

# HTML Fortgeschrittene Grundlagen

## Inhalt

- [Quick-Recap](#)
- [Formulare und Eingabefelder](#)

## Setup

Für Übungen nutzen wir die folgende **CodeSandbox** als Startpunkt:

<https://codesandbox.io/s/82lki>

Die Übungen bauen immer aufeinander auf. Aber keine Angst! Für den Fall, dass bei einer Übung etwas nicht klappt, gibts bei jeder Übung einen Link zur CodeSandbox mit dem aktuellen Stand.

## Quick Recap

- [Was ist HTML?](#)
- [Aufbau einer HTML-Datei](#)
- [Der Browser & Standards](#)
- [IDs & Klassen](#)
- [Block- & Inline-Elemente](#)
- [Neue Elemente](#)

## Formulare und Eingabefelder

Formulare werden genutzt, um Daten oder auch Dateien zu übermitteln.

Die Datenverarbeitung passiert anschliessend auf dem Server mit einer serverseitigen Programmiersprache wie PHP, Python, JavaScript etc.

### Aufbau

- Das `<form>` -Element ist der Container in dem die Eingabefelder platziert werden können
- Die `action` definiert, an welche URL das Formular verschickt wird
- `method` definiert die verwendete Methode:
  - `GET` (verwendbar für "nicht sichere" Daten)
  - `POST` (immer verwendbar)
- Mit einem `<button>` von `type="submit"` kann das Formular abgeschickt und die Daten anschliessend übermittelt werden

### Beispiel

```
<form action="./save-data.php" method="POST">
  <label for="vorname">Vorname *</label>
  <input type="text" name="vorname" id="vorname" value="">

  <button type="submit">Senden</button>
</form>
```

Demo 🤖

- [Formular mit GET](#)
- [Formular mit POST](#)

## Label

- Das `<label>` wird genutzt, um ein Eingabefeld zu benennen.
- Die Verknüpfung von einem Label und einem Eingabefeld geschieht über das `for`-Attribut. Dieses muss den gleichen Wert wie das `id`-Attribut des Eingabefeldes besitzen
- Eine Verknüpfung kann auch erzielt werden, indem das Eingabefeld innerhalb des `<label>`-Elementes platziert wird. Somit ist das `for`-Attribut obsolet
- Beim Klick auf das Label wird das Eingabefeld fokussiert

## Beispiel

```
<label for="vorname">Vorname</label>
<input type="text" name="vorname" id="vorname" value="">

<!-- Ohne for Attribut -->
<label>
  Vorname
  <input type="text" name="vorname" value="">
</label>
```

## Input

Das `<input>`-Element ist das meistgenutzte Eingabefeld.

Spezielle Attribute für Formularfelder:

- `name` : Unter dem `name` wird das Formular an den Server übermittelt
- `type` : Definiert, um was für ein Inputfeld es sich handelt
  - `text` : Default type und passend für Texteingaben
  - `number` : Ermöglicht nur Zahleneingaben
  - `email` : Validiert auf eine E-Mail-Adresse
  - `date` : Zeigt einen Datepicker an und formatiert die Value in einem bestimmten Format
  - `radio` :
  - `checkbox` :
  - [etc ...](#)
- Per `value`-Attribut kann das Inputfeld vorabgefüllt werden
- Per `placeholder`-Attribut kann ein temporärer Wert angezeigt werden, solange das Inputfeld *leer* ist
- Mit dem `required`-Attribut, kann ein Feld als Pflichtfeld gekennzeichnet werden. Das Formular kann anschliessend nicht mehr verschickt werden, solange dieses Feld nicht ausgefüllt ist

## Beispiel

```
<input type="text" name="vorname" value="Vorausgefüllte Daten">

<input type="text" name="vorname" value="" required placeholder="Angezeigt wenn leer">
```

```
<input type="text" name="namea" value="">
<input type="email" name="nameb" value="">
<input type="number" name="namec" value="">
<input type="date" name="named" value="">
<input type="radio" name="namee" value="">
<input type="checkbox" name="namef" value="">
<!-- usw. -->
```

## Checkbox

- `type="checkbox"` ermöglicht Checkboxes
- Die Value einer Checkbox wird nur übermittelt, wenn die Checkbox `checked` ist
- Kann per `checked` -Attribut vorausgefüllt werden

### Beispiele

```
<input type="checkbox" name="foo" value="bar" id="example">
<label for="example">Checkbox</label>
```

## Radio

- `type="radio"` ermöglicht Radiofelder
- Ermöglicht, dass von mehreren Optionen nur eine ausgewählt werden kann
- Nur die Value des Radiofeldes, welches beim Senden ausgewählt ist, wird an den Server übermittelt
- Radiofelder, die eine Gruppe bilden, müssen das gleiche `name` -Attribut besitzen

### Beispiele

```
<input type="radio" name="foo" value="radio1" id="radio1">
<label for="radio1">Radiofeld 1</label>

<input type="radio" name="foo" value="radio2" id="radio2">
<label for="radio2">Radiofeld 2</label>
```

## Select

- Ein Select-Element enthält eine oder mehrere Optionen
- Eine Option kann mit dem `selected` -Attribut vorausgewählt werden, ansonsten ist die erste Option per default gewählt

### Beispiele

```
<select name="sex">
  <option value="">Bitte wählen</option>
  <option value="woman" selected>Frau</option>
  <option value="man">Mann</option>
</select>
```

## Textarea

Textareas eignen sich für grössere Fliesstexte, wie eine Beschreibung oder eine Mitteilung bei einem Kontaktformular.

- Höhe und Breite kann entweder mit `cols="x"` und `rows="y"`, oder per CSS gesetzt werden
- Textareas sind keine **Void-Elements** und der Inhalt wird zwischen Anfangs- und Schluss-Tag geschrieben

### Beispiele

```
<textarea name="name" id="my-textarea" cols="30" rows="10">Vorausgefüllter  
Inhalt</textarea>
```

### Button

- `type` definiert die Funktion des Buttons
- Vorhandene `type`-Values:
  - `submit` : Versickt das Formular
  - `reset` : Setzt alle Eingabefelder innerhalb eines Formulars zurück
  - `button` : Ist generisch, und hat an sich keine spezielle Funktion (wird oft in Verbindung mit JavaScript genutzt)
- Default `type` innerhalb eines `<form>`-Elements ist `submit`
- Default `type` ausserhalb eines `<form>`-Elements ist `button`
- Es ist **good practice** immer einen `type` anzugeben, auch wenn dieser implizit ist

### Beispiele

```
<button type="submit">Senden</button>  
<button type="reset">Zurücksetzen</button>  
<button type="button">Trigger etwas mit JavaScript</button>
```

### Hilfreiches

Mehr zu Formularen/Eingabefeldern und deren Funktionen unter:

- [Web forms — Working with user data](#)
- [HTML Forms — w3schools](#)

### Practice

Öffne diese [CodeSandbox](#) als Startpunkt.

- ☐ Fügt das JavaScript von der Startsandbox bei euch ein (falls eine eigene Codesandbox verwendet wird).  
Wer das JavaScript selber schreiben will, kann einen EventListener auf das submit-Event setzen und der handler sollte alle values in einem alert anzeigen.
- ☐ Erstelle ein Formular mit vier Feldern
  - ☐ Ein Titelfeld. Vergib dem Feld die `id="title"`
  - ☐ Eine Textarea für die Beschreibung. Vergib dem Feld die `id="description"`
  - ☐ Ein Selectfeld mit 5 Optionen, um die Wichtigkeit zu setzen (1-5). Vergib dem Feld die `id="importance"`

- ☐ Ein Datumsfeld, damit wir ein Duedate setzen können. Vergib dem Feld die `id="duedate"`
- ☐ Vergib dem Formular die `id="form"`
- ☐ Füge noch einen Submitbutton hinzu und schaue was passiert, wenn Du das Formular abschickst

Zeit: ~ 15 min

**Solution:** <https://codesandbox.io/s/bnydy>