

Formulare in Bootstrap 4

Inhalt:

- 1) einfaches Formular mit E-Mail und Passwort (login)
- 2) Formular Elemente: form-group, form-control
- 3) Hilfe Text **class="form-text"**
- 4) HÜ: Formular mit Drop-Down-Liste
- 5) Formular mit „File-Upload“
- 6) Explizit unterstützte Formularelemente
- 7) Einzeilige Formulare – form-inline
- 8) Info: Label benutzen
- 9) Zusatztext vor oder nach dem <input>-Feld mit der Klasse „input-group“
- 10) Horizontale Formulare mit <row> oder „form-horizontal“
- 11) Checkboxen und Radio-Buttons inkl. einzeilig mit **form-check-inline**
- 12) Schaltflächen in Formularen

The screenshot shows a contact form on the BILLA website. The form is titled 'Kontaktformular' and is part of a page with the header 'BILLA' and navigation links like 'Vorteils-Club', 'Aktionen', 'Sortiment', 'BILLA Marken', and 'Nachhaltigkeit'. The form includes a dropdown menu for 'Betreff' (with the text 'Bitte auswählen'), a row of 'Anrede*' and 'Titel' dropdowns, a row of 'Vorname*' and 'Nachname*' text inputs, a row of 'Straße', 'PLZ', and 'Ort' text inputs, a row of 'E-Mail Adresse*', 'Vorwahl' (dropdown), and 'Rufnummer' (text input), and finally an 'Alter' text input.

Formulare sind wichtige Kontakt-Möglichkeiten für Besucher.

Die 2 wichtigsten Formular-Elemente:

Formularelemente erhalten automatisch die richtige Formatierung. Mit **.form-group** werden Gruppen aus Label und Eingabefeld gebildet, die sich je nach verfügbarer Breite automatisch nebeneinander oder übereinander anordnen.

Als Container für Steuerelemente wird die Klasse **.form-control** benutzt. Elemente, die eine steuerbare horizontale Ausdehnung haben, also <input>, <textarea> und <select> werden auf eine Breite von 100% des übergeordneten Containers gesetzt

Typisches Bootstrap-Formular-Element:

```
<div class="form-group">
  <label for="strasse">Straße</label>
  <input type="text" class="form-control" id="strasse">
</div>
```

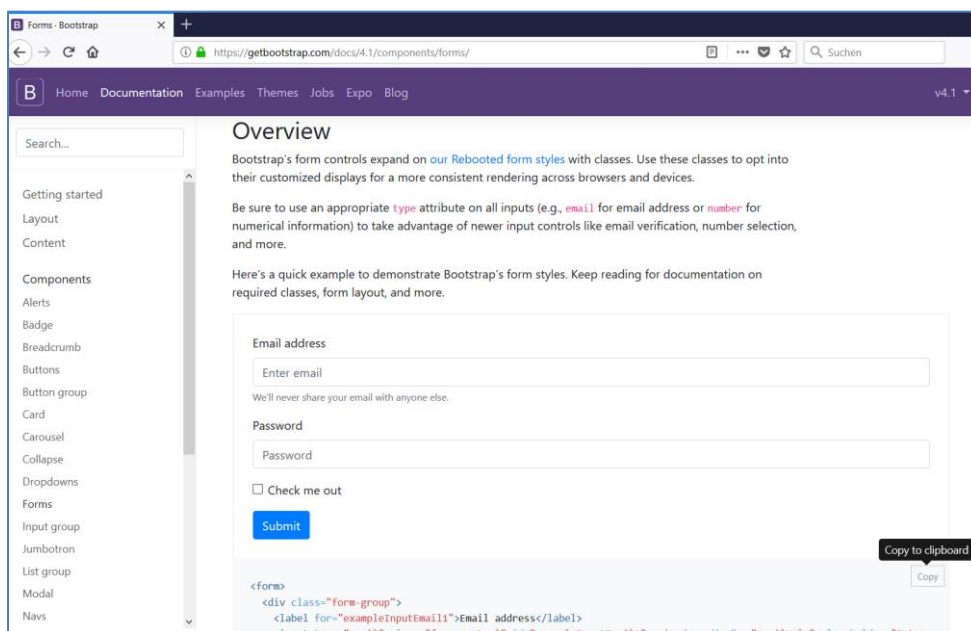
```
<div class="form-group">
  <label for="strasse">Straße</label>
  <input type="text" class="form-control" id="strasse">
</div>
```

1)einfaches Formular 2 Felder: E-Mail und Passwort (Login)

Erstelle eine mit Bootstrap verknüpfte leere „formular.html“. Ändere den Titel auf „Formular“ und die <h1> auf „Bitte füllen Sie das Formular aus.“ Erstelle direkt nach dem <body> einen <div class=“container“> der nach der Überschrift endet.

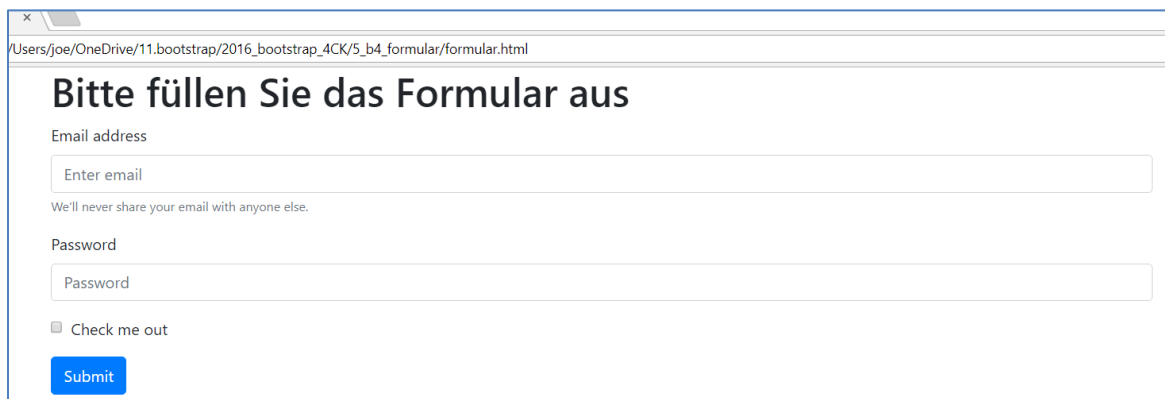
Am Beginn erstelle eine einfache <form> direkt von der www.getbootstrap.com“ Seite. Wähle dort „Documentation“ im „header“ und dann in der linken Navigation unter „Components“ den Bereich „Forms“.

Bereits ganz oben findet man die „Overview“. Deren erstes Beispiel soll mit dem Button „Copy“ rechts im Code kopiert werden.



Füge den Code in die Seite „formular.html“ ein. Nämlich nach der <h1> und vor dem Ende des Schlusses vom <container>. Dadurch klebt das Formular nicht an beiden Enden links und rechts.

Ergebnis:



Erklärung:

```
13 ▾ <body>
14 ▾   <div class="container">
15     <h1>Bitte füllen Sie das Formular aus</h1>
16 ▾   <form>
17 ▾     <div class="form-group">
18       <label for="exampleInputEmail">Email address</label>
19       <input type="email" class="form-control" id="exampleInputEmail" aria-
20         describedby="emailHelp" placeholder="Enter email">
21       <small id="emailHelp" class="form-text text-muted">Die E-Mail Adresse wird nicht
        an Dritte weitergegeben.</small>
22     </div>
```

Die einzelnen Elemente sind eingebettet in eine Klasse namens „**form-group**“.

Hier werden automatisch <label> verwendet. Das ist sehr zu empfehlen, wenn man barrierefreie Webseiten erstellen möchte. Dabei muss man nur aufpassen, dass der Name bei „for“ genau mit dem Namen übereinstimmt, der eine Zeile darunter bei <input id=""> angegeben wird.

Der „aria-describedby“-Bereich ist für die Barrierefreiheit sehr wichtig. Muss aber nicht enthalten sein.

Aufgabe:

Ändere die Angaben auf eine sinnvolle deutsche Beschreibung. Ändere auch die Namen der „id´s“ und „for“, usw. Auch bei allen anderen Elementen des kompletten Formulars.

BEACHTET: DIE KLASSEN selbst, wie z.B. class="form-control" dürfen NICHT verändert werden!!!

Ergebnis:

Bitte füllen Sie das Formular aus

E-Mail Address

E-Mail eingeben

Die E-Mail Adresse wird nicht an Dritte weitergegeben.

Passwort

Passwort eingeben

Checkbox ausprobieren

Absenden

```

15 <h1>Bitte füllen Sie das Formular aus</h1>
16 <form>
17 <div class="form-group">
18 <label for="email1">E-Mail Address</label>
19 <input type="email" class="form-control" id="email1" aria-describedby="emailHelp"
   placeholder="E-Mail eingeben">
20 <small id="emailHelp" class="form-text text-muted">Die E-Mail Adresse wird nicht
   an Dritte weitergegeben.</small>
21 </div>
22 <div class="form-group">
23 <label for="password1">Passwort</label>
24 <input type="password" class="form-control" id="password1" placeholder="Passwort
   eingeben">
25 </div>
26 <div class="form-group form-check">
27 <input type="checkbox" class="form-check-input" id="check1">
28 <label class="form-check-label" for="check1">Checkbox ausprobieren</label>
29 </div>
30 <button type="submit" class="btn btn-primary">Absenden</button>
31 </form>

```

Bei der Checkbox steht das <label> danach. Ist auch ok. Damit steht der Text auch nach der Checkbox.

2)Die Formular-Elemente

Formularelemente erhalten automatisch die richtige Formatierung. Mit ***.form-group*** werden Gruppen aus Label und Eingabefeld gebildet, die sich je nach verfügbarer Breite automatisch nebeneinander oder übereinander anordnen.

Als Container für Steuerelemente wird die Klasse ***.form-control*** benutzt. Elemente, die eine steuerbare horizontale Ausdehnung haben, also <input>, <textarea> und <select> werden auf eine Breite von 100% des übergeordneten Containers gesetzt

3)Hilfstext in Formularen: class="form-text"

Hilfetexte für Formulare werden immer wieder benötigt. Sie sind zum einen für die Anzeige, zum anderen auch für Screenreader, um barrierefreie Seiten zu unterstützen. Sie arbeiten perfekt mit den „aria-describedby-Attributen“ zusammen.

Meist wird sie etwas kleiner geschrieben, deswegen auch in <small>:

```

<div class="form-group">
  <label for="email1">E-Mail Address</label>
  <input type="email" class="form-control" id="email1" aria-
  describedby="emailHelp" placeholder="E-Mail eingeben">
  <small id="emailHelp" class="form-text text-muted">Die E-Mail Adresse wird nicht an
  Dritte weitergegeben.</small>
</div>

```

```

17 ▾ <div class="form-group">
18   <label for="email1">E-Mail Address</label>
19   <input type="email" class="form-control" id="email1" aria-describedby="emailHelp"
    placeholder="E-Mail eingeben">
20   <small id="emailHelp" class="form-text text-muted">Die E-Mail Adresse wird nicht
    an Dritte weitergegeben.</small>
21 </div>

```

Übung:

Ändere die Passwort-Passage in eine mit Hilfstext bei der Passworteingabe.

Ziel:

Passwort

Passwort eingeben

Das Passwort muss mindestens 8 Stellen haben.

Füge daher einiges hinzu. Ergebnis:

```

22 ▾ <div class="form-group">
23   <label for="password1">Passwort</label>
24   <input type="password" class="form-control" id="password1" placeholder="Passwort
    eingeben" aria-describedby="passwordHelpInline">
25   <small id="passwordHelpInline" class="form-text text-muted">
26   Das Passwort muss mindestens 8 Stellen haben.
27   </small>
28 </div>

```

4)HÜ: Formular mit Drop-Down-Liste (select menu) und „form-row“

Erstelle folgendes Formular aus einer leeren „index.html“, die bereits die Links für Bootstrap 4 enthält.

Mit der `<class="form-row">` und den anschließenden `<col>` kann man die Spaltenmöglichkeit (Grid) von Bootstrap perfekt benutzen.

Ziel-Lösung:

Bitte geben Sie Ihre Daten ein

Vorname: Nachname:

E-Mail: Passwort:

Die E-Mail Adresse wird nicht an Dritte weitergegeben.

Strasse und Hausnummer:

Ort: Bundesland: PLZ:

AGB akzeptiert


Info: select menu:

```
49 <label for="bundesland">Bundesland</label>
50 ▼ <select id="bundesland" class="form-control">
51 <option selected>NÖ</option>
52 <option>Bgl.d.</option>
53 <option>Wien</option>
54 <option>OÖ</option>
55 </select>
```

Man benötigt ein `<select>` und mehrere `<option>`. Dabei kann z.B. das erste bereits ausgewählt sein (selected), damit etwas in dem Feld steht. Alle Möglichkeiten sind dann in `<option>` angeführt.

Für spätere Übergabe an eine Datenbank wäre es sinnvoll, hier bereits Werte der einzelnen Optionen anzugeben:

```
<option value="1" selected>NÖ</option>
<option value="2">Bgl.d.</option>
<option value="3">Wien</option>
<option value="4">OÖ</option>
```

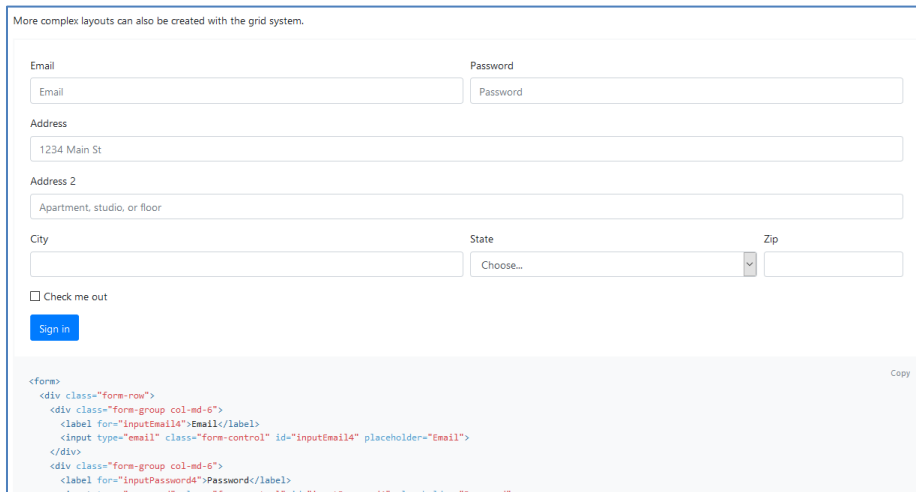
Vielleicht besser wäre eine andere Klasse: custom-select: 

```
50 ▼ <select id="bundesland" class="custom-select">
51 <option selected>NÖ</option>
52 <option>Bgl.d.</option>
53 <option>Wien</option>
54 <option>OÖ</option>
55 </select>
```

Info 2: Die ersten beiden Zeilen sind mittels `<row>` und `<col>` erstellt:

```
18 ▾ <div class="form-row">
19 ▾ <div class="form-group col-md-6">
```

Ausgangsgerüst war von www.getbootstrap.com unter „Documentation, Components und Forms“ zu finden, nämlich im Bereich „Form row“:



More complex layouts can also be created with the grid system.

Email Password

Address

Address 2

City State Zip

Check me out

Sign in

```
<form>
  <div class="form-row">
    <div class="form-group col-md-6">
      <label for="inputEmail4">Email</label>
      <input type="email" class="form-control" id="inputEmail4" placeholder="Email">
    </div>
    <div class="form-group col-md-6">
      <label for="inputPassword4">Password</label>
      <input type="password" class="form-control" id="inputPassword4" placeholder="Password">
    </div>
  </div>
  <div class="form-group">
    <label for="inputAddress">Address</label>
    <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St">
  </div>
  <div class="form-group">
    <label for="inputAddress2">Address 2</label>
    <input type="text" class="form-control" id="inputAddress2" placeholder="Apartment, studio, or floor">
  </div>
  <div class="form-group">
    <label for="inputCity">City</label>
    <input type="text" class="form-control" id="inputCity">
  </div>
  <div class="form-group">
    <label for="inputState">State</label>
    <input type="text" class="form-control" id="inputState" value="Choose...">
  </div>
  <div class="form-group">
    <label for="inputZip">Zip</label>
    <input type="text" class="form-control" id="inputZip">
  </div>
  <div class="form-group">
    <input type="checkbox" id="checkboxCheck" value="Check me out"/> Check me out
  </div>
  <div class="form-group">
    <input type="button" value="Sign in"/>
  </div>
</form>
```

Kopiere dieses und ändere es entsprechend.

Die Zeile mit „Email“ und „Password“ wird kopiert und darunter eingefügt, um für „Vorname und Nachname“ vorhanden zu sein.

5) Beispiel für ein Formular mit „File-Upload“

Vor dem Absenden-Button soll eine Datei auswählbar und eine Upload-Funktion ermöglicht werden. Da kein PHP-Skript hier vorhanden ist, ist es natürlich nur eine optische Variante.

Füge folgenden Code ein:

```
<div class="form-group">
  <label for="exampleFormControlFile1">Example file input</label>
  <input type="file" class="form-control-file" id="exampleFormControlFile1">
</div>
```

Ändere bestimmte Bereiche, damit es auf unser Formular und die deutsche Sprache angepasst wird:
Ergebnis:

```
29 </div>
30 ▾ <div class="form-group">
31 <label for="file1">Upload-Möglichkeit:</label>
32 <input type="file" class="form-control-file" id="file1">
33 </div>
34
```

Upload-Möglichkeit:

Keine ausgewählt

Info: type="file" ermöglicht das Durchsuchen im Explorer, wenn man auf „Datei auswählen“ klickt.

6) Explizit unterstützte Formularelemente

Formularelemente der Gruppe <input> benötigen das type-Attribut, um korrekt angezeigt zu werden. Unterstützt wird praktisch der gesamte Vorrat an HTML5-Typen:

- text
- password
- datetime
- datetime-local
- date
- month
- time
- week
- number
- email
- url
- search
- tel
- color

Beispiel: „text“:

```
<input type="text" class="form-control" placeholder="Text eingeben">
```

Richtige Tastatur

Dank den Formular-Input-Feldern in HTML5 wird bei Smartphones automatisch die richtige Tastatur ausgewählt. Es wäre sehr unangenehm, wenn man bei der Eingabe einer E-Mail Adresse erst umständlich nach dem @-Zeichen suchen müsste. Besser ist es, wenn die richtige Tastatur vorgeschlagen wird.

Beispiel: <type="tel"> für Telefon

```
<input type="tel" name="telefonnummer" id="telnummer" />
```



7) Einzeilige Formulare

Einzeilige Formulare stehen ab einer Breite von 576 Pixeln zur Verfügung. Darunter werden sie nie in einer Zeile dargestellt, weil zu wenig Platz ist.

Mit "einzeilig" ist gemeint, dass der Feldname (Label) und das Eingabefeld **nebeneinander** stehen können. Sie werden eingeleitet mit

.form-inline

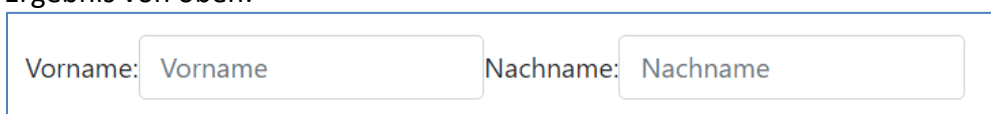
Das umschließende <form>-Tag ist optional. Es kann von der Logik der Seite oder dem Browserverhalten her benötigt werden – Bootstrap reagiert darauf jedoch nicht. Die Breite der Elemente mit variabler Ausdehnung ist "auto". Die Breite wird also innerhalb des umschließenden Containers optimiert. Es kann notwendig sein, die Breite individuell zu steuern.

Beispiel:

Erzeuge unter dem obigen Formular einen kurzen Kommentar „<!--Zweites Formular-->“ und mit einigen
 etwas Platz. Dann eine Überschrift <h1>Einzeiliges Formular mit #form-inline</h1>. Darunter dann das neue Formular:

```
34 </form>
35 <br><br>
36 <!--zweite Form für einzeilige Darstellung-->
37 <form class="form-inline">
38 <label for="name2">Vorname: </label>
39 <input type="text" class="form-control" id="name2" placeholder="Vorname">
40 <label for="username2">Nachname: </label>
41 <input type="text" class="form-control" id="username2" placeholder="Nachname">
42 </form>
43
44 </div> <!--ende container-->
```

Ohne der Klasse „form-group“ würde alles sehr an einander kleben:
Ergebnis von oben:



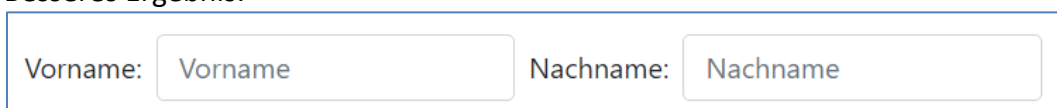
Daher Lösung:

Einfügen von Abständen durch „margin“ right und left:

```
<label for="name2">Vorname: </label>
<input type="text" class="form-control ml-2 mr-2" id="name2" placeholder="Vorname">
<label for="username2">Nachname: </label>
<input type="text" class="form-control ml-2 mr-2" id="username2"
placeholder="Nachname">
```



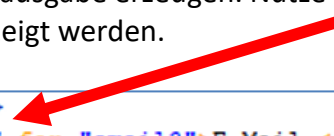
Besseres Ergebnis:



8)Info: Label benutzen

Man sollte auch bei einzeiligen Formularen immer Label einsetzen. Screenreader können sonst keine sinnvolle Sprachausgabe erzeugen. Nutze .sr-only, damit die Label auf normalen Ausgabegeräten nicht angezeigt werden.

```
<div class="form-group">  
  <label class="sr-only" for="email3">E-Mail </label>  
  <input type="email" class="form-control" id="email3"  
</div>
```



Das Label wird nur Screen-Readern angezeigt und bleibt ansonsten unsichtbar.

Wenn eine barrierefreie Ausgabe erforderlich ist, sollten darüber hinaus immer die vom Standard geforderten Attribute aria-label und aria-labelledby benutzt werden.

Tipp: <label> verwenden

Beispiel:

```
<form>  
  <label for="name">Name:</label>  
  <input name="name" id="name">  
  <label for="notiz">Notiz:</label>  
  <input name="notiz" id="notiz">  
</form>
```

<label> markiert, dass der eingeschlossene Text die Beschriftung des Eingabefeldes ist. **Dazu muss das „for“-Attribut des <label>s mit der „id“ des Eingabefeldes übereinstimmen.** Das <label> kann „sein“ Eingabefeld auch umschließen, dann entfällt das „for“-Attribut, aber sonst ändert sich nichts.

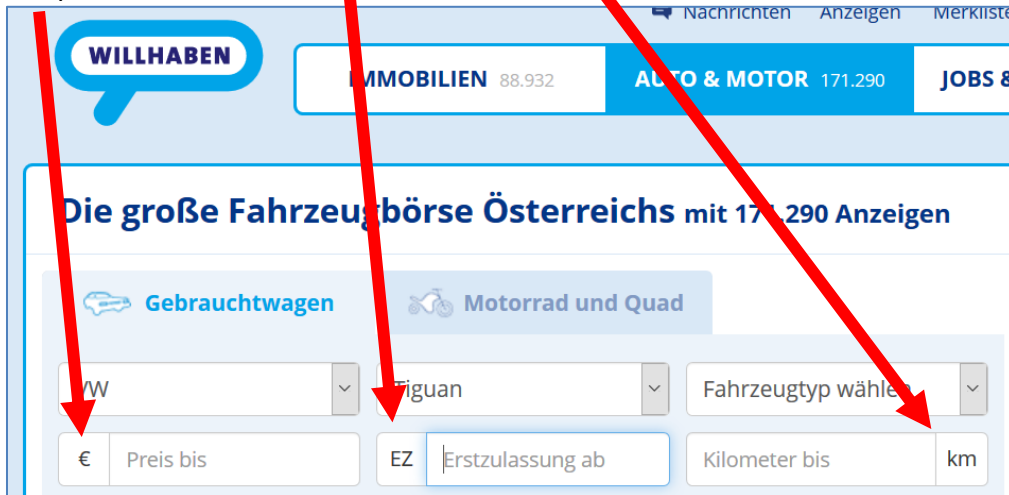
Man könnte <label> einfach weglassen und die Beschriftung des Eingabefeldes als Text ohne Tag schreiben, es sähe vollkommen gleich aus. Ein guter Grund für <label> ist aber die Barrierefreiheit: Lesegeräte können so klarmachen, welcher Text zu welchem Eingabefeld gehört. Aber auch im herkömmlichen Browser gibt es einen Vorteil: ein Klick auf das Label aktiviert das Eingabefeld, genauso als hätte man das Feld angeklickt. Das ist z.B. ein Vorteil auf dem Handy.

9) Klasse „input-group“

Damit können “Extras” vor und hinter dem Element eingefügt werden, nämlich mit

.input-group

Beispiel: willhaben.at



Beispiel:

- Erzeuge im ersten Formular, noch vor dem Absenden-Button einen kurzen Kommentar „<!--zusatz verwenden-->“ und mit einigen
 etwas Platz.

Ergebnis:

Betrag:

Euro Betrag eingeben

Absenden

prepend: voranstellen
append: dahinter stellen

Where *property* is one of:

- **m** - for classes that set `margin`
- **p** - for classes that set `padding`

Where *sides* is one of:

- **t** - for classes that set `margin-top` or `padding-top`
- **b** - for classes that set `margin-bottom` or `padding-bottom`
- **l** - for classes that set `margin-left` or `padding-left`
- **r** - for classes that set `margin-right` or `padding-right`
- **x** - for classes that set both `*-left` and `*-right`
- **y** - for classes that set both `*-top` and `*-bottom`
- blank - for classes that set a `margin` or `padding` on all 4 sides of the element

```

32 </div>
33 <!--zusatz verwenden-->
34 <br>
35 <div class="form-group">
36 <label for="zusatz">Betrag:</label>
37 <div class="input-group mb-2">
38 <div class="input-group-prepend">
39 <div class="input-group-text">Euro</div>
40 <input type="text" class="form-control"
41 id="zusatz" placeholder="Betrag eingeben">
42 </div>
43 </div>
44 <button type="submit" class="btn btn-
45 primary">Absenden</button>

```

```

<!--zusatz verwenden-->
<br>
<div class="form-group">
<label for="zusatz">Betrag:</label>
<div class="input-group mb-2">
<div class="input-group-prepend">
<div class="input-group-text">Euro</div>
<input type="text" class="form-control" id="zusatz" placeholder="Betrag eingeben">
</div>
</div>
</div>

```

Tipp: bei append

Die <input> muss davor stehen und man muss append verwenden:

```

<div class="form-group">
<label for="zusatz">Getränke:</label>
<div class="input-group mb-2">
<input type="text" class="form-control" id="zusatz"
placeholder="Menge in Liter eingeben">
<div class="input-group-append">
<div class="input-group-text">Liter</div>
</div>
</div>
</div>

```

Klasse „input-group-addon“

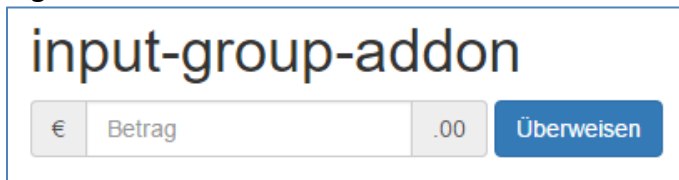
Damit können “Extras” vor und hinter dem Element eingefügt werden, nämlich mit

.input-group-addon

Beispiel:

Erzeuge unter dem obigen Formular einen kurzen Kommentar „<!--addon--> und mit einigen
 etwas Platz. Dann eine Überschrift <h1>input-group-addon</h1>. Darunter dann das neue Formular:

Ergebnis:



The screenshot shows a web form with the title "input-group-addon". Below the title is a form with a text input field. The input field has a Euro symbol (€) on the left and ".00" on the right. To the right of the input field is a blue button labeled "Überweisen".

```
65 <!--addon-->
66 <br><br>
67 <h1>input-group-addon</h1>
68 <form class="form-inline">
69   <div class="form-group">
70     <label class="sr-only" for="amount">Betrag (in Euro) </label>
71     <div class="input-group">
72       <div class="input-group-addon">&euro;</div>
73       <input type="text" class="form-control" id="amount" placeholder="Betrag">
74       <div class="input-group-addon">.00</div>
75     </div>
76   </div>
77   <button type="submit" class="btn btn-primary">Überweisen</button>
78 </form>
```

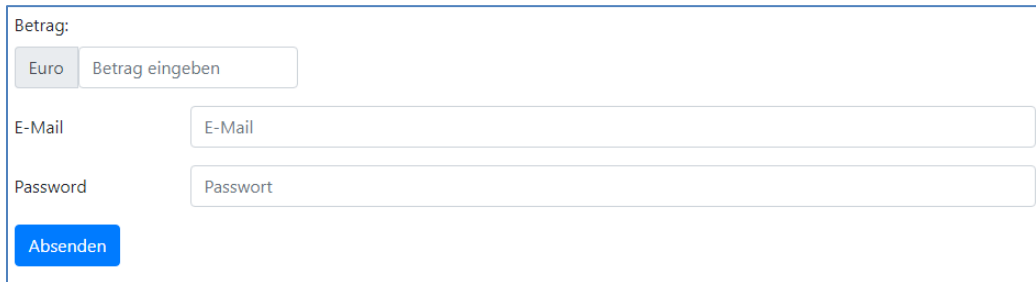
Das Label (Zeile 70) wird nur Screenreadern angezeigt und bleibt ansonsten unsichtbar.

10) Horizontale Formulare mit <row>

Manchmal soll bzw. muss der Text in einer horizontalen Form neben dem Input-Feld stehen und nicht darüber.

Um Label und Felder zu platzieren, wird das Raster benutzt. Man nutzt daher

```
<div class="form-group row">
```



The screenshot shows a form with the following elements:

- A label "Betrag:" followed by a "Euro" button and an input field with the placeholder "Betrag eingeben".
- A label "E-Mail" followed by an input field with the placeholder "E-Mail".
- A label "Password" followed by an input field with the placeholder "Passwort".
- A blue "Absenden" button at the bottom left.

```
44     <!--horizontale Form mit text neben feld-->
45 ▼ <div class="form-group row">
46     <label for="hori-mail" class="col-sm-2 col-form-label">E-
      Mail</label>
47 ▼     <div class="col-sm-10">
48         <input type="email" class="form-control" id="hori-mail"
           placeholder="E-Mail">
49     </div>
50 </div>
51 ▼ <div class="form-group row">
52     <label for="hori-kennwort" class="col-sm-2 col-form-
      label">Password</label>
53 ▼     <div class="col-sm-10">
54         <input type="password" class="form-control" id="hori-
           kennwort" placeholder="Passwort">
55     </div>
56 </div>
```

```
<!--horizontale Form mit text neben feld-->
```

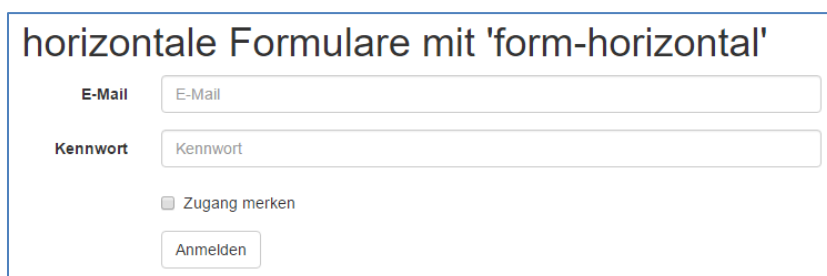
```
<div class="form-group row">
  <label for="hori-mail" class="col-sm-2 col-form-label">E-Mail</label>
  <div class="col-sm-10">
    <input type="email" class="form-control" id="hori-mail" placeholder="E-Mail">
  </div>
</div>
<div class="form-group row">
  <label for="hori-kennwort" class="col-sm-2 col-form-label">Password</label>
  <div class="col-sm-10">
    <input type="password" class="form-control" id="hori-kennwort"
placeholder="Passwort">
  </div> </div>
```

Horizontale Formulare – mit „form-horizontal“

Um Label und Felder zu platzieren, wird das Raster benutzt. Damit die Umbrüche unterdrückt werden, die normalerweise einen kompletten Block von Label und Feld von der nächsten Gruppe trennen, wird

.form-horizontal

benutzt. Die Angabe der Klasse erfolgt in einem umschließende <form>-Element ist. Die Klasse .row innerhalb dieses Containers ist nicht notwendig. Die ohnehin erforderliche Gruppierung mit .form-groups führt dazu, dass sich die Gruppe wie eine Reihe verhält. Man kann .row aber einsetzen, weil sich einige Editoren sonst darüber beschwerten – sichtbare Effekte werden dennoch nicht erzielt.



horizontale Formulare mit 'form-horizontal'

E-Mail

Kennwort

Zugang merken

Die Label stehen links neben den Elementen. Bei geringer Breite fällt dieses Layout (im Bild ist die Breite > 768 Pixel) wieder auf das vorher gezeigte Schema mit darüber platzierten Labeln zurück.

```
79 <!--horizontale Formulare-->
80 <br><br>
81 <h1>horizontale Formulare mit 'form-horizontal'</h1>
82 <form class="form-horizontal">
83 <div class="form-group">
84 <label for="email4" class="col-sm-2 control-label">E-Mail</label>
85 <div class="col-sm-10">
86 <input type="email" class="form-control" id="email4" placeholder="E-Mail">
87 </div>
88 </div>
89 <div class="form-group">
90 <label for="password4" class="col-sm-2 control-label">Kennwort</label>
91 <div class="col-sm-10">
92 <input type="password" class="form-control" id="password4" placeholder="Kennwort">
93 </div>
94 </div>
95 <div class="form-group">
96 <div class="col-sm-offset-2 col-sm-10">
97 <div class="checkbox">
98 <label>
```

11)Checkbox, Radio-Button – form-check

Kontrollkästchen (checkbox) und Auswahlfelder (radio button) funktionieren wie bei Standard-HTML.

Es wird die Klasse „form-check“ verwendet.

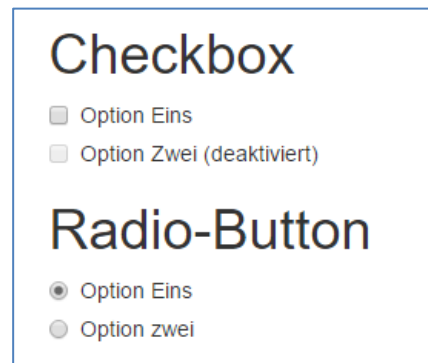
Diese wird sowohl für Checkboxes als auch für Radio-Buttons verwendet.

Checkboxes:

- Es können eine oder mehrere Optionen gewählt werden

Radio-Buttons:

- Nur eine Option ist wählbar



11a)Übung: Checkboxes

Rote Farbe
 blaue Farbe

```
57 <!--checkboxen-->
58 <div class="form-check">
59   <input class="form-check-input" type="checkbox" value="1"
      id="rot">
60   <label class="form-check-label" for="rot">
61     Rote Farbe
62   </label>
63 </div>
64 <div class="form-check">
65   <input class="form-check-input" type="checkbox" value="2"
      id="blau">
66   <label class="form-check-label" for="blau">
67     blaue Farbe
68   </label>
69 </div>
70 <br>
```

```
<!--checkboxen-->
<div class="form-check">
  <input class="form-check-input" type="checkbox" value="1" id="rot">
  <label class="form-check-label" for="rot">
    Rote Farbe
  </label>
</div>
```

```

<div class="form-check">
  <input class="form-check-input" type="checkbox" value="2" id="blau">
  <label class="form-check-label" for="blau">
    blaue Farbe
  </label>
</div>

```

11b)Übung: Radio Buttons

- unter Euro 2000,00
 - zwischen 2000,00 und 3000,00
 - über Euro 3000,00

```

71 <!--radio buttons-->
72 <div class="form-check">
73   <input class="form-check-input" type="radio" name="geld"
74     id="radio1" value="option1" checked>
75   <label class="form-check-label" for="radio1">
76     unter Euro 2000,00
77   </label>
78 </div>
79 <div class="form-check">
80   <input class="form-check-input" type="radio" name="geld"
81     id="radio2" value="option2">
82   <label class="form-check-label" for="radio2">
83     zwischen 2000,00 und 3000,00
84   </label>
85 </div>
86 <div class="form-check">
87   <input class="form-check-input" type="radio" name="geld"
88     id="radio3" value="option3">
89   <label class="form-check-label" for="radio3">
90     über Euro 3000,00
91   </label>
92 </div> <!--ende radio buttons-->

```

```

<!--radio buttons-->
<div class="form-check">
  <input class="form-check-input" type="radio" name="geld" id="radio1" value="option1" checked>
  <label class="form-check-label" for="radio1">
    unter Euro 2000,00
  </label>
</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="geld" id="radio2" value="option2">
  <label class="form-check-label" for="radio2">
    zwischen 2000,00 und 3000,00
  </label>
</div>

```

```

</div>
<div class="form-check">
  <input class="form-check-input" type="radio" name="geld" id="radio3" value="option3">
  <label class="form-check-label" for="radio3">
    über Euro 3000,00
  </label>
</div> <!--ende radio buttons-->

```

Aufgabe: einzeilige Darstellung

Ändere das Formular in eine einzeilige Darstellung durch Hinzufügen von „**form-check-inline**“ zu den Klassen jeder einzelnen Option:

```

57 <!--checkboxen-->
58 <div class="form-check form-check-inline">
59   <input class="form-check-input" type="checkbox" value="1"
      id="rot">
60   <label class="form-check-label" for="rot">
61     Rote Farbe

```

Ergebnis:

12)Schaltflächen in Formularen

Technisch kann eine Schaltfläche nur ein Formular absenden, was praktisch immer eine POST-Anfrage auslöst. Wird dagegen ein Zugriff auf den Server via GET benötigt, kommt ein Hyperlink zum Einsatz. Gestalterisch stellt Bootstrap beides auf eine Stufe.

Schaltflächen lassen sich mit

- `<a>`
- `<button>` oder
- `<input>` erstellen.

Werden `<a>`-Tags als Schaltfläche benutzt, sollten diese nur der Seitennavigation dienen und nicht ihre ursprüngliche Funktion als Hyperlink auslösen, weil dies nicht das vom Benutzer erwartete Verhalten ist. Darüber hinaus sollte die Funktion durch `role="button"` unterstrichen werden. Auch dann ist es fraglich, ob dies eine gute Idee ist, weil nicht bei allen Browsern sichergestellt werden kann, dass sich die Funktionalität wie erwartet verhält.

Fazit:

Nutze im Zweifelsfall für Schaltflächen – wenn immer dies möglich ist – bevorzugt <button>.

Schaltflächen gibt es in 6 + 3 Varianten (– Farbe):



Daher:

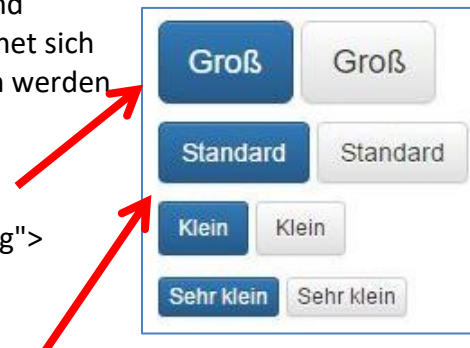
```
<button class="btn btn-primary" type="submit">Submit form</button>
```

Größe der Buttons:

Für größere oder kleinere Schaltflächen sollten die Klassen .btn-lg, .btn-sm oder .btn-xs benutzt werden. Sie ergänzen die Basisklasse btn und sind unabhängig von der Farbgebung. Die Standardgröße eignet sich bereits für die Touch-Bedienung. Mausgesteuerte Seiten werden mit kleineren Schaltflächen besser aussehen.

Teil-Beispiel für Code:

```
<p> <button type="button" class="btn btn-primary btn-lg">
  Groß </button>
<button type="button" class="btn btn-default btn-lg">
  Groß </button>
<p> <button type="button" class="btn btn-primary"> Standard</button></p>
```



Manchmal sollen Schaltflächen den Container, indem sie sind, komplett ausfüllen. Dazu dient die

Klasse .btn-block.

```
<button type="button"
  class="btn btn-primary btn-lg btn-block"> Block </button>
<button type="button"
  class="btn btn-default btn-lg btn-block"> Block </button>
```



Beachte hier, dass dies nur auf Breite wirkt. Eine passende Höhe muss separat mit `.btn-lg` usw. erreicht werde