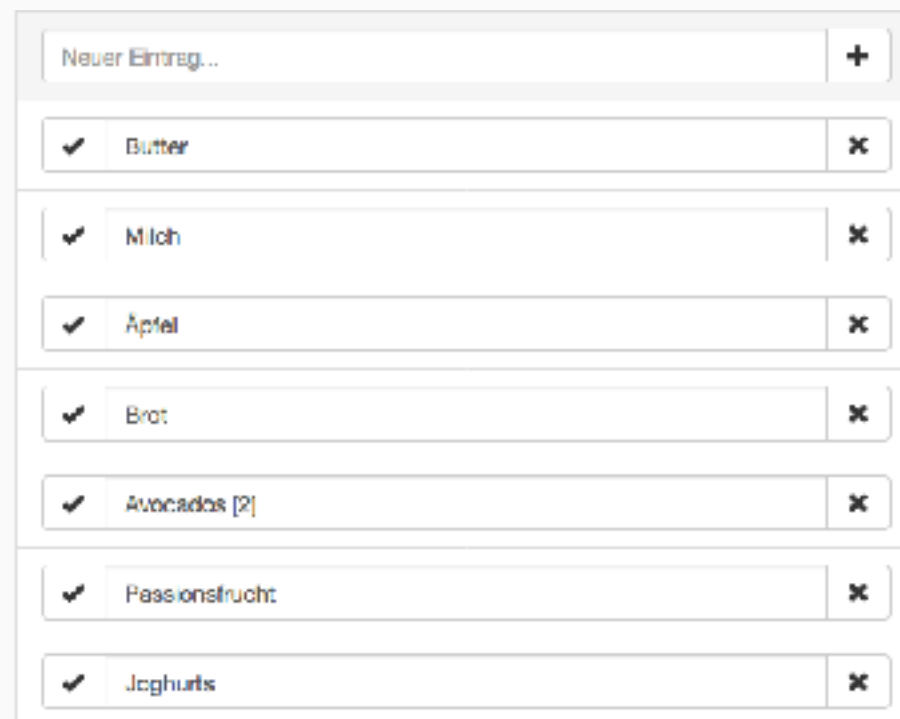


# JavaScript

## Formularinteraktion I



The image shows a web form with a list of items. At the top is a text input field labeled "Neuer Eintrag..." with a "+" button to its right. Below this is a list of seven items, each in a row. Each row contains a checked checkbox, the item name, and a delete button (marked with an 'x'). The items are: Butter, Milch, Apfel, Brot, Avocados [2], Passionsfrucht, and Joghurts.

Item	Checkbox	Delete
Neuer Eintrag...		+
✓ Butter	✓	x
✓ Milch	✓	x
✓ Apfel	✓	x
✓ Brot	✓	x
✓ Avocados [2]	✓	x
✓ Passionsfrucht	✓	x
✓ Joghurts	✓	x

# Fragen zu Codecademy?

# Aufgabe zur Formularvalidierung

# Konsole

# Knackpunkte

# Neue Zeichenketten

# Neue Zeichenketten

```
var university = 'HTW Chur';  
var intro = 'Willkommen an der ' + university;
```

```
// intro: Willkommen an der HTW Chur
```

# Neue Zeichenketten

```
var university = "HTW Chur";  
var intro = "Willkommen an der " + university;
```

```
// intro: Willkommen an der HTW Chur
```



# Neue Zeichenketten

```
var university = 'HTW Chur';  
var intro = `Willkommen an der ${ university }`;
```

```
// intro: Willkommen an der HTW Chur
```

# Neue Zeichenketten



# Neue Zeichenketten

```
var university = 'HTW Chur';  
var studies = 'Multimedia Production';  
  
var intro = `Willkommen an der ${ university }  
im Studiengang ${ studies }`;  
  
/* intro:  
  
Willkommen an der HTW Chur  
im Studiengang Multimedia Production  
*/
```

.click vs .on('click')



**HTW Chur**

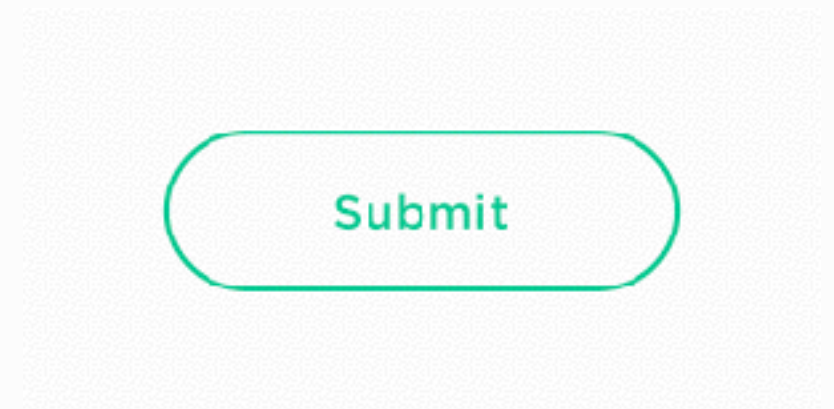
Hochschule für Technik und Wirtschaft  
University of Applied Sciences

# .click vs .on('click')

```
$('.button').click(function() {  
...  
});
```

```
$('.button').on('click', function() {  
...  
});
```

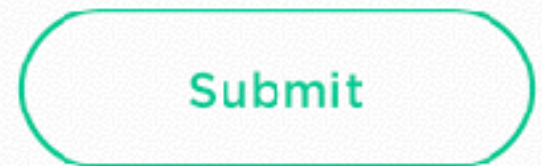
// Wo liegt der Unterschied?



# .click vs .on('click')

// Kein Unterschied in diesem Fall!

// .on ist in manchen Fällen vielseitiger



# Events für neue Elemente

# Events für neue Elemente

```
<div class="box">  
  <!-- Hierhin kommt der Button -->  
</div>
```

```
$('.button').on('click', function() {  
  // Hier sollte etwas passieren  
});
```





# Events für neue Elemente

```
<div class="box">  
  <button class="delete"></button>  
</div>
```

```
$('.button').on('click', function() {  
  // Hier sollte etwas passieren  
});
```



# Events für neue Elemente

```
<div class="box">  
  <button class="delete"></button>  
</div>
```

```
$('.button').on('click', function() {  
  // Die Funktion wird NICHT aufgerufen  
});
```



# Events für neue Elemente

```
<div class="box">  
  <button class="delete"></button>  
</div>
```

```
$( '████████' ).on( '████████', '████████',  
function() {  
  // Die Funktion wird nun aufgerufen  
});
```

A: .delete

B: .box

C: click



# Fragen zu den Videos?

# Aufgaben

# Aufgabe 1

## Fehlersuche

Im Zip "fehlersuche" auf Moodle befinden sich einige Fehler im JavaScript. Korrigieren Sie die Fehler und bringen Sie die Todo-Liste zum laufen.

# Aufgabe 2

## Leere Einträge beim verändern löschen

Jedes mal, wenn man den Text eines Todos ändert, wird der `input` Event aufgerufen. Fügen Sie diesem Event ein neues Verhalten hinzu:

Wenn noch Text im Inputfeld vorhanden ist, so soll er wie anhin einfach verändert werden.

Wenn der Text des Inputfeldes aber leer ist, so soll der Eintrag neu gelöscht werden.

# Aufgabe 3

## Tasks abschliessen

Die Todo-Liste kann schon Einträge hinzufügen, bearbeiten und löschen. Die eine wichtige Funktion der Liste fehlt noch: Die Möglichkeit, Todos abzuschliessen.

Im HTML finden Sie bereits das Häkchen (<button> mit der Klasse `todo-done`). Wenn das Häkchen geklickt wird, soll:

1. der Häkchen-Button die Klasse «btn-success» erhalten.
2. im `tasks` Array die `done` Eigenschaft auf «true» gesetzt werden.

Beim zweiten Klick wird der Task wieder als «nicht abgeschlossen» markiert.



# Bonus Aufgabe 4

Zur Zeit haben wir mehrere Aktionen, um mit dem HTML zu arbeiten:

- Eintrag ins HTML hinzufügen
- Eintrag verändern (HTML wird nicht verändert)
- Eintrag aus dem HTML löschen
- Häkchen im HTML verändern

Statt all diese Änderungen einzeln zu machen, könnte man die Todo-Liste im HTML jeweils ganz entfernen und aus den Daten des `todos` Arrays die Liste noch einmal neu ins HTML schreiben.

So müssen nur noch die Daten im Array angepasst werden, wir brauchen uns dann keine Gedanken um die Art der Änderung im HTML machen.

# Bis zum 25. April

## Formularinteraktion II Videos

### Codecademy «JavaScript»

#### 4. 'While' Loops in JavaScript