrift

Hier steht wichtiger Inhalt, denn 4 < 5.

Dieser Inhalt ist eine Fußnote.

CSS-Dokumente bestehen aus
Selektoren und Kombinatoren,
Deklarationen und **Kommentaren**.

Kommentare bestehen aus Text, sind
gegebenenfalls mehrzeilig und werden in
/* und */ eingeschlossen. Anweisungen
darin werden nicht ausgeführt.

Zu jedem Set von Deklarationen, das in
geschweifte Klammern eingeschlossen
wird, gehört einer oder mehrere durch
Komma voneinander getrennte
Selektoren. Die Deklarationen werden
dann auf die Elemente angewendet, auf
die der jeweilige **Selektor** zutrifft.

CSS: Syntax – Wichtige Selektoren



<https://wiki.selfhtml.org/wiki/CSS/Selektoren>

```
/* Überschriften sollten unterstrichen sein. */
h1, h2, h3, h4, h5, h6 {
    text-decoration: underline;
}

p {
    font-family: 'Courier New', Courier, monospace;
}

hr {
    border: 2px dashed red;
}

.footnote {
    font-size: 0.8rem;
}

[id]::before {
    content: "\1f517 ";
    text-decoration: none;
}

/* nur beim drüberfahren mit der Maus */
[id]:hover::after {
    content: " #"attr(id);
    color: red;
    text-decoration: none;
}
```

Selektoren (Auswahl):

*	-	alle Elemente
h1	-	alle h1-Elemente
.class	-	alle Elemente mit Klasse class
#id	-	Element mit id: id
[attr]	-	alle Elemente mit Attribut attr
[attr=x]	-	alle Elemente mit Attributwert x

Kombinatoren (A und B sind Selektoren):

A B	-	alle B innerhalb A
A > B	-	alle B direkt unterhalb von A
A ~ B	-	alle B, die auf ein A folgen
A + B	-	alle B, die direkt auf A folgen

CSS: Syntax – Wichtige Selektoren



<https://wiki.selfhtml.org/wiki/CSS/Selektoren>

```
/* Überschriften sollten unterstrichen sein. */
h1, h2, h3, h4, h5, h6 {
    text-decoration: underline;
}

p {
    font-family: 'Courier New', Courier, monospace;
}

hr {
    border: 2px dashed red;
}

.footnote {
    font-size: 0.8rem;
}

[id]::before {
    content: "\1f517 ";
    text-decoration: none;
}

/* nur beim drüberfahren mit der Maus */
[id]:hover::after {
    content: " #"attr(id);
    color: red;
    text-decoration: none;
}
```

Pseudoelemente:

Eigentlich nicht vorhandene (Teil-)Elemente

::first-line	-	erste Zeile (Absatz o.ä.)
::first-letter	-	erster Buchstabe
::before/::after	-	dynamischer Inhalt
::selection	-	markierter Text o.ä.
::placeholder	-	Platzhalter in Formularen

Pseudoklassen (Auswahl):

„Filter“ für selektierte Elemente

:empty	-	leere Elemente
:first-child	-	Kind-Elemente, die „erste“ sind
:last-child	-	Kind-Elemente, die „letzte“ sind
:nth-child(x)	-	x-te Kind-Elemente
:first-of-type / :last-of-type / :nth-of-type(x) ...		
:link	-	unbesuchte Links
:visited	-	besuchte Links
:hover	-	beim Drüberfahren mit Maus
:active/:focus	-	aktive Elemente / mit Fokus
:target	-	„angesprungenes“ Element
:disabled / :enabled / :checked	-	Formularstatus
:lang(x)	-	Inhalt in Sprache x
:not(A)	-	was von A nicht selektiert wird
:matches(A, B)	-	Gruppierung von Selektoren

CSS: Syntax – Eigenschaften



<https://wiki.selfhtml.org/wiki/CSS/Eigenschaften>

```
/* Überschriften sollten unterstrichen sein. */
h1, h2, h3, h4, h5, h6 {
    text-decoration: underline;
}

p {
    font-family: 'Courier New', Courier, monospace;
}

hr {
    border: 2px dashed red;
}

.footnote {
    font-size: 0.8rem;
}

[id]::before {
    content: "\1f517 ";
    text-decoration: none;
}

/* nur beim drüberfahren mit der Maus */
[id]:hover::after {
    content: " #"attr(id);
    color: red;
    text-decoration: none;
}
```

Überschrift

Hier steht wichtiger Inhalt, denn 4 < 5.

Dieser Inhalt ist eine Fußnote.

In CSS-Deklarationen werden verschiedenen **Eigenschaften** **Werte** zugewiesen.

Je nach **Eigenschaft** sind verschiedene Arten von **Werten** (Maßeinheiten, Farbwerte, Schlüsselworte, ...) möglich; teilweise auch die Kombination mehrerer Werte.

Welche Eigenschaften gibt es?



CSS-Eigenschaften: Schrift und Text

font-family: Liste, von, Schriftarten;	-	Schriftart für den beinhalteten Text
font-style: italic/oblique/normal;	-	Kursivdruck
font-weight: lighter/bold/bolder/Zahl;	-	Fettdruck
font-variant: normal/small-caps;	-	Kapitälchen
line-height: Zahl/Längenangabe;	-	Zeilenhöhe
color: Farbangabe	-	Schriftfarbe
letter-spacing/word-spacing: Längenangabe	-	Abstand Zeichen / Wörter
text-decoration: underline/overline/line-through/...	-	z.B. Unterstreichung
text-shadow: offset-x offset-y blur-radius color	-	Textschatten
text-transform: uppercase / lowercase / capitalize	-	Klein- und Großbuchstaben
text-indent: Längenangabe	-	Einrückung
vertical-align: middle / bottom / top /...	-	vertikale Ausrichtung von Text
columns: Spaltenzahl Mindestbreite	-	mehrspaltige Darstellung

...und einiges mehr

CSS-Eigenschaften: Gestaltung

Äußere Gestaltung:

border: Dicke Art Farbe - Rahmen, Art z.B. dotted/dashed/solid/inset ...
border-top / border-right / border-bottom / border-left - Einzelne Rahmentteile
border-radius: oben rechts unten links - Radius für abgerundete Rahmenecken
outline: Dicke Art Farbe - Rahmen *außerhalb* des Objekts

box-shadow: offset-x offset-y blur-radius spread color - Schlagschatten

background-color: Farbangabe - Hintergrundfarbe
background-image: url("../Pfad_Hintergrundbild.jpg") - Hintergrundbild
background-repeat / background-size / background-clip ... - Feinjustierung
background-image: linear-gradient(...) - Farbverläufe

Drehung etc:

transform: translate(x)/scale(x)/rotate(x)/skew(x)/perspektive(x) ...

CSS-Eigenschaften: Anzeige & Positionierung



<https://wiki.selfhtml.org/wiki/CSS/Eigenschaften>

position: absolute / relative / ...
top / right / bottom / left / : Längenangabe...

- Positionierungsart
- Angaben zur Position

float: left / right

- Umfließen von Elementen

display: block / inline / inline-block / none

visibility: visible / hidden / ...

opacity: Zahl (zwischen 0 und 1)

clip: ... / overflow: ...

- Anzeigeart, Eigenschaften
- Sichtbarkeit
- Transparenz
- Abschneiden / Ausschnitt

Flexbox - Moderne und flexible Layouts - <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Flexbox>

Grid Layout – Raster-Anordnung - <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/grid>

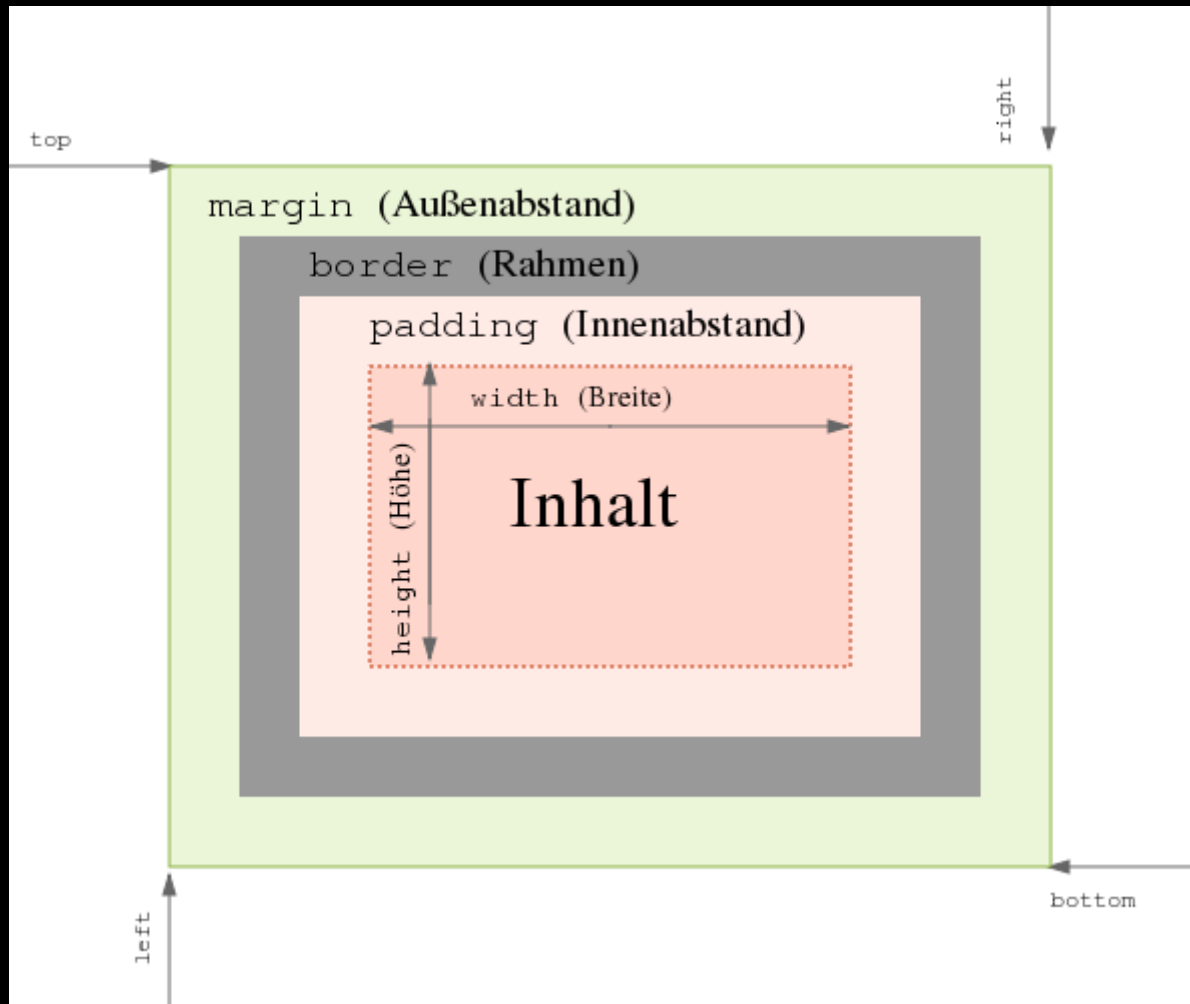
CSS-Animationen

transition: ...

animation: ...

- Einfache Animationen - <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Animation/Transition>
- Komplexe Animationen - <https://wiki.selfhtml.org/wiki/CSS/Eigenschaften/Animation/animation>

CSS-Eigenschaften: Box-Modell



box-sizing:

content-box
(Abmessungen sind nur der Inhalt, Standard)

border-box
(Abmessungen sind Inhalt, Innenabstand und Rahmen)

Zentrieren mit:

margin: auto;

Eigenschaften gelten nur für Blockelemente

CSS-Eigenschaften: Werte

Zahlenwerte: Dezimalzahlen mit . als Dezimaltrenner

Maßeinheiten:

px (Pixel – Achtung, keine Real-Pixel, sondern virtuelle Pixel)
in (Zoll) – pc (Pica) – pt (Punkt) – cm (Zentimeter) – mm (Millimeter)
em (aktuelle Schriftgröße) – rem (Schriftgröße des <html>-Elements)
vw (Viewport-Breite) – vh (Viewport-Höhe) – vmin/vmax
% (Prozent im Vergleich zum aktuellen Standardwert)

Empfehlungen:

Schriftgrößen in em oder rem; sub/sup/rt/br ... mit em gestalten
padding / margin / border-radius mit em, rem oder %
Längenangaben: %, rem, em, evtl. px

Gradangaben: deg oder rad, turn (Vollwinkel)

Zeitangaben: s oder ms

Funktionen wie calc(), var(), attr(), ...

Farbangaben: benannte Farben, #rrggbb (Hexadezimal), rgb(r,g,b),
rgba(r,g,b,alpha), hsl(h,s,l), hsla(h,s,l, alpha)

Wie kommt das CSS ins HTML?



CSS: Einbindung

Möglichkeit 1: Angaben direkt im Style-Attribut

```
<p style="color: blue; font-size: 0.5em">
```

Vorteile von CSS werden so nicht ausgeschöpft.

Möglichkeit 2: Angaben im Style-Element

```
<style>p { color: blue; font-size: 0.5em; }</style>
```

Angaben müssen auf jeder Seite einzeln neu notiert werden.

Möglichkeit 3: Einbindung aus Stylesheet-Datei

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

Einbindung einer zentralen Style-Datei im <head>. Hohe Wiederverwertbarkeit, keine unnötigen Wiederholungen, gute Wartbarkeit.

...und wofür steht Cascading?



CSS: Die Kaskade

Ein Element wird oft durch mehrere Selektoren in einer CSS-Datei selektiert. Dabei kommt es vor, dass den verschiedenen Eigenschaften dadurch ganz unterschiedliche Werte zugewiesen werden sollen. Welcher Wert in diesem Fall wirklich zählt, wird folgendermaßen entschieden:

- Angaben im Style-Attribut gehen immer vor
- Angaben mit Stichwort `!important` gewinnen immer über Angaben ohne `!important` (Verwendung: `color: green !important;`)
Achtung: Nur im Ausnahmefall verwenden, große Gefahr für Unübersichtlichkeit!
- Spezifischere Selektoren gewinnen über unspezifischere:
 - ID-Selektoren haben die stärkste Spezifität
 - Jeder vorkommende Attribut- / Klassenselektor / Pseudoklasse erhöht Spezifität
 - Bei kombinierten Selektoren nimmt die Spezifität durch beide Selektoren zu
- Bei gleich spezifischen Selektoren:
 - Die zuletzt im Stylesheet stehende Regel gewinnt

Wie geht man mit verschiedenen Displaygrößen um?



CSS: Media Queries

Unterschiedlich große Bildschirme verlangen unterschiedliche Darstellungsformen. Es gibt mehrere Möglichkeiten, das zu realisieren. Eine besonders sinnvolle Möglichkeit ist es, medienabhängige Bereiche im Stylesheet zu definieren.

Beispiel:

```
@media (width: 60em) {  
    /* Breite entspricht genau 60em */  
    p {  
        /* ...Deklarationen... */  
    }  
    /* ... */  
}  
@media (min-width: 50em) {  
    /* Breite beträgt mindestens 50em */  
    /* ...Deklarationen... */  
}  
@media (max-width: 70em) {  
    /* Breite beträgt höchstens 70em */  
    /* ...Deklarationen... */  
}
```

Tipps für responsive Designs:

- Mobile First
- Media-Queries mit em
- Don't repeat yourself

Noch Fragen?



CSS im SELFHTML-Wiki:

<https://wiki.selfhtml.org/wiki/CSS>

Can I use?

<https://caniuse.com/>