PROGRAMMER'S CHOICE

Michael Bowers

CSS & HTML Design Patterns

350 CSS-Rezepte
für jedes Design und jeden Browser

ADDISON-WESLEY

3 CSS-Selektoren und Vererbung

In diesem Kapitel geht es um Design Patterns, mit denen Elemente für die Gestaltung ausgewählt werden.

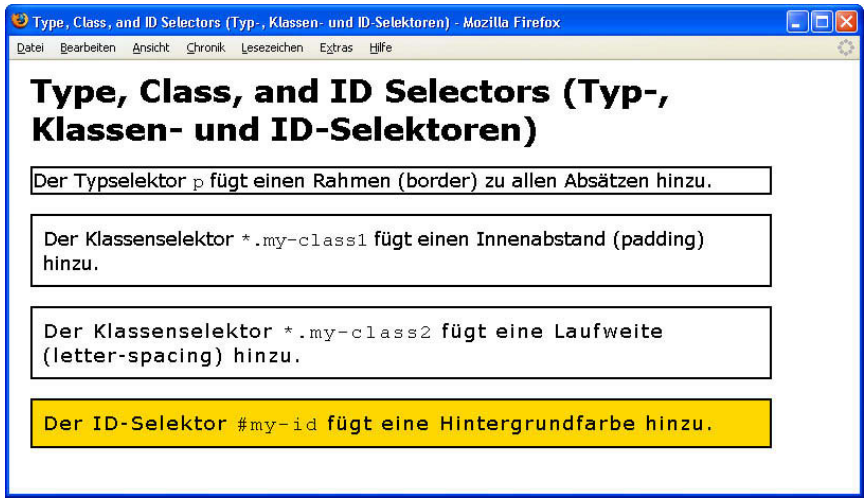
Da die Design Patterns für Selektoren einfach sind, bespreche ich sie gruppenweise statt einzeln. Dadurch lassen sich verwandte Formen von Selektoren besser vergleichen und voneinander abgrenzen. Obwohl dieses Kapitel nur *sechs* Beispiele enthält, werden darin *dreizehn* verschiedene Design Patterns angesprochen.

Ich habe die Vererbung mit in dieses Kapitel aufgenommen, da es sich dabei einfach um eine eingebaute Möglichkeit handelt, voneinander abstammende Elemente auszuwählen. Die Vererbung steht in einem engen Zusammenhang mit Nachkommenselektoren. In diesem Kapitel wird auch das Pattern Visual Inheritance besprochen, da es sich um eine optische Form der Vererbung handelt.

3.1 Was Sie in diesem Kapitel erwartet

- **Type, Class, and ID Selectors (Typen-, Klassen- und ID-Selektoren)** zeigt, wie Sie Elemente aufgrund von Tags, Klassen und IDs auswählen.
- **Position and Group Selectors (Positions- und Gruppenselektoren)** zeigt, wie Sie Elemente danach auswählen, wie sie in dem Dokument verschachtelt sind. Sie erfahren hier auch, wie Sie mehrere Selektoren auf einen Deklarationsblock anwenden.
- **Attribute Selectors (Attributselektoren)** zeigt, wie Sie Elemente aufgrund ihrer Attribute auswählen.
- **Pseudo-element Selectors (Pseudoelement-Selektoren)** zeigt, wie Sie den ersten Buchstaben oder die erste Zeile eines abschließenden Blockelements auswählen.
- **Pseudo-class Selectors (Pseudoklassen-Selektoren)** zeigt, wie Sie einen Hyperlink gestalten, wenn er noch nicht besucht oder bereits besucht ist, wenn der Mauszeiger über ihm schwebt und wenn er den Fokus hat, weil der Benutzer mit dem Tabulator zu ihm gesprungen ist oder ihn angeklickt hat.
- **Subclass Selectors (Subklassen-Selektoren)** zeigt, wie Sie mithilfe von Klassen und Subklassen mehrere Stile zu einem Element hinzufügen.
- **Inheritance (Vererbung)** zeigt, wie Sie Elemente mithilfe von Regeln gestalten, die ihren Vorfahren zugewiesen sind.
- **Visual Inheritance (Optische Vererbung)** zeigt, wie Elemente optisch den Hintergrund ihrer Elternelemente erben.

3.2 Type, Class, and ID Selectors (Typ-, Klassen- und ID-Selektoren)



HTML

```
<h1>Type, Class, and ID Selectors (Typ-, Klassen- und ID-Selektoren</h1>

<p>
  Der Typselektor <code>p</code> fügt einen Rahmen (border) zu allen Absätzen
  hinzu.</p>

<p class="my-class1">
  Der Klassenselektor <code>*.my-class1</code> fügt einen Innenabstand
  (padding)hinzu.</p>

<p class="my-class1 my-class2">
  Der Klassenselektor <code>*.my-class2</code> fügt eine Laufweite
  (letter-spacing) hinzu.</p>

<p class="my-class1 my-class2" id="my-id">
  Der ID-Selektor <code>#my-id</code> fügt eine Hintergrundfarbe hinzu. </p>
```

CSS

```
p { border:2px solid black; }

*.my-class1 { padding:10px; }
*.my-class2 { letter-spacing:0.11em; }

#my-id { background-color:gold; }
```

Type, Class, and ID Selectors (Typ-, Klassen- und ID-Selektoren)

Problem Sie möchten Elemente anhand ihres Typs, ihrer Klasse oder ihrer ID auswählen, damit Sie sie gestalten können.

Lösung Wenden Sie Stile wie folgt auf die gewünschten Klassen oder IDs an:

- Mit dem Typselektor wählen Sie alle Elemente eines bestimmten Typs aus. Der Typselektor ist der Name des Elements ohne die Größer-als- und Kleiner-als-Zeichen.
- Mit dem Klassenselektor wählen Sie alle Elemente aus, die Sie einer Klasse zugewiesen haben. Der Klassenselektor besteht aus einem Punkt, gefolgt vom Namen der Klasse, und wird an das Ende eines Typselektors angehängt. Um alle Elemente des Dokuments auszuwählen, die die richtige Klasse haben, fügen Sie den Klassenselektor hinter den universellen Selektor * ein, z. B. `*.my-class1`. Sie können auch den Selektor allein verwenden, z. B. in der Form `.my-class1`, was eine Abkürzung von `*.my-class1` darstellt.
- Mit dem ID-Selektor wählen Sie alle Elemente in dem Dokument aus, denen Sie die entsprechende ID zugewiesen haben. Jedes Element hat eine ID, die innerhalb eines Dokuments eindeutig sein sollte.

Patterns

HTML

```
<ELEMENT>
<ELEMENT class="class class class etc">
<ELEMENT id="id">
<ELEMENT id="id" class="class">
```

CSS

```
type    { STYLES }
*.class { STYLES }
#id     { STYLES }
```

Anwendung Diese Design Patterns gelten für alle Elemente.

Tipps Sie können mehreren Klassen ein Element zuweisen, indem Sie die Klassennamen durch ein Leerzeichen trennen. Der Klassenoperator wählt alle Elemente in der entsprechenden Klasse aus. Im Beispiel habe ich den zweiten und den dritten Absatz den beiden Klassen `my-class1` und `my-class2` zugewiesen.

Bei den Namen von Klassen und IDs wird zwischen Groß- und Kleinschreibung unterschieden. Sie müssen mit einem Buchstaben beginnen und dürfen Buchstaben, Zahlen und den Bindestrich enthalten. Ich empfehle, Klassen und IDs grundsätzlich in Kleinbuchstaben zu schreiben, da ein Browser eine Klasse oder ein Element nicht auswählen kann, wenn die Schreibweise im Selektor nicht genau mit der im Klassennamen übereinstimmt. Bei `div.selectme` kann der Browser z. B. `<div class="SelectMe" >` nicht auswählen.

Wenn mehrere Selektoren dasselbe Element auswählen, werden ihm die Stile aller Selektoren zugewiesen. Selektoren mit höherer Kaskadenpriorität überschreiben die Werte, die von Selektoren mit geringerer Priorität zugewiesen wurden. ID-Selektoren überschreiben Klassenselektoren, Klassenselektoren überschreiben Typselektoren. Wenn Sie mehrere Stylesheets auf ein Dokument anwenden, überschreiben die ID-Selektoren die Klassen- und Typselektoren in allen Stylesheets.

Verwandte Themen Position and Group Selectors, Pseudo-element Selectors, Pseudo-class Selectors

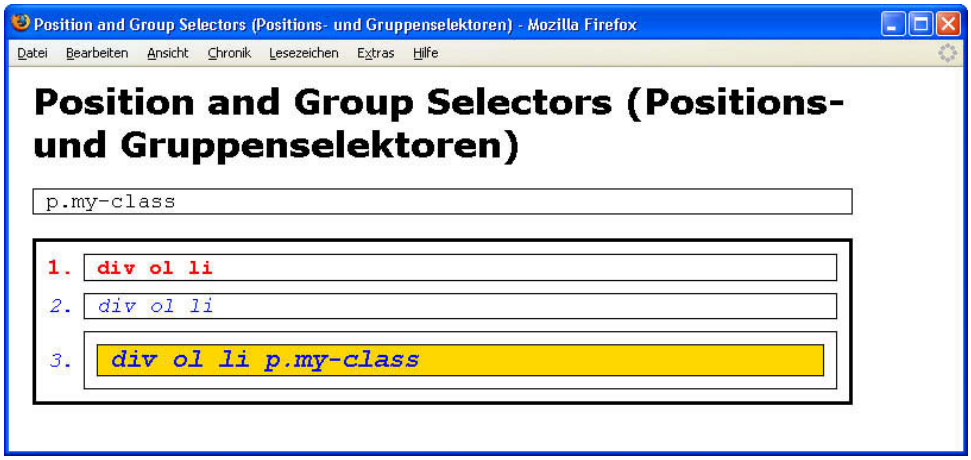
Siehe auch

www.cssdesignpatterns.com/type-selectors

www.cssdesignpatterns.com/class-selectors

www.cssdesignpatterns.com/id-selectors

3.3 Position and Group Selectors (Positions- und Gruppenselektoren)



HTML

```
<h1>Position and Group Selectors (Positions- und Gruppenselektoren)</h1>

<p class="my-class">p.my-class</p>
<div id="my-id">
  <ol>
    <li>div ol li</li>
    <li>div ol li</li>
    <li>
      <p class="my-class">div ol li p.my-class </p>
    </li>
  </ol>
</div>
```

CSS

```
/* Gruppenselektoren */
p,ol,li { border:1px solid black; padding-left:10px; font-family:monospace;
margin:10px; margin-left:0px; }
ol { margin-left:0px; padding-left:40px; margin-top:20px; }

/* Positionsselektoren */
div *.my-class { font-size:1.2em; font-weight:bold; } /* Nachkommenselektor */
#my-id p { background-color:gold; }                   /* Nachkommenselektor */
#my-id > * { border:3px solid black; }                /* Kindselektor */

li:first-child { font-weight:bold; color:red; } /* Selektor für erstes Kind */
li + li { font-style:italic; color:blue; }      /* Nachbarselektor */
```

Position and Group Selectors (Positions- und Gruppenselektoren)

Problem Sie möchten Selektoren kombinieren, um die Auswahl anhand der Elementposition einzugrenzen. Mit anderen Worten, Sie möchten Elemente danach auswählen, ob sie Nachkommen, Kinder oder Nachbarn anderer Elemente sind. Außerdem können Sie verschiedene Selektoren auf denselben Deklarationsblock anwenden.

Lösung Sie können Selektoren wie folgt kombinieren:

- Um einer Gruppe von Deklarationen verschiedene Selektoren zuzuweisen, verketten Sie die Selektoren mit einem Komma. Dies ist der *Gruppenselektor.* Jeder Selektor in der Kette wird unabhängig von den anderen auf denselben Satz von Stilen angewendet.
- Um Nachkommenelemente auszuwählen, verketten Sie mehrere Selektoren mit Leerzeichen. Das Leerzeichen fungiert dabei als *Nachkommenselektor*. Jeder Nachkommenselektor engt die Auswahl der Nachfahren des vorhergehenden Selektors ein. Ein Nachkomme kann ein Kind, ein Enkelkind, ein Urenkel usw. sein.
- Um Kindelemente auszuwählen, verketten Sie mehrere Selektoren mit dem Größer-als-Zeichen. Dies ist der *Kindselektor*. Jeder Kindselektor engt die Auswahl auf die Elemente ein, die Kinder der vorhergehenden Auswahl sind.
- Um das erste Kindelement auszuwählen, hängen Sie `:first-child` an einen beliebigen Selektor an. Dadurch wird die Auswahl auf das Element eingeschränkt, das das erste Kind seiner Eltern ist.
- Um Nachbarelemente auszuwählen, verketten Sie mehrere Selektoren mit dem Pluszeichen. Dies ist der *Nachbarselektor*. Jeder Nachbarselektor engt die Auswahl auf die Nachbarelemente der vom vorherigen Selektor ausgewählten Elemente ein.

Patterns

CSS

```
selector, selector, etc { STYLES }
```

oder

```
selector  selector  etc { STYLES }
```

oder

```
selector > selector > etc { STYLES }
```

oder

```
selector + selector + etc { STYLES }
```

oder

```
selector:first-child      { STYLES }
```

Anwendung Diese Design Patterns gelten für alle Elemente.

Einschränkungen In Internet Explorer 6 funktionieren nur der Gruppen- und der Nachkommenselektor. In Internet Explorer 7 und allen anderen wichtigen Browsern funktionieren dagegen alle diese Selektoren.

Beispiel Der Gruppenselektor `p, ol, li` wendet dieselben Stile auf alle Absätze, geordnete Listen und Listeneinträge an. Der Selektor `div *.my-class` wählt alle Elemente aus, die `my-class` zugewiesen und Nachfahren eines `div`-Bereichs sind. Dies trifft nur auf den Absatz im dritten Listeneintrag zu. Der Selektor `#my-id p` wählt alle Absätze aus, die Nachkommen von `<div id="my-id">` sind. Auch dieser Selektor gilt nur für den Absatz im dritten Listeneintrag. Der Selektor `#my-id > p` wählt alle Kindelemente von `<div id="my-id">` aus. Dies trifft nur auf die geordnete Liste zu. Der Selektor `li:first-child` wählt den ersten Listeneintrag in jeder Liste aus. Mit dem Selektor `li + li` werden alle Listeneinträge ausgewählt, die Nachbarn von Listeneinträgen sind. Damit werden alle Listeneinträge außer dem ersten ausgewählt.

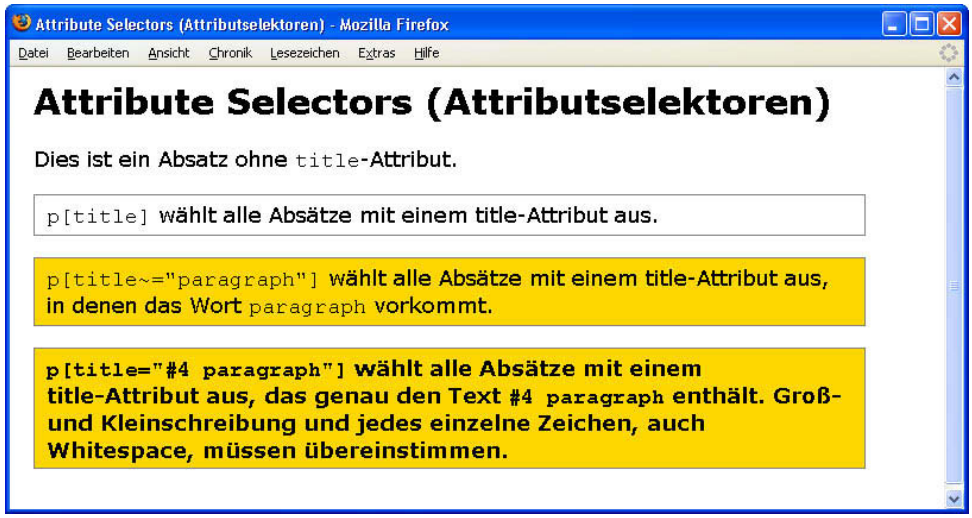
Verwandte Themen Inheritance

Siehe auch

www.cssdesignpatterns.com/position-selectors

www.cssdesignpatterns.com/group-selectors

3.4 Attribute Selectors (Attributselektoren)



HTML

```
<h1>Attribute Selectors (Attributselektoren)</h1>

<p>Dies ist ein Absatz ohne <code>title</code>-Attribut.</p>

<p title="Second">
  <code>p[title]</code> wählt alle Absätze mit einem title-Attribut aus.</p>

<p title="Third paragraph">
  <code>p[title~="paragraph"]</code> wählt alle Absätze mit einem
  title-Attribut aus, in denen das Wort <code>paragraph</code> vorkommt.</p>

<p title="#4 paragraph">
  <code>p[title="#4 paragraph"]</code> wählt alle Absätze mit einem
  title-Attribut aus, das genau den Text <code>#4 paragraph</code> enthält. Groß- und
Kleinschreibung und jedes einzelne Zeichen, auch Whitespace,
müssen übereinstimmen.</p>
```

CSS

```
code { white-space:pre; }

p[title] { padding:5px 10px; border:1px solid gray; }
p[title~="paragraph"] { background-color:gold; }
p[title="#4 paragraph"] { font-weight:bold; }
```

Attribute Selectors (Attributselektoren)

Problem Sie möchten Elemente abhängig davon auswählen, ob sie ein bestimmtes Attribut oder bestimmte Wörter oder Werte innerhalb eines gegebenen Attributs enthalten.

Lösung Für diesen Zweck stellt CSS drei Attributselektoren bereit. In CSS tragen sie keine eigenen Bezeichnungen, doch ich nenne sie Attributexistenz-Selektor, Attributwort-Selektor und Attributwert-Selektor. Sie können diese Attributselektoren an jeden anderen Selektor anhängen.

Mit dem *Attributexistenz-Selektor* können Sie Elemente auswählen, die ein bestimmtes Attribut enthalten. Der Selektor besteht aus dem Namen des Attributs in eckigen Klammern. `p[title]` wählt z. B. alle Absätze aus, die das Attribut `title` enthalten. Wenn ein Element das Attribut enthält und diesem Attribut ein Wert zugewiesen ist, erkennt der Attributexistenz-Selektor eine Übereinstimmung. Das Attribut kann jeden beliebigen Wert enthalten, doch einige Browser finden bei einem leeren Attribut wie `<p title="">` *keine* Übereinstimmung.

Mit dem *Attributwort-Selektor* können Sie Elemente auswählen, bei denen in einem gegebenen Attribut ein bestimmtes Wort vorkommt. Dieser Selektor besteht aus einer öffnenden eckigen Klammer, dem Attributnamen, einer Tilde, einem Gleichheitszeichen, dem gesuchten Wort in doppelten Anführungszeichen und der schließenden eckigen Klammer. `p[title~="paragraph"]` wählt z. B. alle Absätze aus, in deren `title`-Attribut das Wort `paragraph` vorkommt, z. B. `<p title="Third paragraph">`. Neben dem gesuchten Wort kann das Attribut auch noch andere enthalten. Wörter sind voneinander durch Leerzeichen getrennt. Beim Vergleich wird auf die Groß- und Kleinschreibung geachtet.

Mit dem *Attributwert-Selektor* können Sie Elemente auswählen, bei denen ein gegebenes Attribut einen bestimmten Wert aufweist. Dieser Selektor besteht aus einer öffnenden eckigen Klammer, dem Attributnamen, einem Gleichheitszeichen, dem Wert in doppelten Anführungszeichen und der schließenden eckigen Klammer. `p[title="#4 paragraph"]` wählt z. B. alle Absätze aus, deren `title`-Attribut genau den Wert `#4 paragraph` aufweist, also `<p title="#4 paragraph">`. Beim Vergleich wird zwischen Groß- und Kleinschreibung unterschieden. Außerdem muss der gesamte Attributwert einschließlich aller Leerzeichen übereinstimmen.

Patterns

CSS

```
SELECTOR[title] { STYLES }
```

oder

```
SELECTOR[title~="WORD"] { STYLES }
```

oder

```
SELECTOR[title="EXACT_MATCH_OF_ENTIRE_VALUE"] { STYLES }
```

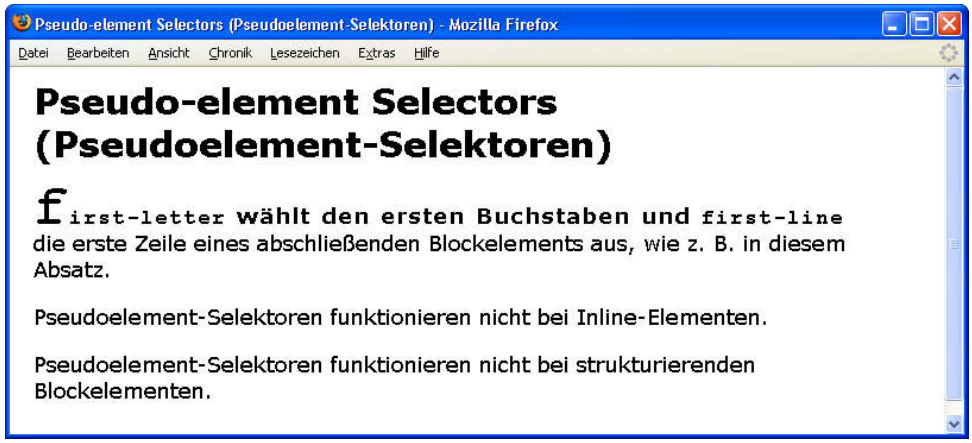
Anwendung Diese Design Patterns gelten für alle Elemente.

Einschränkungen Attributselektoren funktionieren nicht in Internet Explorer 6, aber in Internet Explorer 7 und allen anderen wichtigen Browsern. CSS definiert auch einen anderen Selektor, den ich den *Sprachattribut-Selektor* nenne (z. B. `[lang=de]`), aber er wird nicht sehr gut unterstützt.

Verwandte Themen Inheritance

Siehe auch *www.cssdesignpatterns.com/attribute-selectors*

3.5 Pseudo-element Selectors (Pseudoelement-Selektoren)



HTML

```
<h1>Pseudo-element Selectors (Pseudoelement-Selektoren)</h1>

<p><code>first-letter</code> wählt den ersten Buchstaben und
  <code>first-line</code> die erste Zeile eines abschließenden Blockelements
  aus, wie z. B. in diesem Absatz.</p>
  <div><span>Pseudoelement-Selektoren funktionieren nicht bei
  Inline-Elementen.</span></div>
<dl>
  <dt>Pseudoelement-Selektoren funktionieren nicht bei strukturierenden
  Blockelementen.</dt>
</dl>
```

CSS

```
p:first-line { font-weight:bold; word-spacing:2px; letter-spacing:1px; }
p:first-letter { font-size:48px; }

span:first-line { font-weight:bold; word-spacing:2px; letter-spacing:1px; }
span:first-letter { font-size:48px; }

dl:first-line { font-weight:bold; word-spacing:2px; letter-spacing:1px; }
dl:first-letter { font-size:48px; }
```

Pseudo-element Selectors (Pseudoelement-Selektoren)

Problem Sie möchten den ersten Buchstaben oder die erste Zeile eines Elements auswählen.

Lösung

HTML

Hierfür ist kein Markup erforderlich.

CSS

Kombinieren Sie die Pseudoelement-Selektoren `first-letter` und `first-line` nach Bedarf mit Klassen-, ID- und Typselektoren.

Patterns

CSS

```
ELEMENT:first-letter { STYLES }
```

oder

```
*.CLASS:first-letter { STYLES }
```

oder

```
#ID:first-letter { STYLES }
```

oder

```
ELEMENT:first-line { STYLES }
```

oder

```
*.CLASS:first-line { STYLES }
```

oder

```
#ID:first-line { STYLES }
```

Anwendung `first-letter` und `first-line` funktionieren nur bei abschließenden Blockelementen, nicht bei Inline-Elementen oder strukturierenden Blockelementen.

Hinweise `first-letter` und `first-line` werden Pseudoelement-Selektoren genannt, da sie nicht den Inhalt eines Elements auswählen, sondern nur eine Teilmenge davon. Mit anderen Worten, sie erstellen ein Pseudoelement.

Einschränkungen Internet Explorer 6 ignoriert Pseudoelement-Selektoren, es sei denn, es handelt sich um den letzten Selektor in einer Kette. In Version 7 ist dieses Problem behoben.

Der Selektor `first-letter` funktioniert am besten bei Schrift- und Texteigenschaften. Browser können Pseudoelemente nicht positionieren und nur sehr schwer ausrichten. Mit anderen Worten, `position`, `left`, `right`, `top` und `bottom` haben keine Auswirkung auf Pseudoelemente. `vertical-align` funktioniert bei Pseudoelementen nur inkonsistent.

Bei Browsern gibt es Ausnahmefälle, in denen sie den ersten Buchstaben nicht auswählen können oder mehr als den ersten Buchstaben auswählen. So kann z. B. keiner der wichtigen Browser den ersten Buchstaben auswählen, wenn ihm ein Bild oder ein Objekt vorausgeht. Opera 9 wählt den ersten Buchstaben in Tabellenzellen nicht aus, und Internet Explorer 6 wählt zusammen mit dem ersten Buchstaben eines Listeneintrags auch das Aufzählungszeichen aus. Bei den Pseudoelement-Selektoren treten die Bugs von Browsern zutage, weshalb Sie diese Selektoren in allen wichtigen Browsern testen sollten.

Beispiel In dem Beispiel habe ich drei verschiedenen Pseudoelement-Selektoren denselben Satz von Stilen zugewiesen. Ich habe keine Gruppenselektoren verwendet, da Internet Explorer keine Pseudoelement-Selektoren erkennt, wenn sie Teil eines Gruppenselektors sind.

Verwandte Themen Class Selector, Pseudo-class Selectors

Siehe auch *www.cssdesignpatterns.com/pseudo-element-selectors*

3.6 Pseudo-class Selectors (Pseudoklassen-Selektoren)

HTML

```
<h1>Pseudo-class Selectors (Pseudoklassen-Selektoren)</h1>

<p>
  <a href="http://www.cssdesignpatterns.com">a:link -- Nicht besuchter
  Link</a>
  <a href="http://www.htmldesignpatterns.com">a:visited -- Besuchter Link</a>
  <a href="http://www.cssdesignpatterns.com">a:hover -- Mauszeiger über
  Link</a>
  <a href="http://www.cssdesignpatterns.com">a:active -- Aktiver Link</a>
</p>
```

CSS

```
a { padding:3px 10px; margin:20px 10px; text-decoration:none;
  display:block; width:260px;
  border-left:1px solid dimgray; border-right:2px solid black;
  border-top:1px solid dimgray; border-bottom:2px solid black; }

a:link { color:black; background-color:white; }
a:visited { color:gray; background-color:white; }
a:hover { color:white; background-color:green; }
a:active, a:focus { color:green; background-color:gold; }
```

Pseudo-class Selectors (Pseudoklassen-Selektoren)

Problem Sie möchten einen Hyperlink in Abhängigkeit davon gestalten, ob er bereits besucht oder noch nicht besucht wurde, ob der Mauszeiger über ihm steht oder ob er gerade aktiviert wird.

Lösung

HTML

Fügen Sie Hyperlinks mit `<a>` ein.

CSS

Wählen Sie die Hyperlinks nach ihrem Status aus:

- Mit `a:link` wählen Sie einen Hyperlink aus, der noch nicht besucht wurde.
- Mit `a:visited` wählen Sie einen Hyperlink aus, der bereits besucht wurde.
- Mit `a:hover` wählen Sie einen Hyperlink aus, wenn sich der Mauszeiger über ihm befindet.
- Mit `a:focus` wählen Sie einen Hyperlink aus, wenn er in Browsern außer Internet Explorer den Fokus erhält.
- Mit `a:active` wählen Sie einen Hyperlink aus, wenn er in Internet Explorer den Fokus erhält.

Patterns

HTML

```
<a>
```

CSS

```
a:link { STYLES }
a:visited { STYLES }
a:hover { STYLES }
a:active, a:focus { STYLES }
```

Anwendung Pseudoklassen-Selektoren werden auf Hyperlinks (`<a>`) angewendet.

Einschränkungen Internet Explorer 6 unterstützt die Pseudoklasse `hover` nur für Hyperlinks, IE7 und alle anderen wichtigen Browser dagegen für alle Elemente.

In CSS 2.1 sind zwei weitere Pseudoklassen definiert: `first-child` und `lang()`. `first-child` wählt ein Element aus, wenn es das erste Kind eines anderen Elements ist. `lang()` wählt ein Element aus, wenn ihm die angegebene menschliche Sprache zugewiesen wurde. Diese Pseudoklassen werden in Internet Explorer 6 nicht unterstützt. Internet Explorer 7 versteht `first-child`, aber nicht `lang`. Sie sollten diese Pseudoklassen nicht verwenden, solange nicht ein Großteil der Benutzer Browser verwendet, die sie unterstützen.

Tipps Die Unterstreichung ist der standardmäßige optische Indikator für einen Hyperlink. Wenn Sie sie entfernen, sollten Sie die Hyperlinks so gestalten, dass sie anklickbar aussehen. In dem Beispiel habe ich sie so formatiert, dass sie wie Schaltflächen aussehen.

Die Pseudoklassen-Selektoren sollten Sie in Ihrem Stylesheet in der zuvor angegebenen Reihenfolge aufführen (`link`, `visited`, `hover`, `active`, `focus`). Um sich die Reihenfolge einzuprägen, eignet sich der Merkspruch »Las Vegas Hells Angels Fight«.

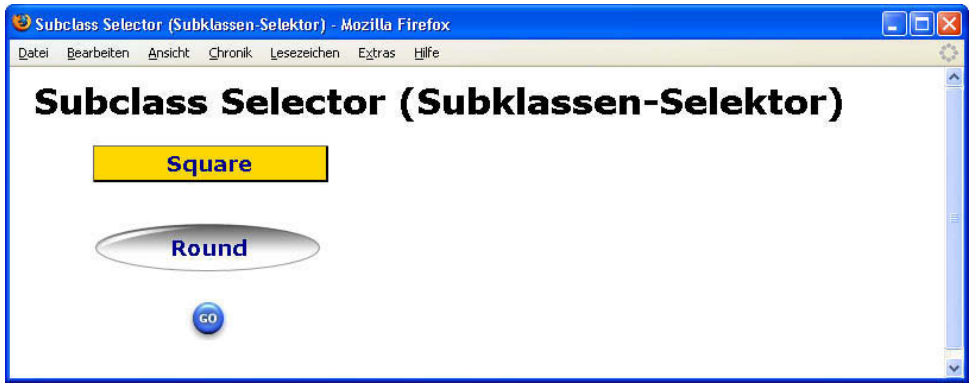
Ein Browser zeigt den aktiven Status an, wenn ein Benutzer mit dem Tabulator zu dem Hyperlink springt. Dieser Status wird auch für weniger als eine Sekunde angezeigt, wenn der Benutzer auf den Link klickt. Sie können der Pseudoklasse `active` einen kontrastierenden Stil zuweisen, damit der Hyperlink »aufblitzt«, wenn der Benutzer darauf klickt. Dadurch erhält der Benutzer eine unmittelbare Rückmeldung, dass der Browser den Klickvorgang erkannt hat.

Varianten Zum Gestalten von Hyperlinks können Sie jegliche Kombination von CSS-Stilen verwenden.

Verwandte Themen Class Selector, Pseudo-element Selectors

Siehe auch *www.cssdesignpatterns.com/pseudo-class-selectors*

3.7 Subclass Selector (Subklassen-Selektor)



HTML

```
<h1>Subclass Selector (Subklassen-Selektor)</h1>
<div>
  <p class="button square">Square</p>
  <p class="button rounded">Round</p>
  <p class="button go">Go</p>
</div>
```

CSS

```
*.button { width:175px; padding:3px 10px; margin:20px 0; text-align:center;
  font-weight:bold; margin-left:50px; line-height:normal; }
*.button.square { color:darkblue; background-color:gold;
  border-left:1px solid dimgray; border-right:2px solid black;
  border-top:1px solid dimgray; border-bottom:2px solid black; }
*.button.rounded { color:darkblue; background-color:white;
  line-height:45px; margin-top:30px;
  background:url("oval.gif") no-repeat center center; }
*.button.go { background-color:white; line-height:26px;
  text-indent:-9999px; font-size:10px;
  background: url("go.jpg") no-repeat center center; }
```

Subclass Selector (Subklassen-Selektor)

Problem Sie möchten eine Klasse von Elementen mit gemeinsamen Regeln gestalten. Allerdings sollen diese Elemente in Unterklassen aufgeteilt und mit besonderen Regeln formatiert werden, die die Grundregeln überschreiben.

Lösung

HTML

Im HTML-Code können Sie den Elementen mithilfe des Attributs `class` Klassen zuweisen. Ein `class`-Attribut kann eine unbegrenzte Anzahl von Klassen enthalten, die durch Leerstellen voneinander getrennt sind. Die Reihenfolge der Klassen in dem Attribut ist ohne Bedeutung, ich empfehle aber der besseren Lesbarkeit halber, zuerst die Basisklasse und danach die Subklassen aufzuführen. Die Klassen, denen ein Element zugewiesen ist, müssen nicht in einer Beziehung zueinander stehen, doch der Code wird logischer, wenn Sie sie nach Klassen und Subklassen gliedern.

CSS

Um alle Elemente auszuwählen, die der Basisklasse zugewiesen sind, verwenden Sie den Universalselektor, gefolgt vom Punktoperator und dem Namen der Klasse.

Wenn Sie alle Elemente einer Subklasse auswählen möchten, verwenden Sie den Universalselektor, gefolgt vom Punktoperator, dem Namen der Basisklasse, einem weiteren Punktoperator und dem Namen der Subklasse. Diesen Vorgang bezeichne ich als Verketten von Klassen. Sie können eine unbegrenzte Anzahl von Klassen verketten. Die Reihenfolge im Selektor ist unwichtig. Aus Gründen der Lesbarkeit empfehle ich aber, zuerst die Basisklasse und dann deren Subklassen aufzuführen. Die Klassen, die Sie verketten, müssen nicht in einer Beziehung zueinander stehen, doch der Code wird logischer, wenn Sie sie nach Klassen und Subklassen gliedern.

Patterns

HTML

```
<ELEMENT class="class subclass etc">
```

CSS

```
*.class { SHARED_BASE_STYLES }
*.class.subclass.etc { SUBCLASS_STYLES }
```

Anwendung Sie können dieses Design Pattern auf jedes Element anwenden.

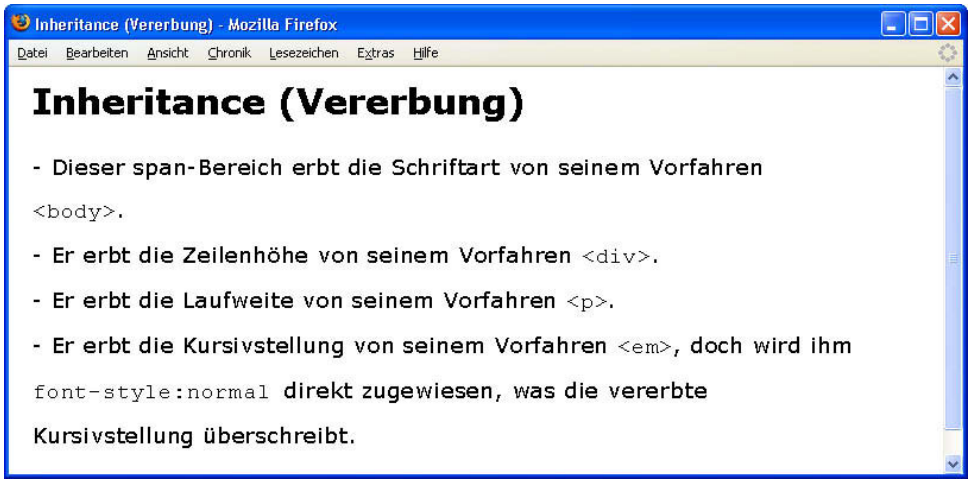
Vorteile Mit diesem Design Pattern können Sie eine Hierarchie von Regeln auf der Grundlage von Klassen und Subklassen erstellen. Wie in der objektorientierten Programmierung »erben« Subklassen die Regeln ihrer Basisklasse und deren Subklassen. Die Kaskadenreihenfolge von CSS stellt sicher, dass Regeln von Subklassen die Regeln der Basisklasse überschreiben.

Beispiel In dem Beispiel sind alle Absätze der Klasse `button` zugewiesen. Außerdem gehört jede zu einer der Subklassen `square`, `rounded` und `go`. Alle Absätze, die zur Klasse `button` gehören, verwenden gemeinsam die Grundregeln, die durch `*.button` zugewiesen sind, z. B. `width:175px`. Jedem Subklassen-Absatz sind aber durch `*.button.square`, `*.button.round` bzw. `*.button.go` besondere Regeln zuwiesen. So verwendet jede Subklasse z. B. einen anderen Hintergrund für ihre Art von Schaltfläche. Einige der Subklassen-Regeln, z. B. `margin` und `line-height`, überschreiben Grundregeln.

Verwandte Themen Class Selector

Siehe auch *www.cssdesignpatterns.com/subclass-selector*

3.8 Inheritance (Vererbung)



HTML

```
<body>
  <h1>Inheritance (Vererbung)</h1>

  div>
  <p>
   <em>
    <span>
     - Dieser span-Bereich erbt die Schriftart von seinem
     Vorfahren <code>&lt;body&gt;</code>. <br />
     - Er erbt die Zeilenhöhe von seinem
     Vorfahren <code>&lt;div&gt;</code>. <br />
     - Er erbt die Laufweite von seinem
     Vorfahren <code>&lt;p&gt;</code>. <br />
     - Er erbt die Kursivstellung von seinem
     Vorfahren <code>&lt;em&gt;</code>,doch wird ihm <code>font-style:normal
     </code> direkt zugewiesen, was die vererbte Kursivstellung überschreibt.
    </span>
   </em>
  </p>
 </div>
</body>
```

CSS

```
body { font-family:verdana,arial,sans-serif; font-size:18px; }
div { line-height:2em; }
p { letter-spacing:0.8px; }
em { font-style:italic; }
span { font-style:normal; }
```

Inheritance (Vererbung)

Problem Sie möchten ein Element gestalten und wünschen, dass all seine Nachkommen denselben Stil aufweisen.

Lösung CSS ist so entworfen, dass viele Eigenschaften automatisch vererbt werden. Das bedeutet, dass Sie einem Element eine dieser Eigenschaften zuweisen können und dessen Nachkommen dann die Eigenschaft erben. Die meisten Inline-Eigenschaften werden standardmäßig vererbt. Weiter unten finden Sie eine Liste aller Eigenschaften mit Angaben zu ihrer Vererbung.

Patterns Die Vererbung ist eine Art Selektor, der in die Sprache CSS integriert ist. Um sie zu nutzen, müssen Sie nichts tun. Wenn ein Browser auf eine vererbte Eigenschaft stößt, wählt er automatisch alle entsprechenden Nachkommenelemente aus und wendet die Regel auf sie an. Eigenschaften, die Sie einem Element direkt zuweisen, überschreiben jeglichen vererbten Wert.

Vererbte Eigenschaften

Die folgenden Eigenschaften werden von allen Elementen vererbt:

`visibility` **und** `cursor`

Die folgenden Eigenschaften werden von Inline-Elementen vererbt:

`letter-spacing`, `word-spacing`, `white-space`, `line-height`, `color`, `font`, `font-family`, `font-size`, `font-style`, `font-variant`, `font-weight`, `text-decoration`, `text-transform` **und** `direction`

Die folgenden Eigenschaften werden von abschließenden Blockelementen vererbt:

`text-indent` **und** `text-align`

Die folgenden Eigenschaften werden von Listenelementen vererbt:

`list-style`, `list-style-type`, `list-style-position` **und** `list-style-image`

Die folgende Eigenschaft wird von Tabellenelementen vererbt:

`border-collapse`

Nicht vererbt Die folgenden Eigenschaften werden *nicht* vererbt:

`display`, `margin`, `border`, `padding`, `background`, `height`, `min-height`, `max-height`, `width`, `min-width`, `max-width`, `overflow`, `position`, `left`, `right`, `top`, `bottom`, `z-index`, `float`, `clear`, `table-layout`, `vertical-align`, `page-break-after`, `page-break-before` **und** `unicode-bidi`

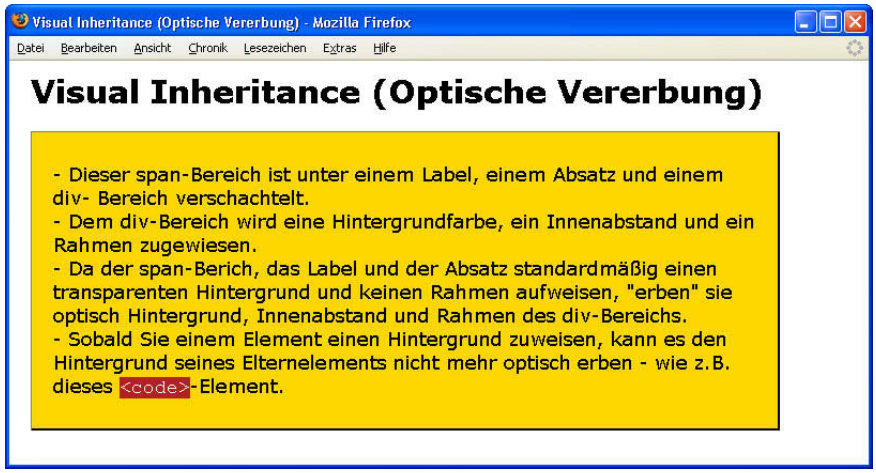
Einschränkungen CSS stellt einen konstanten Wert namens `inherited` bereit, den Sie jeder Eigenschaft zuweisen können. In diesem Fall erbt die betreffende Eigenschaft ihren Wert von ihrem Elternelement. Dadurch können Sie die Vererbung von Eigenschaften erzwingen. Internet Explorer 7 und frühere Versionen implementieren `inherit` jedoch nicht. Im folgenden Tipp sehen Sie, wie Sie die Vererbung bei jeder Eigenschaft simulieren können.

Tipp Sie können die Vererbung bei Eigenschaften simulieren, die normalerweise keine Vererbung aufweisen. Wählen Sie zunächst mit einem beliebigen Selektor ein Anfangselement aus. Hängen Sie dann den Nachkommen- und den Universalselektor daran an. Das Pattern lautet `SELECTOR *`. Beispielsweise können Sie mit `hmtl * { border:1px solid black; }` einen Rahmen um alle Elemente legen, die von `<html>` abstammen. Ich verwende diesen Code häufig, um die Verschachtelung der Elemente in einem Dokument einzusehen.

Verwandte Themen Position and Group Selectors

Siehe auch *www.cssdesignpatterns.com/inheritance*

3.9 Visual Inheritance (Optische Vererbung)



HTML

```
<h1>Visual Inheritance (Optische Vererbung)</h1>
<div>
 <p>
  <label>
   <span>
    - Dieser span-Bereich ist unter einem Label, einem Absatz und einem div-
      Bereich verschachtelt. <br />
    - Dem div-Bereich wird eine Hintergrundfarbe, ein Innenabstand und ein
      Rahmen zugewiesen. <br />
    - Da der span-Bereich, das Label und der Absatz standardmäßig einen
      transparenten Hintergrund und keinen Rahmen aufweisen, »erben« sie
      optisch Hintergrund, Innenabstand und Rahmen des div-Bereichs. <br />
    - Sobald Sie einem Element einen Hintergrund zuweisen, kann es den
      Hintergrund seines Elternelements nicht mehr optisch erben - wie z. B.
      dieses <code>&lt;code&gt;</code>-Element.
   </span>
  </label>
 </p>
</div>
```

CSS

```
div { background-color:gold; color:black; padding:10px 20px;
  border-left:1px solid gray; border-right:2px solid black;
  border-top:1px solid gray; border-bottom:2px solid black; }

p { background-color:transparent; background-image:none; }
label { background-color:transparent; background-image:none; }
span { background-color:transparent; background-image:none; }

code { background-color:firebrick; color:white; }
```

Visual Inheritance (Optische Vererbung)

Problem Sie möchten, dass ein Kindelement den gleichen Hintergrund hat wie sein Elternelement.

Lösung CSS schichtet Elemente automatisch transparent übereinander. Kindelemente liegen dabei über ihren Elternelementen. Wenn Nachbarelemente aufgrund von Außenabständen oder der Positionierung überlappen, kommen die nachfolgenden Nachbarn über den früheren zu liegen. Bei positionierten Elementen können Sie die Schichtung mit der Eigenschaft `z-index` ausdrücklich festlegen. Dieses Design Pattern ist in CSS integriert. Sie müssen nichts tun, um es zu nutzen.

Die Eigenschaft `background-color` ist standardmäßig `transparent`, die Eigenschaft `background-image` standardmäßig `none`. Dadurch kann der Hintergrund der Vorfahren eines Elements durchscheinen. Mit anderen Worten, ein Browser stellt Kindelemente auf transparenten Schichten oberhalb der Elternelemente dar, sofern Sie die Eigenschaften `background-color` oder `background-image` des Kindes nicht ausdrücklich auf eine Farbe bzw. ein Bild setzen.

Da Kindelemente innerhalb ihrer Elternelemente verschachtelt sind, erbt jedes Kindelement *optisch* die Werte für `border` (Rahmen) und `padding` (Innenabstand) seines Elternelements. Anders ausgedrückt, `border` und `padding` des Elternelements umgeben das Kindelement. Wenn ein Kind einen transparenten Hintergrund und keinen `border` hat, sieht es so aus, als gehörten `border` und `padding` des Elternelements zu dem Kindelement. Ohne Rahmen um das Kind können Sie nicht unterscheiden, wo der Innenabstand des Elternelements endet und wo der des Kinds beginnt. Sobald Sie einem Kindelement `border` hinzugefügt haben, erbt es nicht mehr optisch Rahmen und Innenabstand des Elternelements, da Sie jetzt genau erkennen können, wo das eine anfängt und das andere aufhört.

Patterns Um die optische Vererbung zu nutzen, müssen Sie nichts tun, denn `background-color` ist standardmäßig `transparent` und `background-image` standardmäßig `none`. Wenn ein Kind den Hintergrund seines Elternelements nicht optisch erben soll, können Sie dem Element wie folgt eine eigene Hintergrundfarbe oder ein Hintergrundbild zuweisen:

```
SELECTOR { background-color:COLOR;
   background-image:url("FILE.EXT"); }
```

Anwendung Dieses Design Pattern gilt für alle Elemente.

Beispiel In diesem Beispiel hat der `div`-Bereich einen goldenen Hintergrund, den alle Nachkommenelemente erben – bis auf das `<code>`-Element, dem die Hintergrundfarbe `firebrick` zugewiesen ist. Beachten Sie, dass ich Absatz, Label und `span`-Bereich `background-color:transparent` und `background-image:none` zugewiesen habe. Dies habe ich nur getan, um die Regeln in Aktion zu zeigen. Normalerweise müssen Sie diese Regeln im Code nicht angeben, da es sich dabei um die Standardwerte für alle Elemente handelt. Sie können diese Regeln jedoch verwenden, wenn Sie ein Element auf einen transparenten Hintergrund zurücksetzen müssen, nachdem ihm von einer anderen Regel eine Hintergrundfarbe oder ein Bild zugewiesen wurde.

Verwandte Themen Inheritance

Siehe auch *www.cssdesignpatterns.com/visual-inheritance*