

# DOM-Manipulation



Jonas Leonhard, Tammo Heppner

# Was ist “DOM-Manipulation”?

Document Object Model als nodeTree/Elementen

Objekt Orientierte Representation des Htmls

JavaScript kann benutzt werden um das DOM zu lesen/schreiben/manipulieren

# Was kann man damit machen?

Selection (byId,...)

Navigating the Dom (Parents, Children, Siblings)

Changing the Dom (Styles, Text, Html...)

Creating in the Dom (createElement, newDiv.className,...)

Add Events (addEventListener)

# Selection

`console.dir(document);`



# Selection

document.domain → Domain name

document.URL → URL

document.title → Page title

```
> console.log(document.title);  
Labor 01
```

document.head → head element

document.all → html collection

```
> document.all;  
< HTMLAllCollection(30) [html, head, meta, meta, meta, meta, link, title, body, div.wrapper, div.item01,  
img.img-left, div.item02, div.item02-line-right, div.item02-line-left, div.item02-text, div.item02-coloredtxt,  
span.special-word, div.item02-line-bot, div.item03, div.item04, div.item05, p, p, p, div.item06, div.item07,  
img.img-right, div.item07-blue-line-top, div.item07-blue-line-bot, viewport: meta, Description: meta, author:  
meta]
```

# Selection

`console.log(document.links);` → google docs links

*weiter Möglichkeiten wären:*

*document.forms,*

*document.images, ...*

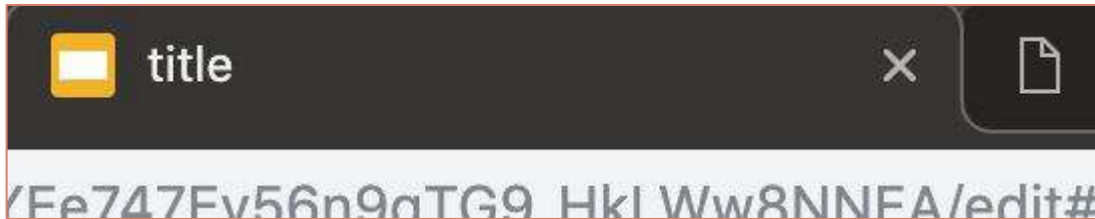
[illegible]

# Selection

document.all → html collection

```
> document.all[12].textContent = 'title'  
◀ "title"
```

*Schlechte Art zu selektieren - bei Veränderung des htmls verändert sich der Index*



# Selection (getElementById())

```
document.getElementById('item01');
```

```
> console.log(document.getElementById("item01"));  
▶ <div id="item01" class="item01">...</div>
```

*Nimmt das Erste Element von allen Ids mit der id 'item01'*



# Selection (getElementsByClassName())

```
var items = document.getElementsByClassName('item02');
```

```
> var items = document.getElementsByClassName('item02');
```

```
< undefined
```

```
> console.log(items);
```

```
▶ HTMLCollection [div.item02]
```

VM702:1

*Elements htmlCollection von Elementen mit Klassen Name 'item02';*

```
> items[0];
```

```
< ▶ <div class="item02">...</div>
```

# Selection (getElementByTagName())

```
> document.getElementsByTagName('div');  
< HTMLCollection(15) [div.wrapper, div#item01.item01,  
  div.item02, div.item02-line-right, div.item02-line-left,  
  div.item02-text, div.item02-coloredtxt,  
  ► div.item02-line-bot, div.item03, div.item04, div.item05,  
  div.item06, div.item07, div.item07-blue-line-top,  
  div.item07-blue-line-bot, item01: div#item01.item01]
```

oder andere: ('li'), ('p') ...

# Selection (querySelector())

var header = document.querySelector('header'); → tags

var header = document.querySelector('.header'); → class

var header = document.querySelector('#header'); → ids

```
> document.querySelector('.item01');  
< ▶ <div id="item01" class="item01">...</div>
```

*alle Css Selektoren möglich - gibt nur das erste vorkommende Element*

# Selection (querySelector())

```
var submit = document.querySelector('input[type="submit"]');
```

```
> var submit = document.querySelector('.input[type="submit"]');  
< undefined  
> console.log(submit);  
    <input class="input" type="submit">
```

*alle Css Selektoren möglich - gibt nur das erste vorkommende Element*

```
> document.querySelector('.list  
-group-item:nth-child(2)');  
< <li class="list-group-  
item">item02</li>
```

*oder :last-child, first-child ...*

# Selection (querySelectorAll())

```
var divs = document.querySelectorAll('div');
```

```
> document.querySelectorAll('div');  
< NodeList(16) [div.wrapper,  
  div#item01.item01, div.item02,  
  div.item02-line-right,  
  div.item02-line-left, div.item02-text,  
  ▶ div.item02-coloredtxt,  
  div.item02-line-bot, div.item03,  
  div.item04, div.item05, div.item06,  
  div.item07, div.item07-blue-line-top,  
  div.item07-blue-line-bot, div.lists]
```

*gibt alle vorkommende Elemente*

# Navigation (Parents, Child, Sibling)

```
<div class="list">  
<li class="list-group-item">item01</li>  
<li class="list-group-item">item02</li>  
<li class="list-group-item">item03</li>  
<li class="list-group-item">item04</li>
```

*list* ist **Parent** von *list-group-item*

*list-group-item* ist **Child** von *list*

*list-group-item* ist **Sibling** zu *list-group-item*

# Navigation (Parents)

.parentElement

```
> var listgroupitem = document.querySelector('.list-group-item');  
< undefined  
  
> listgroupitem.parentElement  
< ▶ <div class="lists">...</div>  
  
> listgroupitem.parentElement.parentElement  
< ▶ <body>...</body>
```

# Navigation (Children)

.children

```
> divs.children  
< HTMLCollection(9) [div#item01.item01, div.item02, div.item03, div.item04,  
  ► div.item05, div.item06, div.item07, div.item07-blue-line-top,  
    div.item07-blue-line-bot, item01: div#item01.item01]
```

*divs.firstElementChild* → für erstes Child



# Navigation (Siblings)

`.nextSibling`

`.previousSibling`

`.nextElementSibling`

`.previousElementSibling`

# Changing

```
<html>
<body>

<p id="p1">Hello World!</p>
```

```
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>

</body>
</html>
```

Beliebiges Element aus dem HTML Document verändern.

← In dem Fall wird der HTML Text aus p mit der ID “p1” auf “New text!” gesetzt.

# Changing (Value)

```
<!DOCTYPE html>
<html>
<body>



<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>

</body>
</html>
```

---

`document.getElementById(id).attribute = new value`

# Changing (Styles)

```
<p id="p2">Hello World!</p>
```

```
<script>
```

```
document.getElementById("p2").style.color = "blue";
```

```
</script>
```

```
<h1 id="id1">My Heading 1</h1>
```

```
<button type="button"
```

```
onclick="document.getElementById('id1').style.color = 'red'">
```

```
Click Me!</button>
```

# Events

```
<!DOCTYPE html>
<html>
<body>

<h1 onclick="changeText(this)">Click on this text!</h1>

<script>
function changeText(id) {
  id.innerHTML = "Ooops!";
}
</script>

</body>
</html>
```

---

# Event (addEventListener)

```
<h2>JavaScript addEventListener()</h2>
```

```
<p>This example uses the addEventListener() method to add many events on the same button.</p>
```

```
<button id="myBtn">Try it</button>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var x = document.getElementById("myBtn");  
x.addEventListener("mouseover", myFunction);  
x.addEventListener("click", mySecondFunction);  
x.addEventListener("mouseout", myThirdFunction);
```

```
function myFunction() {  
  document.getElementById("demo").innerHTML += "Moused over!<br>";  
}
```

```
function mySecondFunction() {  
  document.getElementById("demo").innerHTML += "Clicked!<br>";  
}
```

```
function myThirdFunction() {  
  document.getElementById("demo").innerHTML += "Moused out!<br>";  
}  
</script>
```

Try it

Moused over!  
Moused out!  
Moused over!  
Moused out!  
Moused over!  
Clicked!  
Clicked!  
Clicked!  
Moused out!  
Moused over!  
Moused out!

# Creating

```
var a = document.createElement("a");  
a.href = "http://thefair.com/";  
a.textContent = "thefair.com";  
  
document.body.appendChild(a);
```

Beliebiges HTML Element erschaffen und zuweisen.

Hier wird es einfach als letztes Element im Body angesetzt.

# Creating

```
// create a new div for each container div
var chat = document.createElement("div");
var info = document.createElement("div");

// create elements inside the info container
var img = document.createElement("img");
var user = document.createElement("div");
var time = document.createElement("div");

// create the div for the simple chat message
var msg = document.createElement("div");
```

```
img.classList.add("profile-pic");
img.src = "netizen42.png";

user.classList.add("username");
user.textContent = "netizen42";

time.classList.add("timestamp");
time.textContent = "10:34 PM";

msg.classList.add("message");
msg.textContent = "I'm hacking into the mainframe now. You better be ready.";
```



# Creating

```
// add the image, the user and the timestamp to the info container
info.appendChild(img);
info.appendChild(user);
info.appendChild(time);

chat.classList.add("chat-message");

// add the info container and the message to the total chat container
chat.appendChild(info);
chat.appendChild(msg);

// finally, attach the entire new structure to some container on the page.
var parent = document.getElementById("some-container");
parent.appendChild(chat);
```

