

Medientechnik

Markup Languages, Stylesheets
und JavaScript

Übungsblatt 3

<http://www.uni-koblenz.de/~ugotit>

Fragen?



Was bisher geschah

- HTML
 - Grundlagen
 - Strukturierung von Inhalten
- CSS
 - Grundlagen
 - einfache Layouts
 - Schriften

Themen heute

- Markup Languages
Wiederholung zu XML, HTML und SVG
- CSS
Wiederholung & Vertiefung
- JavaScript
Grundlagen, jQuery, click-Events

XML: Extensible Markup Language

Wofür XML?



XML

- ist eine Auszeichnungssprache
- Darstellung hierarchisch strukturierter Daten in Form von Textdateien
- u.a. für plattformunabhängigen Austausch von Daten
- Besonders im Einsatz in Webapplikationen (Stichwort AJAX)

SVG - Scalable Vector Graphics

SVG - Basisdatei

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve" width="200px" height="50px" viewBox="0 0 200 50" zoomAndPan="disable" >
```

```
<!-- hier dann die restliche SVG-Datei -->
```

```
</svg>
```

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >



Namespace

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >



Namespace

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >



Namespace

<svg

- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

<svg

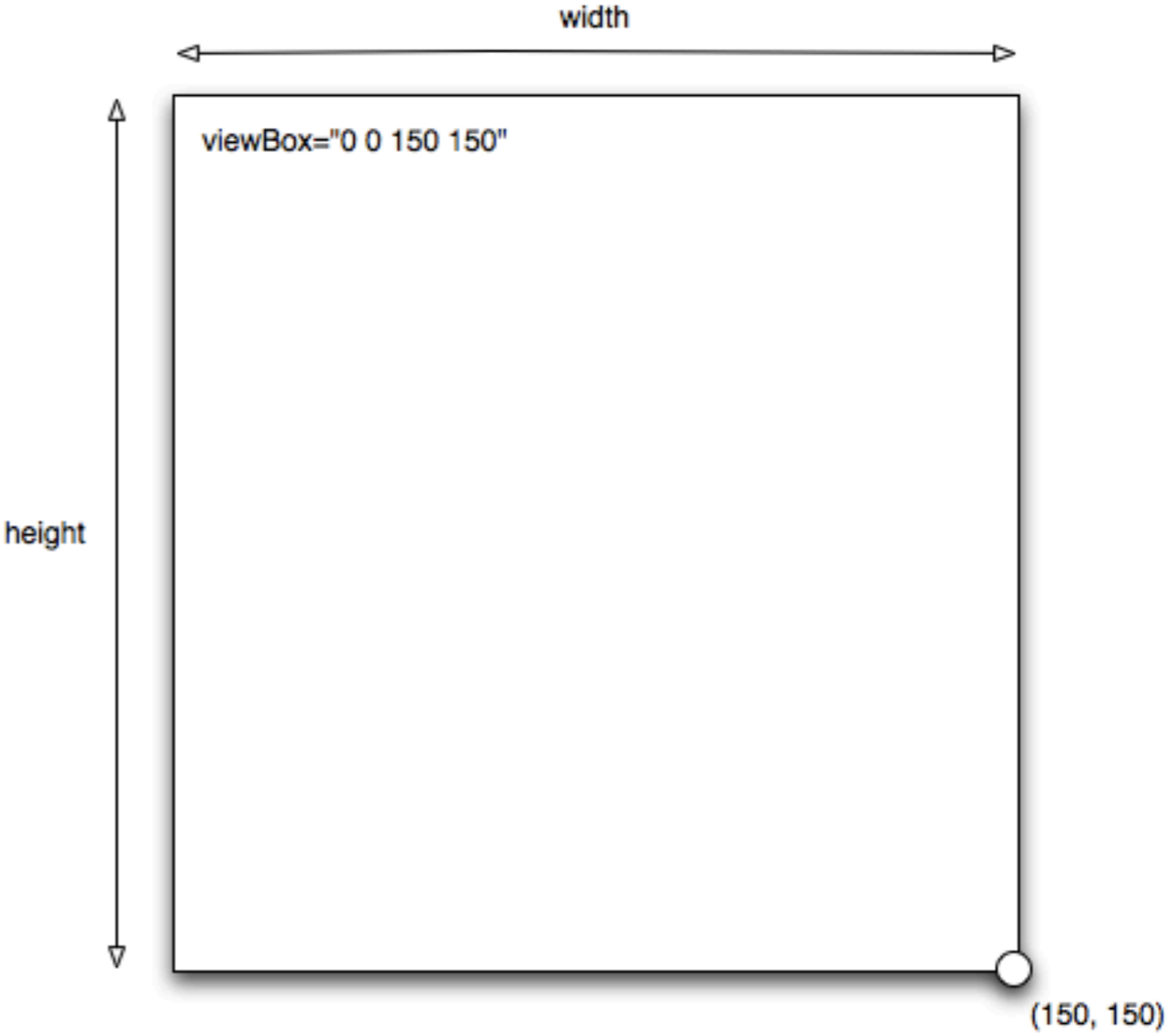
- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

Zooming und Panning
deaktivieren

<svg

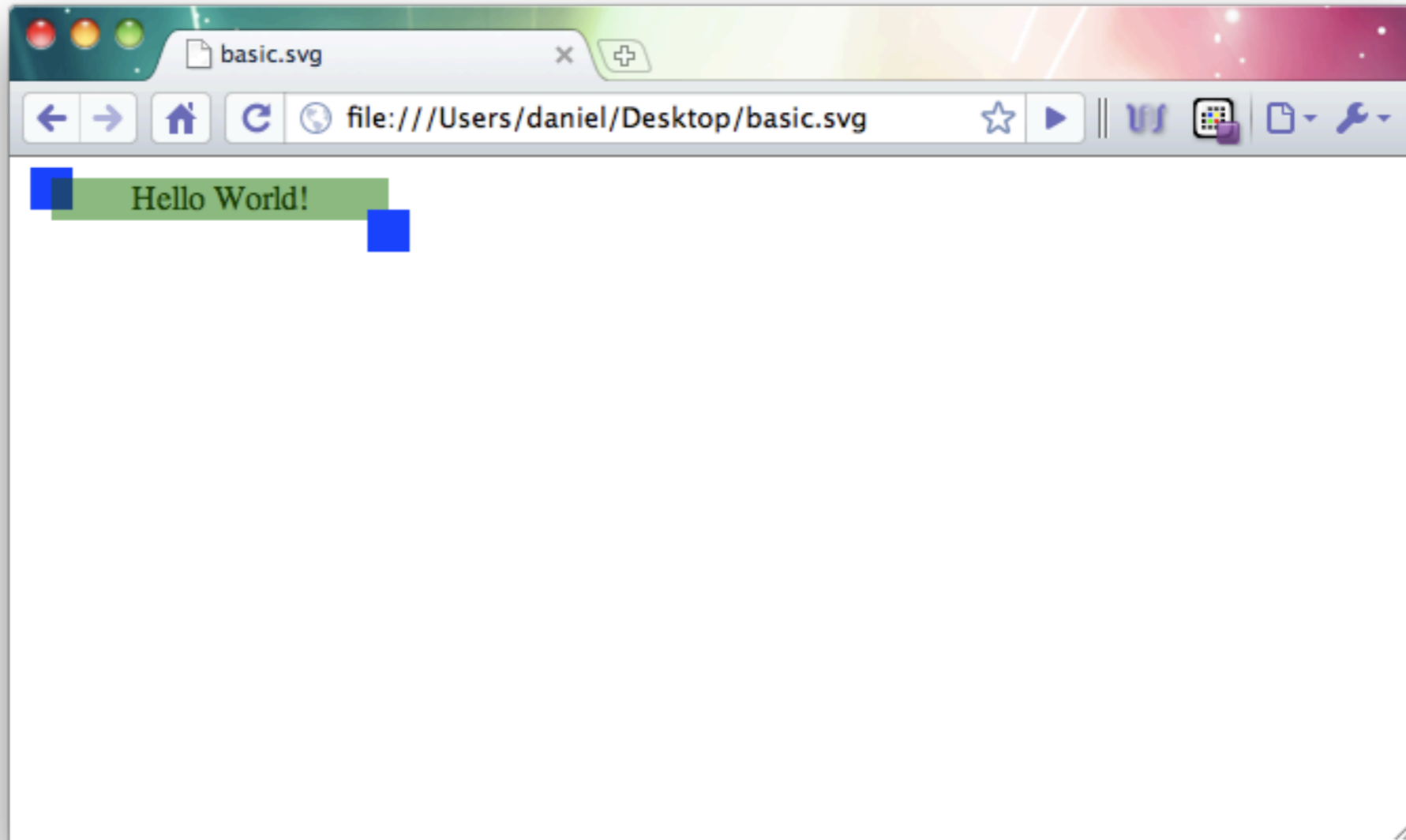
- xmlns="<http://www.w3.org/2000/svg>"
- xmlns:xlink="<http://www.w3.org/1999/xlink>"
- xml:space="preserve"
- width="200px"
- height="50px"
- viewBox="0 0 200 50"
- zoomAndPan="disable" >

width, height und viewBox



Ein paar SVG-Elemente

- `<text style="fill:black; text-anchor:middle;" x="100" y="25">Hello World!</text>`
- `<rect x="20" y="10" width="160" height="20" fill="green" opacity="0.5" />`



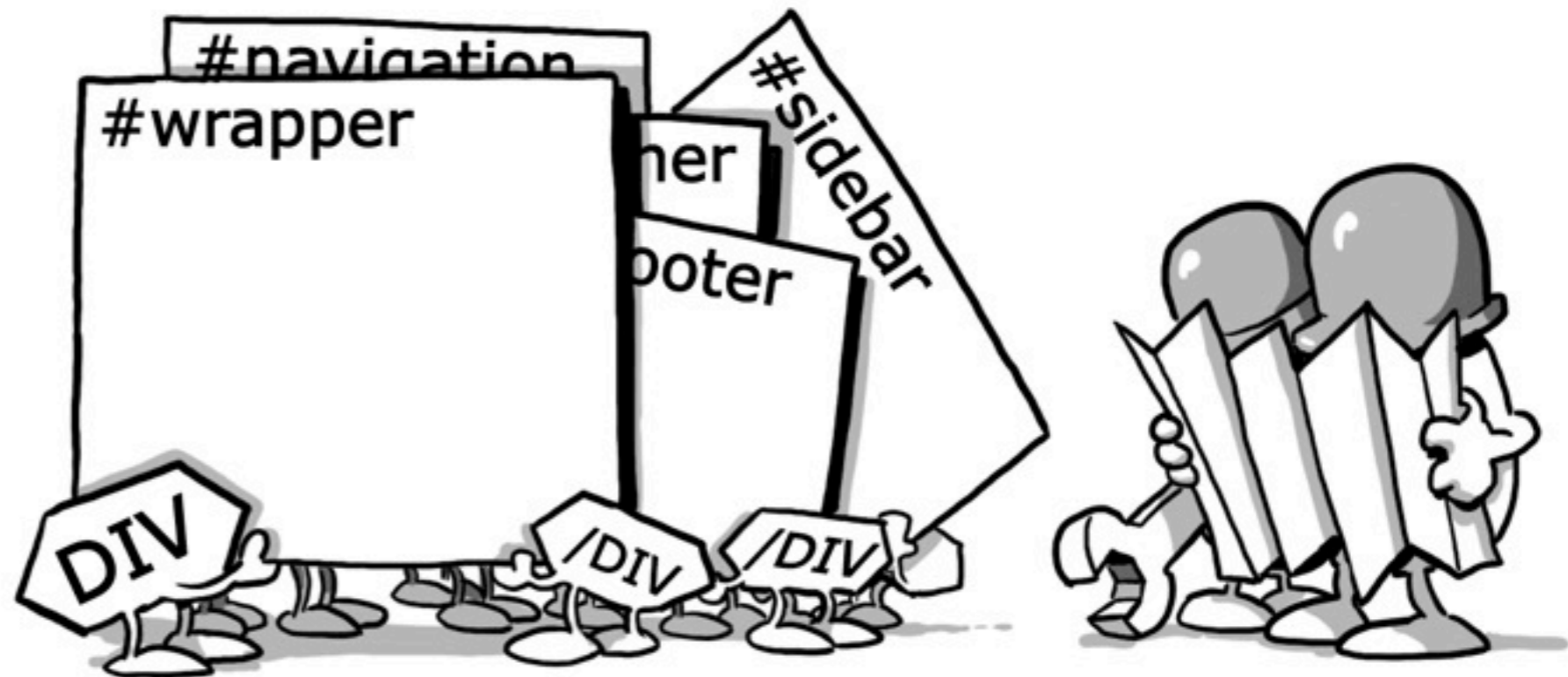
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/
svg10.dtd">
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xml:space="preserve" width="200px"
height="50px" viewBox="0 0 200 50" zoomAndPan="disable" >
  <text style="fill:black; text-anchor:middle;" x="100" y="25">Hello World!</text>
  <rect x="10" y="5" width="20" height="20" fill="blue" />
  <rect x="20" y="10" width="160" height="20" fill="green" opacity="0.5" />
  <rect x="170" y="25" width="20" height="20" fill="blue" />
</svg>
```

Mehr zu SVG: http://apike.ca/prog_svg_intro.html

Fragen?



CSS: Wiederholung & mehr



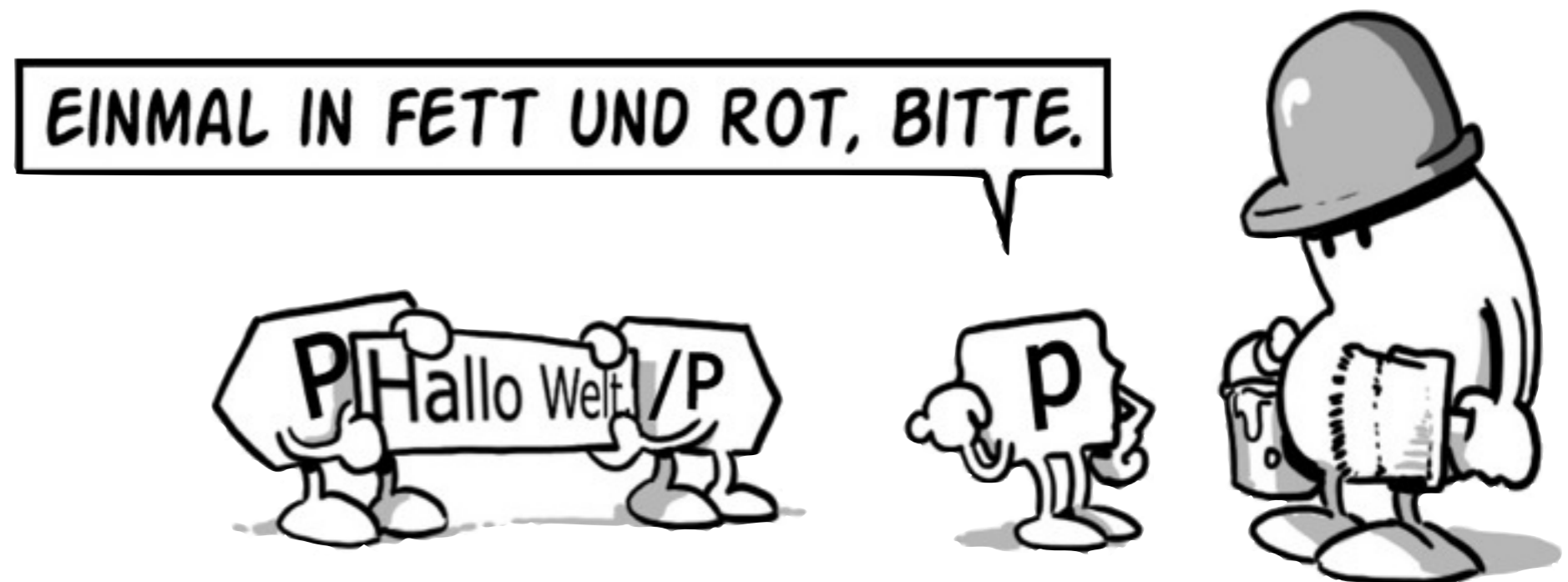
Selektor

- weist einem HTML-Element Layoutanweisungen zu
- selektor{
 eigenschaft:wert;
 eigenschaft: wert wert;
}
- /* Kommentar */

Elementselektoren

- formatieren einen Absatz:

```
p {  
  color:red;  
  font-weight:bold;  
}
```



Kombinierte Selektoren

- Formatieren Elemente die in anderen Elementen vorkommen.

HTML:

```
<h1>Hallo <span>Welt</span></h1>
```

CSS:

```
h1 span{...}
```



Praxisbeispiele

```
<p id="test">
```

```
  Hallo <strong>Welt</strong>
```

```
</p>
```

1) Wie färbt man den Inhalt des Absatzes schwarz?

2) Wie färbt man das Wort „Welt“ grün?

Praxi

p#test

```
<p id="test">
```

```
Hallo <strong>Welt</strong>
```

```
</p>
```

1) Wie färbt man den Inhalt des Absatzes schwarz?

2) Wie färbt man das Wort „Welt“ grün?

Praxi



p#test

```
<p id="test">
```

```
Hallo <strong>Welt</strong>
```

```
</p>
```



p#test strong

1) Wie färbt man den Inhalt des Absatzes schwarz?

2) Wie färbt man das Wort „Welt“ grün?

Praxisbeispiele

```
p#test{
```

```
  color:black;
```

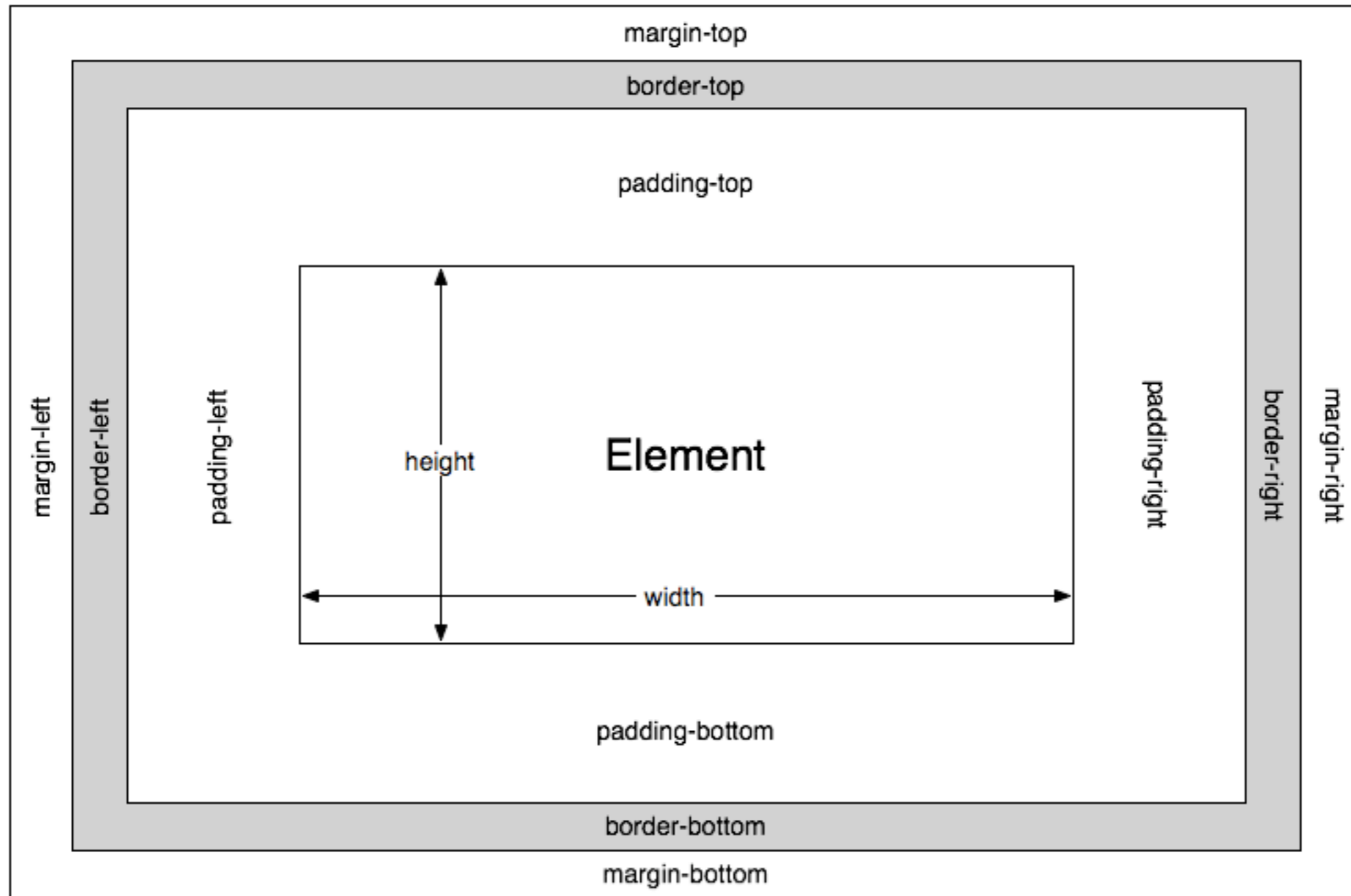
```
}
```

```
p#test strong{
```

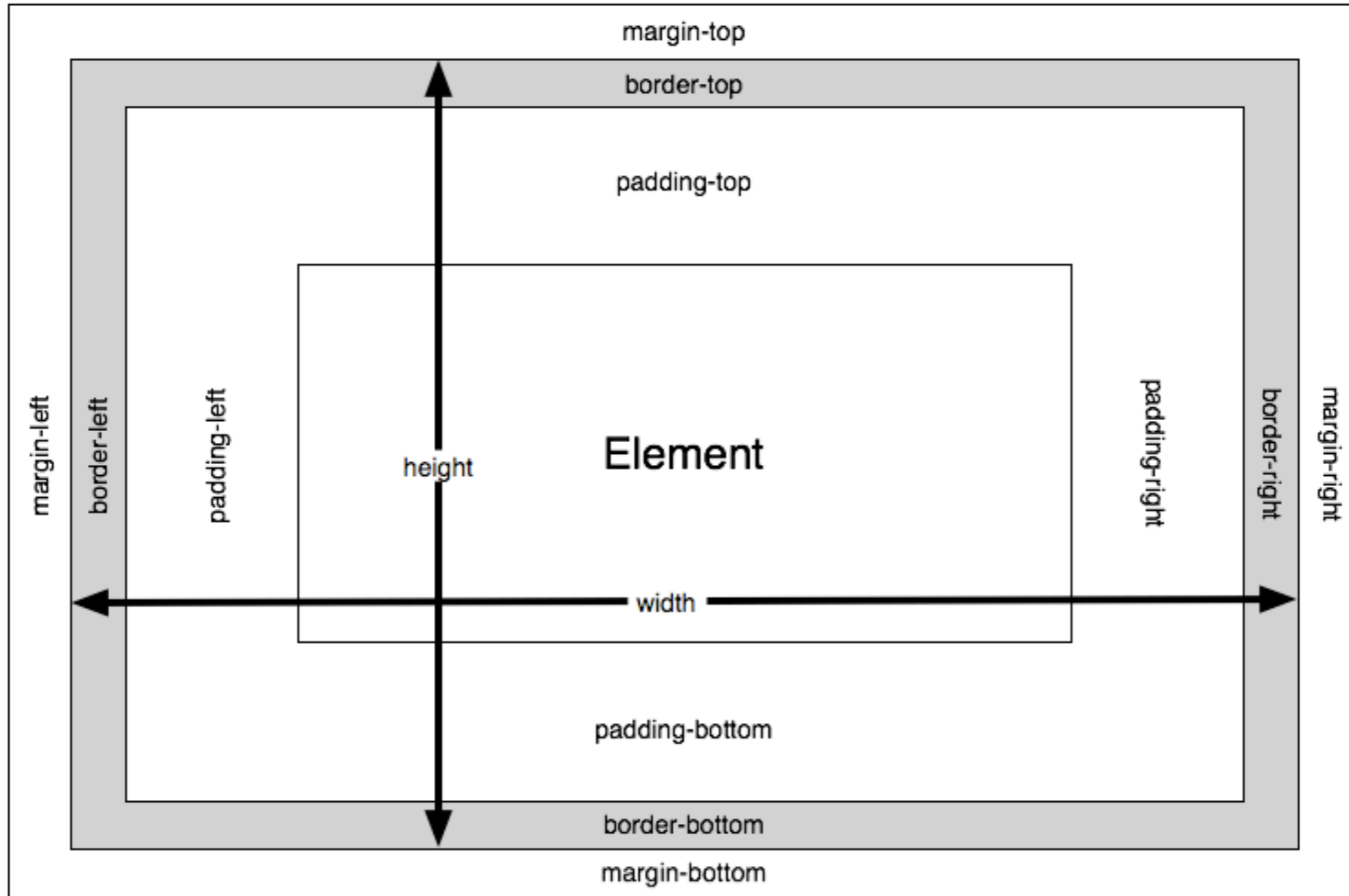
```
  color:green;
```

```
}
```

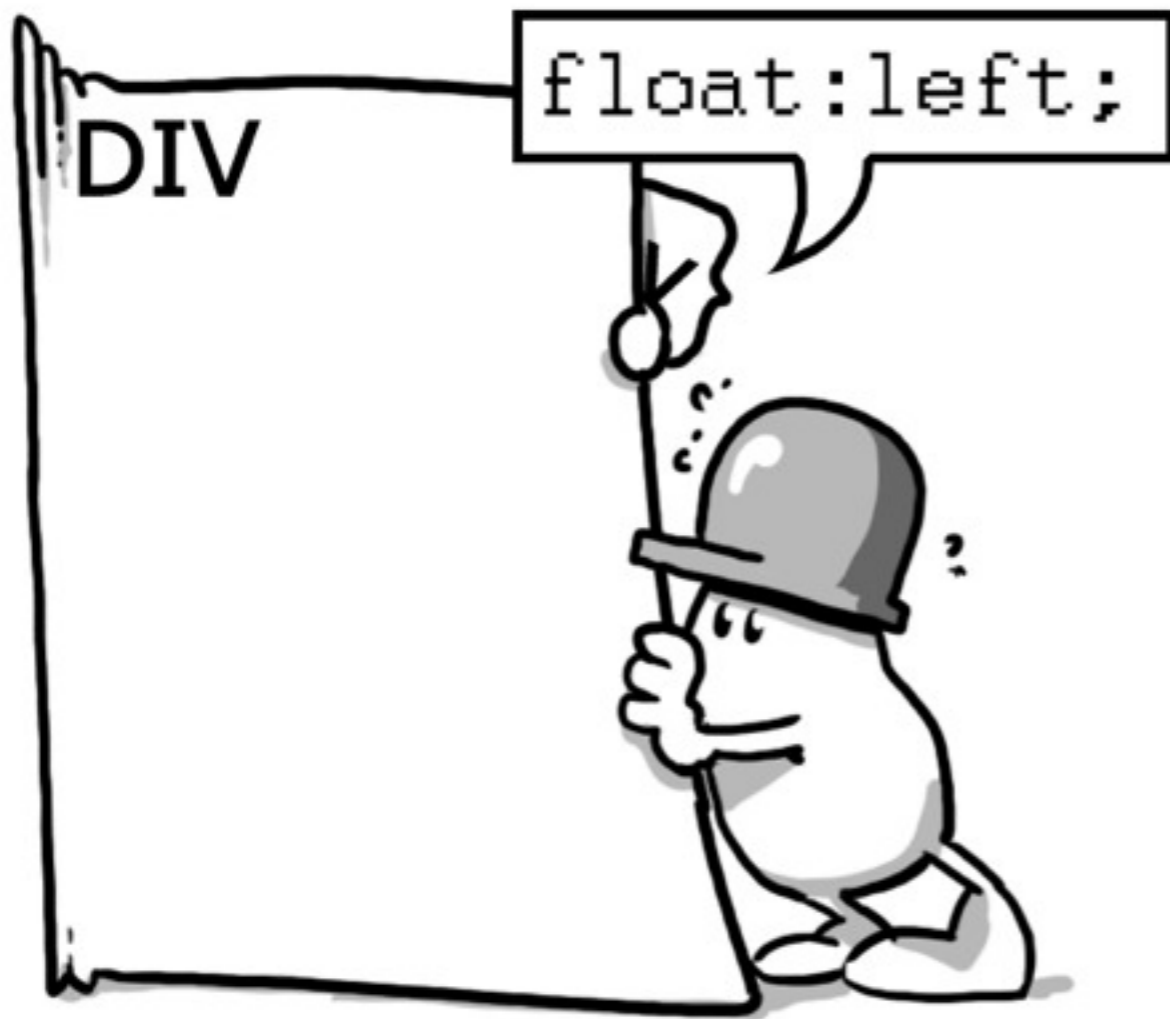
Box-Modell (normale Browser)



Box-Modell (Internet Explorer)



float



Praxis: horizontale Navigation (HTML)

```
<ul id="navigation">
```

```
<li><a href="#" title="title">Link</a></li>
```

```
<li><a href="#" title="title">Link</a></li>
```

```
<li><a href="#" title="title">Link</a></li>
```

```
</ul>
```

Praxis: horizontale Navigation (CSS)

```
#navigation{
```

```
#navigation li{
```

```
float:left;
```

```
width:150px;
```

```
list-style:none;
```

```
}
```

```
#navigation li a{
```

```
display:block;
```

```
padding:10px 15px;
```

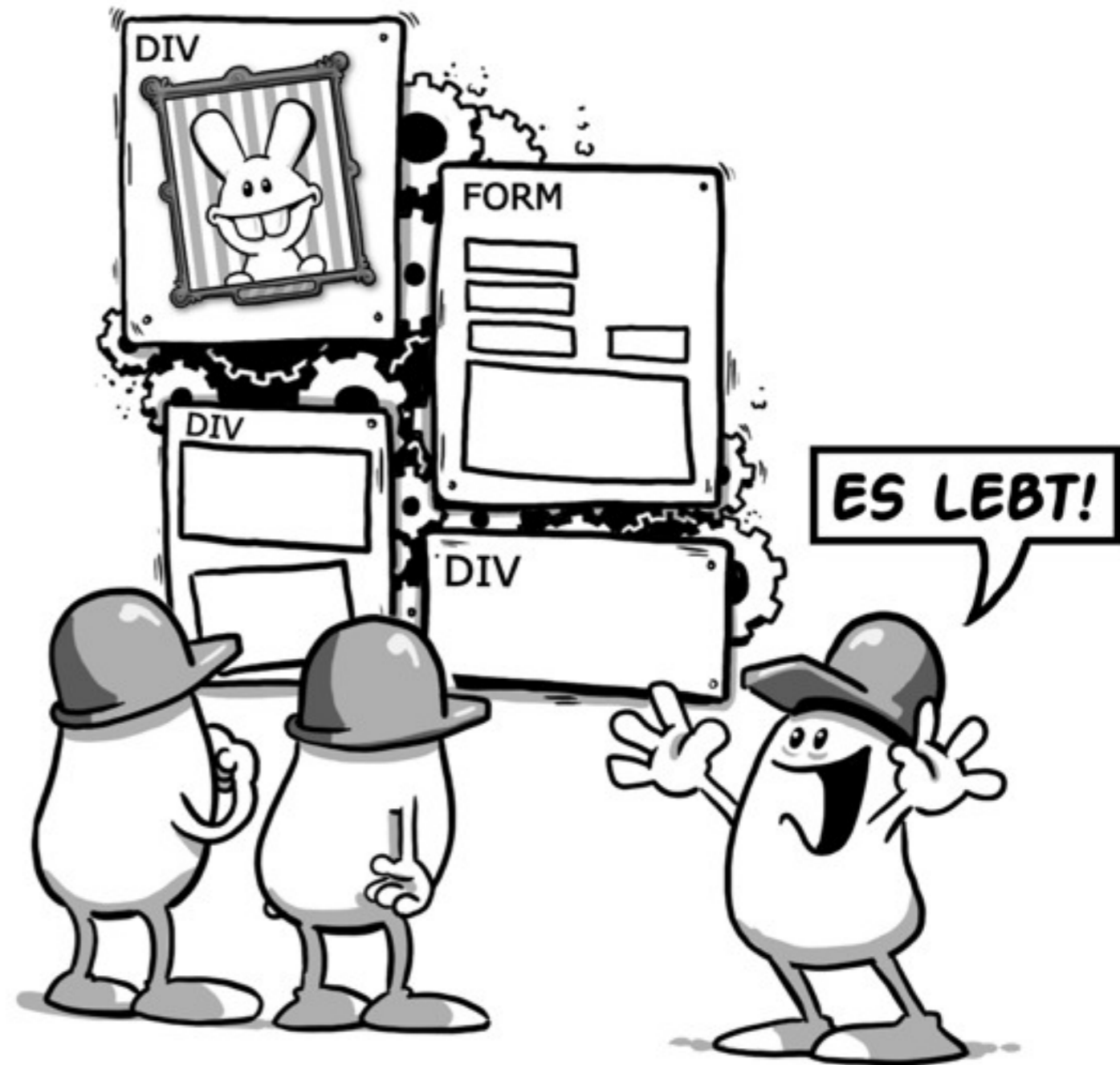
```
}
```

```
/* dem Element nach #navigation  
die CSS-Eigenschaft clear:both  
geben */
```

Fragen?



JavaScript / jQuery



Grundlagen

- Skriptsprache. die hauptsächlich für das DOM-Scripting in Web-Browsern eingesetzt wird
- dynamisch typisiert: Typ einer Variable ergibt sich aus deren Wert und wird, im Gegensatz zu Java, nicht explizit typisiert.
- objektorientiert
- ist **nicht** Java!

DOM - Document Object Model

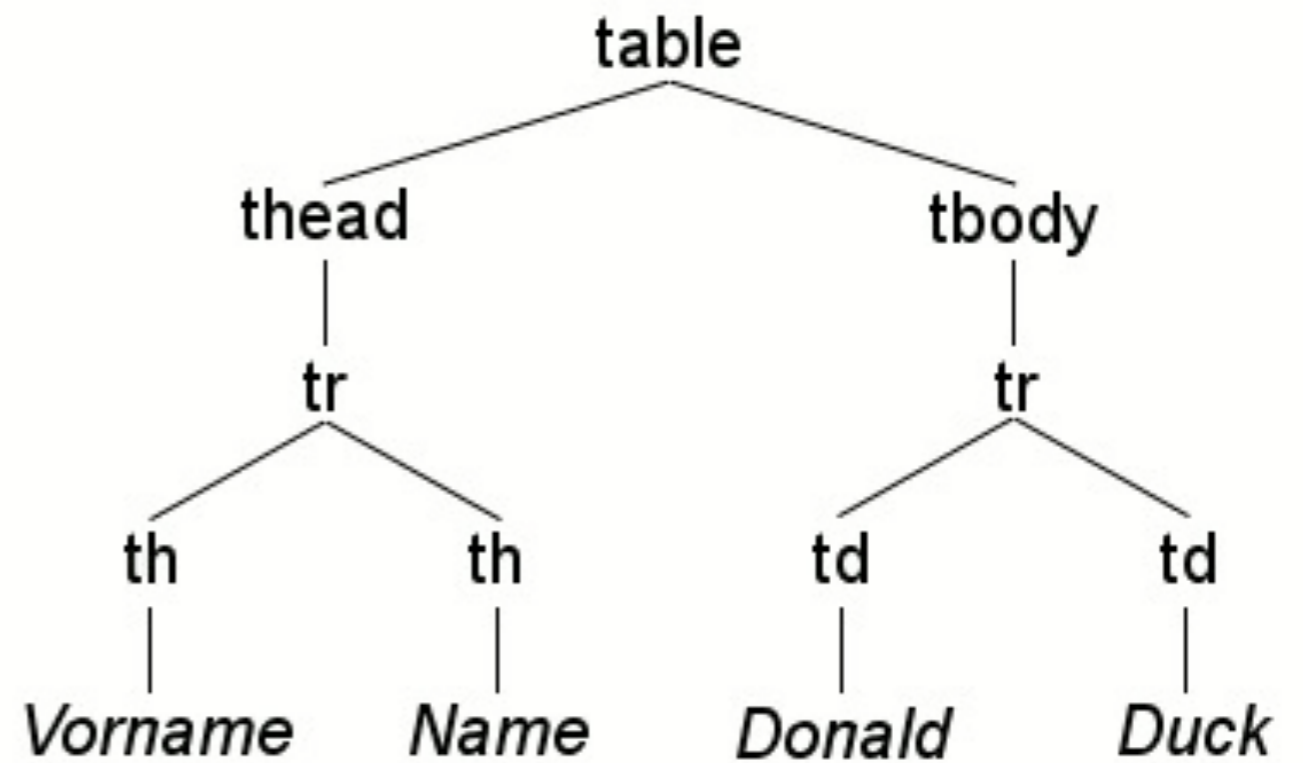
- Das Document Object Model (DOM) ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Sie wird vom World Wide Web Consortium definiert.
- Eine Implementierung, die dieser Spezifikation genügt, besteht im Sinne der objektorientierten Programmierung aus einem Satz von Klassen zusammen mit deren Methoden und Attributen. Sie erlaubt Computerprogrammen, dynamisch den Inhalt, die Struktur und das Layout eines Dokuments zu verändern.

Knoten im DOM

- Dokumentknoten: komplette Baumstruktur
- Dokumentfragmentknoten: ein Teil der Baumstruktur
- Elementknoten: ein HTML Element
- Attributknoten: ein Attribut
- Textknoten: Inhalt eines Elements

DOM

```
<table>
  <thead>
    <tr>
      <th>Vorname</th>
      <th>Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Donald</td>
      <td>Duck</td>
    </tr>
  </tbody>
</table>
```



Hallo Welt

```
<script type="text/javascript">
```

```
<!--
```

```
    alert("Hallo Welt!");
```

```
//-->
```

```
</script>
```

JavaScript einbinden: <script>

- `<script type="text/javascript" src="pfad/zum/script.js"></script>`

Wo `<script>`
einbinden?

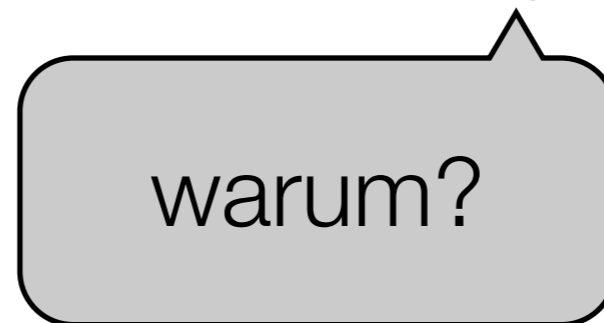


<script> einbinden

- Grundsätzlich überall möglich
- Gefühlte 95% tun dies im <head>
- Wenn es um Geschwindigkeit geht: vor dem </body>-Tag

<script> einbinden

- Grundsätzlich überall möglich
- Gefühlte 95% tun dies im <head>
- Wenn es um Geschwindigkeit geht: vor dem </body>-Tag



JavaScript...

- war lange Zeit unbeliebt und
- ist teilweise sehr umständlich (z.B. „einfacher“ Zugriff nur auf Elemente mit einer ID)
- darum nutzen wir ein JavaScript Framework: jQuery

Framework

- „Programmiergerüst“
- bieten häufig Funktionen, die mit der ursprünglichen Sprache nur schwer / umständlich zu realisieren sind
- stellen in den meisten Fällen eine Vereinfachung dar.

jQuery

- von John Resig
- existiert seit 2008
- wird bei immer, auch großen, Webprojekten verwendet. studiVZ, wer-kennt-wen,...

jQuery Merkmale

- Einfacher Zugriff auf HTML Elemente (u.a. die gleichen Selektoren wie CSS)
- DOM-Manipulation: Objekte hinzufügen und entfernen
- erweitertes Eventsystem (onClick, ...)
- einfache Schleifen
- Effekte und Animationen
- AJAX
- durch Plugins erweiterbar

JavaScript vs jQuery

HTML für die JavaScript-Version

```
<ul id="liste">  
<!-- muss eine ID sein -->  
</ul>
```

```
<span onclick="javascript:addli  
();">ein Listenelement anhängen  
</span>
```

JavaScript

```
function addli(){  
  
    var list =  
    document.getElementById('liste');  
  
    var listitem =  
    document.createElement('LI');  
  
    list.appendChild(listitem);  
  
}
```

JavaScript vs jQuery

HTML für die jQuery-Version

```
<ul class="liste"> <!-- kann ID oder  
Klasse sein -->
```

```
</ul>
```

```
<span class="addlist">ein  
Listenelement anhängen</span>
```

jQuery

```
$(".addlist").click(function(){
```

```
    $("ul.liste").append("<li></li>");
```

```
});
```

jQuery in die eigene Seite einbinden

- Aktuelle Version von www.jquery.com laden
- JavaScript-Datei mit script-Element einbinden
- Hinter diesem Element eine weitere script-Datei einbinden mit eigenem JavaScript-Code

```
<script type="text/javascript" src="js/jquery.js"></script>
```

```
<script type="text/javascript">
```

```
    // Hier kommt unser Code hin ...
```

```
</script>
```


Fragen?



Ein einfaches jQuery Beispiel

```
$(document).ready(function(){
```

```
  $("a").click(function(){
```

```
    alert("Auf Wiedersehen");
```

```
  });
```

```
});
```

\$() - Die jQuery-Funktion

- `$("selektor").aufgabe();`
- selektor: Vor allem die aus CSS bekannten Selektoren
- `$("a")` - gilt für alle Links im DOM. Erzeugt intern ein Objekt
- Dieses Objekt kann mit mehreren Funktionen verbunden werden.
- `$("a").aufgabe();` erzeugt dann ein weiteres Objekt usw:
- `$("a").aufgabe().aufgabe2();`

Beispiele

```
$("#a.hideMe").click(function(){  
  
    $(this).hide("slow");  
  
    return false;  
  
});
```

this bezieht sich, analog zu Java, auf das aktuelle Objekt. In diesem Fall alle Links mit der Klasse hideMe.

Beispiele

```
$("#a").click(function(){
```

```
    $(this).addClass("test").html("hier wurde schon geklickt");
```

```
    return false;
```

```
});
```

Fragen?



Vielen Dank für Eure Aufmerksamkeit!