



ID versus CLASS



- Für Klassen wird in der CSS-Definition vor dem Namen ein Punkt geschrieben
- Für eine ID wird in der CSS-Definition vor dem Namen ein # geschrieben.
- Die vergebenen Namen dürfen keinerlei Sonderzeichen und Umlaute (öäüß) enthalten und keine Leerzeichen.
- Die Namen sind case sensitiv!
 - Also am besten alles klein schreiben.



- CSS-Elemente können sowohl über
 - `class="klassenname"`, als auch über
 - `id="idname"` angesprochen werden.
- Im folgenden Beispiel wird das anhand von Absätzen (`<p>`) und Hervorhebungen (``) gezeigt:

```
<p class="inhalt">Text</p>  
<strong class="inhalt">Text</strong>  
<p id="inhalt">Text</p>  
<strong id="inhalt">Text</strong>
```

- Das besondere an class ist, dass damit mehrere Elemente ausgezeichnet werden können, dagegen die ID nur für ein Element pro HTML-Seite verwendet werden darf.
- ID dient also zur eindeutigen Bestimmung eines Elementes.



- ID-Bereiche haben mehr Gewicht.
- Wenn also sowohl eine ID als auch eine Klasse vergeben wurde, werden die CSS-Angaben der ID verwendet.

```
<head>
  <style>
    .beispiel_class {
      color: blue;
    }
    #beispiel_id {
      color: red;
    }
  </style>
</head>
<body>
  <p id="beispiel_id" class="beispiel_class">Beispieltext</p>
</body>
```

Beispieltext



- Über die Klassen können statt dessen sich wiederholende Designelemente bestimmt werden.
- Im folgenden Beispiel werden im Text mehrere Stellen mit einer Art Textmarker hervorgehoben.

```
<head>
  <style>
    .wichtig {
      color: red;
    }
  </style>
</head>
<body>
<p>Dies ist <span class="wichtig">ein</span> Beispieltext.<br/>
Bestimmte Wörter sind in <span class="wichtig">diesem</span> Text besonders wichtig.</p>
</body>
```

Dies ist **ein** Beispieltext.

Bestimmte Wörter sind in **diesem** Text besonders wichtig.



- Elemente, die es nur einmal gibt, z.B. die Navigation, sollten mit einer ID und dem dazugehörigen CSS-Design ausgestattet werden.

```
<head>
  <style>
    #banner {
      background-color: red;
      color: white;
    }
  </style>
</head>
<body>
  <div id="banner">Herzlich willkommen!</div>
  <p>Dies ist ein Beispieltext.<br/>Bestimmte Wörter sind in diesem Text besonders wichtig.</p>
</body>
```

Herzlich willkommen!

Dies ist ein Beispieltext.

Bestimmte Wörter sind in diesem Text besonders wichtig.



shorthand-Eigenschaften



- In CSS können für Werte für Eigenschaften zugewiesen werden, die miteinander verwandt sind.
- Ein Beispiel hierfür ist
 - **font-family** (Schriftart),
 - **font-size** (Schriftgröße),
 - **font-weight** (Schriftstärke) etc.
 - All diese Eigenschaften gehören zu der Familie "**font**" (Schrift).
- Ein anderes Beispiel sind die inneren Abstände
 - **padding-top**,
 - **padding-right**,
 - **padding-bottom** und
 - **padding-left**,
 - die alle zur Familie "**padding**" gehören.



- In einer CSS-Regel können diese Eigenschaften einzeln angegeben werden.
- In dem Fall muss man für jede Eigenschaft den dazugehörigen Wert angeben:

```
div {  
    padding-top: 10px;  
    padding-right: 10px;  
    padding-bottom: 10px;  
    padding-top: 10px;  
}
```



- Bei manchen Eigenschaften ist es möglich, statt der einzelnen Eigenschaft/Werte-Paare die "Familie" anzugeben, zu der die Eigenschaften gehören.
- Anstelle von `padding-top`, `padding-right`, `padding-bottom` und `padding-left` könnte man auch lediglich die Eigenschaft `padding` angeben.
- Solche Eigenschaften werden in CSS Shorthand-Eigenschaften genannt.
- Wenn die einzelnen Werte für `padding` identisch sind, dann genügt ein einzelner Eintrag:

```
div{  
    padding: 10px;  
}
```



- Wenn die einzelnen Werte nicht identisch sind, so ist die Shorthand-Kurzschreibweise trotzdem möglich.
- Mehrere Werte werden dann mit einem Leerzeichen getrennt.
- In dem Fall muss man jedoch die Reihenfolge beachten.
- Beim **padding** beginnt man mit -bottom und die restlichen Werte gelten für die übrigen Eigenschaften im Uhrzeigersinn, also
 - bottom
 - left
 - top
 - right

```
div{  
    padding: 10px, 15px, 20px, 25px;  
}
```



Kaskadierung & Vererbung



- Die Möglichkeit der Kaskadierung ist eine Kernidee der CSS!
- Wenn es für ein Element mehrere Regeln für ein Attribut (z.B. color) gibt, dann gilt folgende Rangfolge
 - kontextabhängig individuell (p#id)
 - vor individuell (#id)
 - vor kontextabhängiger Klasse (p.achtung)
 - vor allgemeiner Klasse (.warnung)
 - vor redefiniertem Standard Format (p)
- Die Reihenfolge in einer Datei bestimmt das letzte Kriterium.
- Und es gilt die Regel: inline vor embedded vor externem Stylesheet



Beispiel einer kaskadierten Definition



```
<head>
  <style>
    p#hervorhebung { color: white; }
    p               { color: olive; }
    .achtung        { color: green; }
    p.achtung       { color: orange; }
    p#hervorhebung { color: red; }
    #hervorhebung  { color: grey; }
  </style>
</head>
<body>
  <p id="hervorhebung" class="achtung">Dies soll hervorgehoben werden.</p>
  <p id="auch" class="achtung">Und das hier auch.</p>
  <p id="noch" class="achtung" style="color: green;">Dies soll hervorgehoben werden.</p>
</body>
```

Dies soll hervorgehoben werden.

Und das hier auch.

Dies soll hervorgehoben werden.



- Wenn es für ein Element kein direktes Format gibt, wird das Format des übergeordneten Elements übernommen!
- Die Vererbung folgt dem Dokumenten-Baum.
- Manche Attribute werden aber nicht mit-vererbt!

```
<head>
  <style>
    body {
      color: green;
    }
  </style>
</head>
<body>
  <p>Ich bin ein Absatz</p>
  <p style="color: blue;">Ich bin noch ein Absatz</p>
  <div style="color: red;">
    <p>Ich bin ein Absatz</p>
  </div>
</body>
```

Ich bin ein Absatz

Ich bin noch ein Absatz

Ich bin ein Absatz



Bereich	Eigenschaft	vererbt?
Hintergrund	background-attachment, -color, -image, nein -position, -repeat	nein
Rahmen	border-collapse, -spacing	ja
	border-color, -style, -width	nein
Text	color	ja
	font-family, -size, -style, -variant, -weight	ja
	letter-spacing	ja
	line-height	ja
	text-align, -indent, -transform	ja
	text-decoration	nein
	word-spacing	ja
Listen	liste-style-image, -position, -type	ja
Positionierung	position, bottom, left, top, right	nein
	float, clear	nein
	z-index	nein



Bereich	Eigenschaft	vererbt?
Weitere Elemente	display	nein
	height, width	nein
	margin (betrifft alle Richtungen)	nein
	max-height, -width	nein
	min-height, -width	nein
	padding (betrifft alle Richtungen)	nein
	visibilty	ja
Pseudoelement	content	nein
Tabellen	caption-side	ja