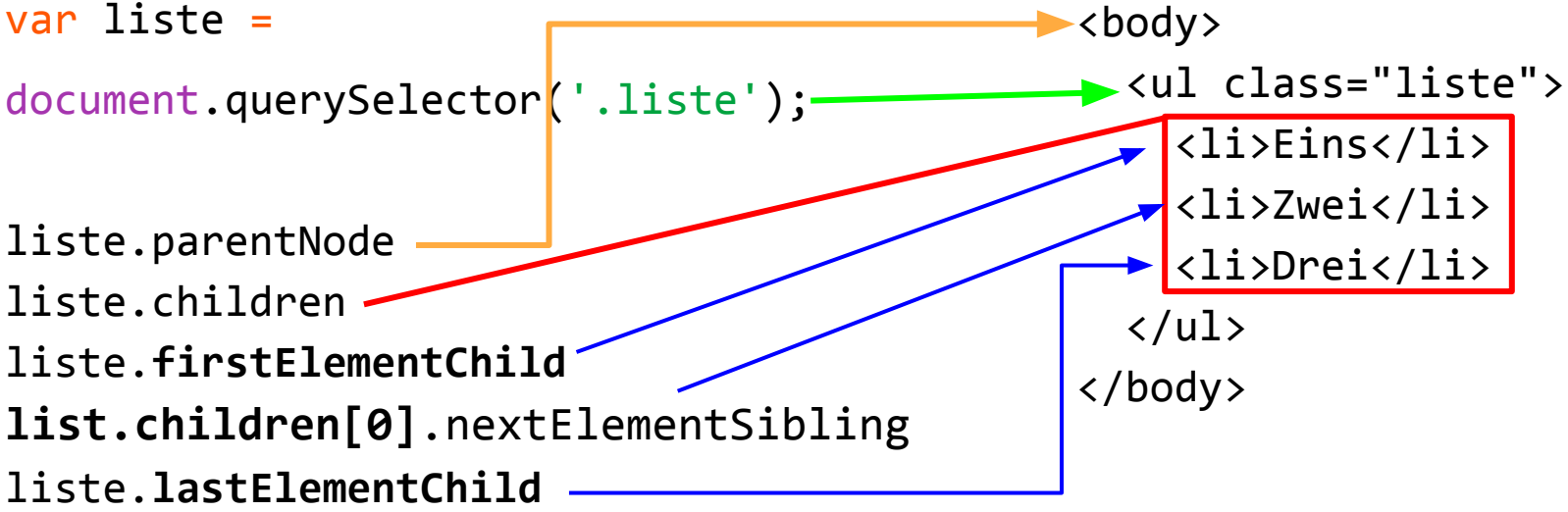
Kurze Repetition:

Javascript
DOM & Events

DOM

Weitere Attribute: Die DOM-Familie



neue DOM Elemente erstellen

```
// Element wird erzeugt
var newElement = document.createElement('div');

// Style properties des Elements anpassen
newElement.innerText = "Das ist der Text im Div";

// Element an bestimmtes Element hängen
var ziel = document.querySelector('#ziel');
ziel.appendChild(newElement);
```

DOM Elemente entfernen

```
// Ein bestimmtes Element entfernen, ohne den Elternknoten zu kennen
var node = document.querySelector("li");
if (node.parentNode) {
  node.parentNode.removeChild(node);
}
```

Klasse hinzufügen (classList)

```
// Element wird erzeugt
var newElement = document.createElement('div');
newElement.classList.add('klassen-name');
document.body.appendChild(newElement);

// Mit bestehendem Element
var title = document.querySelector('h1');
title.classList.add('klassen-name');
```

Klasse entfernen (classList)

```
// Element wird erzeugt
var newElement = document.createElement('div');
newElement.classList.remove('klassen-name');
document.body.appendChild(newElement);

// Mit bestehendem Element
var title = document.querySelector('h1');
title.classList.remove('klassen-name');
```

Klasse "togglen" (classList)

```
var element = document.querySelector('div');
element.classList.toggle('rot');
element.classList.toggle('rot');
element.classList.toggle('rot');
element.classList.toggle('rot');
```

```
<div class="rot"></div>
<div></div>
<div class="rot"></div>
<div></div>
<div class="rot"></div>
```

Timeout und Interval

```
window.setInterval(myCallback, 2000);
function myCallback() {
    document.querySelector('div').classList.toggle('rot');
}
```

(nach 2 Sekunden)

(nach 4 Sekunden)

(nach 6 Sekunden)

Werte von Eingabefeldern lesen & setzen

```
var inputElem = document.querySelector('input');

console.log(inputElem.value); // "5000"

inputElem.value = 1337;
```

Einbettung (inline)

In <head> (schlecht)

```
<html>
  <head>
    <title>JS Repetition</title>
    <script>
       // Javascript kommt hier
    </script>
  </head>
  <body>
    <h1>Titel</h1>
  </body>
</html>
```

Ende <body> (gut)

```
<html>
  <head>
    <title>JS Repetition</title>
  </head>
  <body>
    <h1>Titel</h1>
    <script>
       // Javascript kommt hier
    </script>
  </body>
</html>
```

Beispiele Events

change	Wert auf ein Eingabefeld wurde geändert
click	Ein Element wurde angeklickt
mouseover	Der Mauszeiger wurde über ein Element geführt
mouseout	Der Mauszeiger hat das Element wieder verlassen
keydown	Eine Taste auf dem Eingabefeld wurde gedrückt
keyup	Eine Taste auf dem Eingabefeld wurde losgelassen

Event Listeners (Ideal)

```
<button id="myButton"></button>
<script>
function doSomething() {
    console.log('button clicked');
}
var myButton = document.querySelector('#myButton');
myButton.addEventListener('click', doSomething);
</script>
```

Event Listeners (Kurzform)

```
<button id="myButton"></button>
<script>

var myButton = document.querySelector('#myButton');
myButton.addEventListener('click', function() {
    console.log('button clicked');
});
</script>
```

Event Listeners (Super Kurzform)

```
<button id="myButton"></button>


<script>
document.querySelector('#myButton')
.addEventListener('click', function() {
    console.log('button clicked');
});
</script>
```

Fragen zu Übungen?

Übungen zum DOM

Kopiert aus dem Google Drive Datei aus:
Module/Javascript/lektion4.zip in euer **htdocs** Verzeichnis in einen eigenen Ordner und macht:

- uebung_1_3.html
- uebung_1_4.html
- uebung_2_0.html
- uebung_2_1.html
- uebung_2_2.html
- uebung_2_3.html
- uebung_2_4.html

Ein Teil der Übungen ist aus der letzten Lektion, mach nur die Übungen, welche du noch nicht gemacht hast.

Verwende aber das neue lektion4.zip, es gab Änderungen

Javascript:
Events strike back

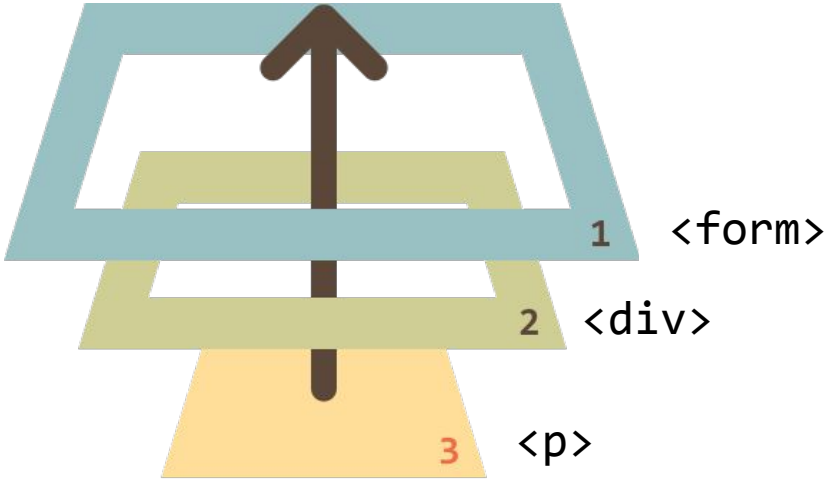
Interessante Events

- **online Event**
- **orientationchange Event**
- resize
- load
- scroll

Event Bubbling

Event Bubbling

- Events steigen von unten nach oben
- Genau gleich wie Wasserblasen
- Deshalb heisst es "bubbling"
- Bubbling kann gestoppt werden




```
<form onclick="alert('form')">
   <div onclick="alert('div')">
     <p onclick="alert('p')"></p>
   </div>
 </form>
```

Bubbling Demo

Event Objekte

Event Objekte

```
element.addEventListener('click', function (event) {
    console.log(event)
}
```

- Eure Event Funktion erhält ein Event Objekt
- Dieses enthält informationen über den ausgelösten Event
- Je nach Event-Typ sind unterschiedliche Informationen enthalten

Event Objekte Demo

Basis Attribute von Event Objekten

type	Typ des Events (click, focus, blur, etc...)
target	**Element auf welches der Event ausgelöst wurde**
currentTarget	Element zu welchem der Event aufgestiegen ist
bubbles	Kann der Event aufsteigen?

MouseEvent

event.button	Welcher Knopf auf der Maus wurde auf der Maus gedrückt (bei click)
event.clientX	X koordinate der Maus (DOM Koordinaten).
event.clientY	Y koordinate der maus (DOM Koordinaten).
event.screenX	X koordinate der Maus (Bildschirm Koordinaten).
event.screenY	Y koordinate der maus (Bildschirm Koordinaten).

Keyboard Event

event.key	Taste die gedrückt wurde
event.code	Code der Taste die gedrückt wurde

Beispiele:

a	key = 'a'	code = 'KeyA'
J	key='J'	code = 'KeyJ'
Enter	key = 'Enter'	code = 'Enter'
ShiftL	key='Shift'	code = 'ShiftLeft'
ShiftR	key='Shift'	code = 'ShiftRight'
Enter	key='Enter'	code = 'Enter'
Backsp	key='Backspace'	code = 'Backspace'

Input Event

event.data	Eingegebener Buchstabe	
event.inputType	Art der Eingabe z.B:	
	"insertText"	normales Tippen
	"insertFromPaste"	Einfügen
	"deleteByCut"	Ausschneiden

Keycodes auslesen

```
var input = document.querySelector('input');

input.addEventListener("keydown", function(event) {
    console.log("Key: " + event.key + " Code: " + event.code);
});
```

Keyboard Event Demo

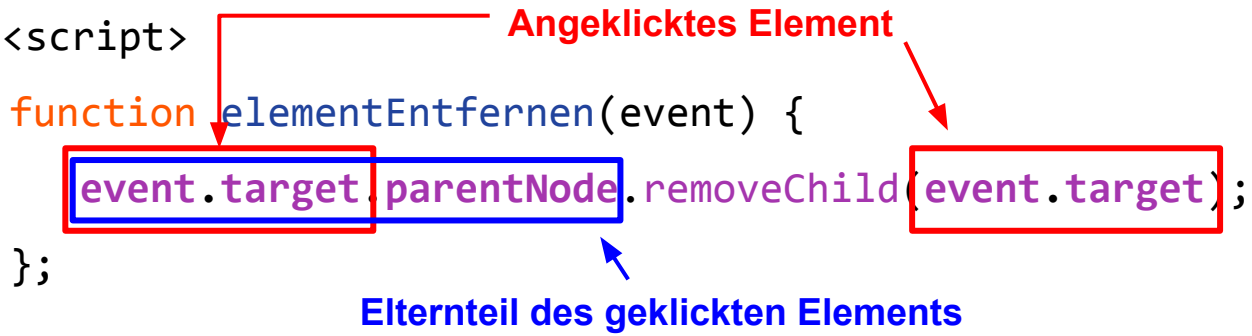
Event Target

```
var input = document.querySelector('button');

input.addEventListener("click", function(event) {
    console.log(event.target);
});
```

Elemente entfernen mit Event Target

```
<button>Entfern mich!</button>
```

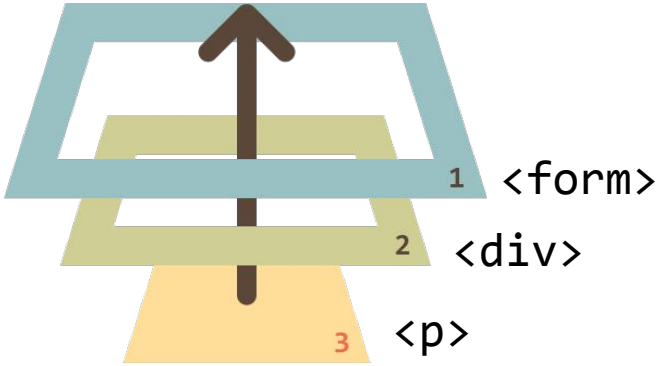



```
var button = document.querySelector('button');
buttons.addEventListener('click', elementEntfernen);
</script>
```

Bubbling Stoppen

```
document.querySelector('form').addEventListener('click', function (e) {
  e.stopPropagation();
  console.log('form');
});
document.querySelector('div').addEventListener('click', function (e) {
   e.stopPropagation();
   console.log('div');
});
document.querySelector('p').addEventListener('click', function (e) {
   e.stopPropagation();
   console.log('p');
});
```

```
<form>
   <div>
      <p></p>
   </div>
</form>
```



Default Events Stoppen

```
<a href="http://www.google.ch" id="popup">Google</a>
```

```
var link = document.querySelector('#popup')
link.addEventListener('click', function (event) {
    event.preventDefault();
    open(event.target.href, 'popup-beispiel','height=400,width=400,resizable=no');
});
```

Übung

Nimm deine TODO Liste aus dem CSS Modul und bau ein, dass du neue Elemente in die Liste eintragen kannst

Tipps:

- Damit es mit Enter Funktioniert brauchst du KeyCodes