

Internet and Web Principal

Week 12

Bootstrap 2

Content

1. Forms
2. Additional Modules
3. Components

Forms

Forms are fully supported in Bootstrap 4. Many of the components are mainly used to make the forms responsive and can be used with any screen width.

Structure of a Form

Form elements automatically receive the correct formatting. The main class for controls is `.form-control`. Elements that have controllable horizontal extensions such as `<input>`, `<textarea>`, and `<select>` are set to a width of 100% of the parent container. Using `.form-group` the labels and inputs are grouped. They arrange themselves depending on the available width either side-by-side or above one another.

Forms

Simple Form Elements

Listing 5-1 gives an example of a typical form.

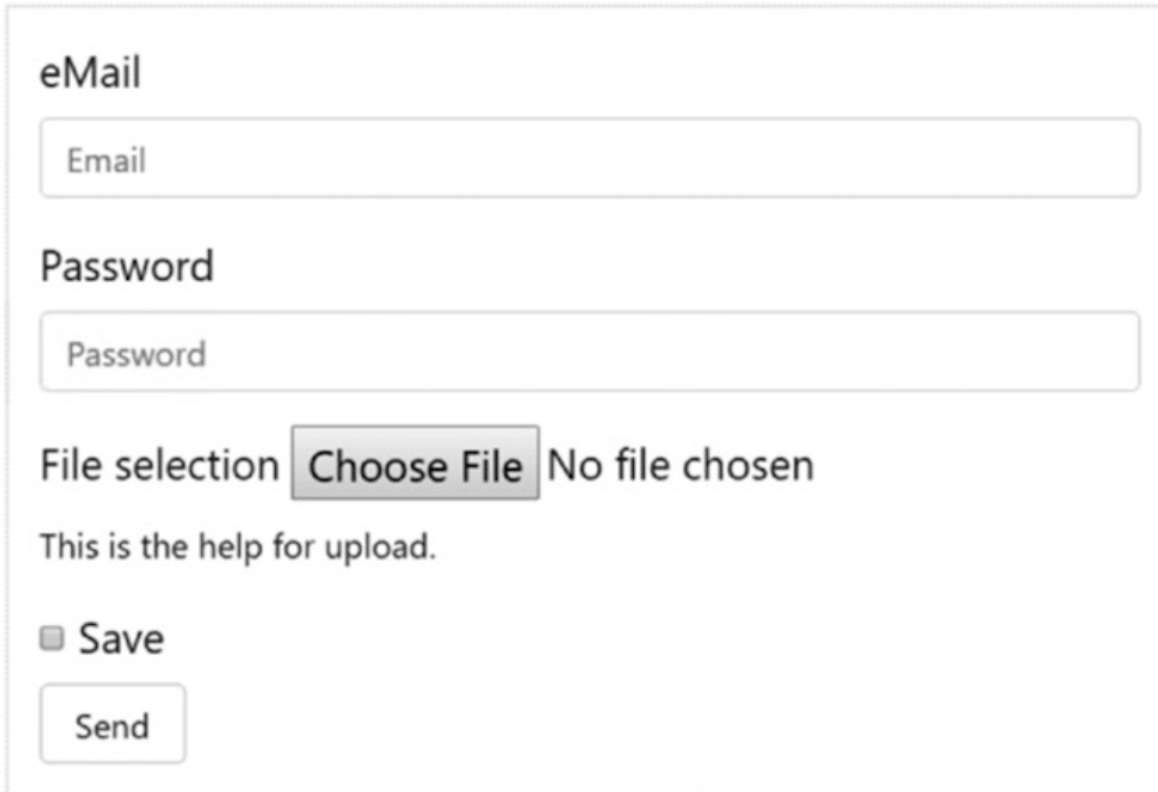
Listing 5-1. Standard structure of a form (Form_Base.html)

```

1  <form>
2    <div class="form-group">
3      <label for="txtMail">eMail</label>
4      <input type="email" class="form-control"
5          id="txtMail" placeholder="eMail">
6    </div>
7    <div class="form-group">
8      <label for="txtPassword">password</label>
9      <input type="password" class="form-control"
10         id="txtPassword" placeholder="Password">
11    </div>
12    <div class="form-group">
13      <label for="txtFile">File Selection</label>
14      <input type="file" id="txtFile">
15      <p class="form-text small">
16        Here is the help for uploading.
17      </p>
18    </div>
19    <div class="checkbox">
20      <label>
21        <input type="checkbox"> Save
22      </label>
23    </div>
24    <button type="submit" class="btn btn-secondary">
25      Send
26    </button>
27  </form>

```

Forms



eMail

Password

File selection No file chosen

This is the help for upload.

Save

The outer section is always the element group `.form-group`. The element is again given the `.form-control` class. All the other parts do not require explicit classes. Figure 5-1 shows an example of a simple form.

Figure 5-1. A simple form

Forms

Single-Line Forms

Single-line forms are available from a width of 768 px and up. “One-line” means that the field name (label), box, and other elements are able to stand side-by-side, as long as the horizontal space is sufficient. They are preceded by `.form-inline`. The enclosing `<form>` tag is optional. It may be required by the logic of the page or the browser behavior—Bootstrap does not care. The width of the elements with variable expansion is “auto.” Therefore, the width is optimized within the enclosing container. Sometimes, it may be necessary to control the width individually.

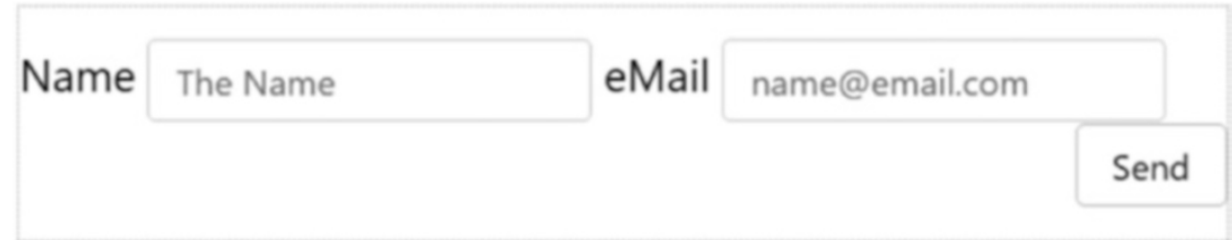
Forms

Listing 5-2. Compact form (Form_Inline.html)

```

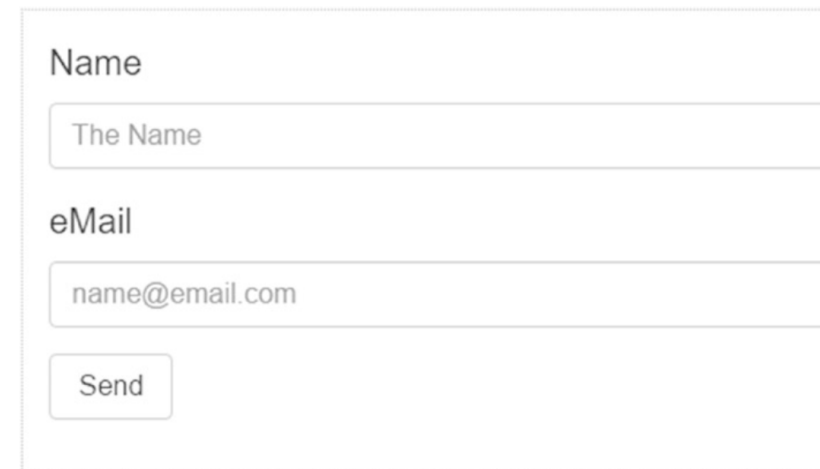
1  <form class="form-inline">
2    <div class="form-group">
3      <label for="exampleInputName2">Name</label>
4      <input type="text"
5        class="form-control"
6        id="exampleInputName2"
7        placeholder="The Name">
8    </div>
9    <div class="form-group">
10     <label for="exampleInputEmail2">Email</label>
11     <input type="email"
12       class="form-control"
13       id="exampleInputEmail2"
14       placeholder="name@email.com">
15   </div>
16   <div class="form-group">
17     <button type="submit" class="btn btn-secondary float-md-right">
18       Submit
19     </button>
20   </div>
21 </form>

```



A wide horizontal form with two input fields and a submit button. The first field is labeled "Name" and contains the text "The Name". The second field is labeled "eMail" and contains the text "name@email.com". A "Send" button is positioned to the right of the second field.

Figure 5-2. A form with a horizontal orientation (wide)



A narrow horizontal form with two input fields and a submit button. The labels "Name" and "eMail" are stacked vertically. The first field contains "The Name" and the second field contains "name@email.com". A "Send" button is positioned below the second field.

Figure 5-3. A form with a horizontal orientation (narrow)

Forms

Form Elements with Blocks

Input fields, especially those for text entry, can be supplemented for left or right text, icons, or buttons. This is particularly interesting when the values have units. Use the class `.input-group` with `.input-group-addon` to display elements before or after using `.form-control` to create. This works very well with all the `<input>`-elements, with limited success with `<select>`, but not with `<textarea>`.

Forms

Tooltips and overlapping effects also require additional effort. At least, using the optional container: “body” in JavaScript is required to avoid side effects. This parameter determines where the dynamic element in the DOM (document object model) of the page will be inserted.

```
$("#toolbarBtn1").dropdown({  
  container: 'body' });
```

The columns of the grid classes cannot mix with the input groups. Instead, the entire input group should be placed in a container if they are required, which in turn is provided with pitches.

Forms

Listing 5-4. Fields (Toolbar_Inputgroups.html)

```

1  <div class="input-group">
2    <span class="input-group-addon" id="basic-addon1">@</span>
3    <input type="text" class="form-control"
4      placeholder="User"
5      aria-describedby="basic-addon1">
6  </div>
7
8  <div class="input-group">
9    <input type="text" class="form-control"
10     placeholder="Email"
11     aria-describedby="basic-addon2">
12    <span class="input-group-addon" id="basic-addon2">
13      @example.com
14    </span>
15  </div>
16
17 <div class="input-group">
18   <span class="input-group-addon">&euro;</span>
19   <input type="text" class="form-control"
20     aria-label="Amount (in EUR)">
21   <span class="input-group-addon">.00</span>
22 </div>

```



Figure 5-5. Fields

Forms

Dealing with Checkboxes and Radio Buttons

Checkboxes and radio buttons must be placed right next to input fields. Again, groups are used to force the mapping as shown in Listing 5-6.

Listing 5-6. Fields with options (Toolbar_Inputradio.html)

```

1  <div class="row">
2    <div class="col-lg-6">
3      <div class="input-group">
4        <span class="input-group-addon">
5          <input type="checkbox" aria-label="">
6        </span>
7        <input type="text" class="form-control" aria-label="">
8      </div>
9    </div>
10   <div class="col-lg-6">
11     <div class="input-group">
12       <span class="input-group-addon">
13         <input type="radio" aria-label="">
14       </span>
15       <input type="text" class="form-control" aria-label="">
16     </div>
17   </div>
18 </div>

```

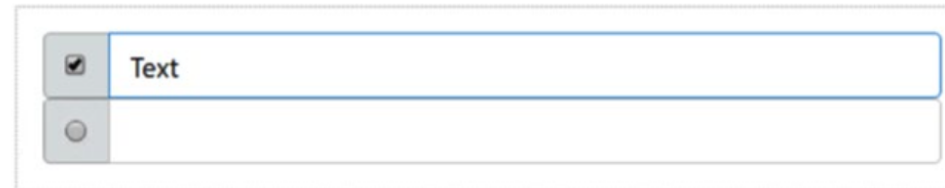


Figure 5-7. Input fields with options

Forms

Additional Buttons

Buttons in box groups are treated a little differently. They require another nesting of HTML elements based on the class `.input-group-btn`, which is shown in Listing 5-7 and Figure 5-8.

Listing 5-7. Fields with buttons (Toolbar_InputBtn.html)

```

1  <div class="row">
2    <div class="col-lg-6">
3      <div class="input-group">
4        <span class="input-group-btn">
5          <button class="btn btn-default" type="button">
6            Go!
7          </button>
8        </span>
9        <input type="text" class="form-control"
10         placeholder="Search for ...">
11      </div>
12    </div>
13    <div class="col-lg-6">
14      <div class="input-group">
15        <input type="text" class="form-control"
16         placeholder="Search for...">
17        <span class="input-group-btn">
18          <button class="btn btn-default" type="button">
19            Go!
20          </button>
21        </span>
22      </div>
23    </div>
24  </div>

```



Figure 5-8. Input fields with buttons

Forms

Horizontal Forms

To place labels and fields, a grid is used. Thus, breaks are suppressed. The complete sections of label from one group are normally separated from another using `.form-horizontal`. The particular class is carried in a container element, it is either the enclosing `<form>` element or an equivalent `<div>` is employed. The class `.row` within this container is not necessary. The required grouping `.form-groups` means that the group behaves like one element. You can use `.row` but some editors complain about it as the visible effects are not achieved.

Forms

Listing 5-11. Horizontal form (Form_Horizontal.html)

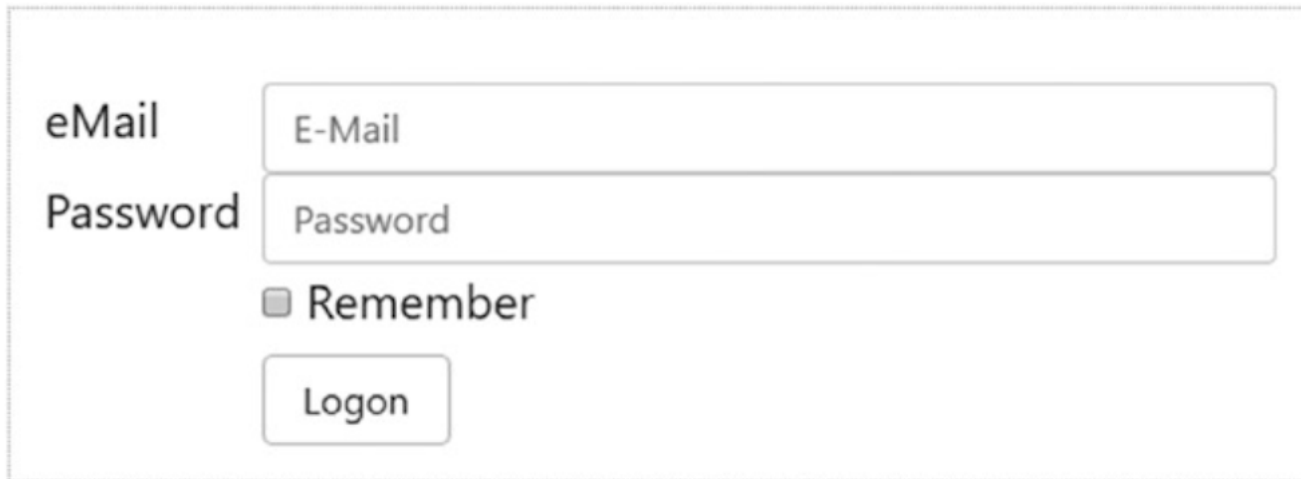
```

1  <form class="form-horizontal">
2    <div class="form-group">
3      <label for="inputEmail3" class="col-sm-2
4        form-form-control-label">E-Mail</label>
5      <div class="col-sm-10">
6        <input type="email" class="form-control"
7          id="inputEmail3" placeholder="eMail">
8      </div>
9    </div>
10   <div class="form-group">
11     <label for="inputPassword3" class="col-sm-2
12       form-form-control-label">Password</label>
13     <div class="col-sm-10">
14       <input type="password" class="form-control"
15         id="inputPassword3" placeholder="Password">
16     </div>
17   </div>
18   <div class="form-group">
19     <div class="offset-sm-2 col-sm-10">
20       <div class="checkbox">
21         <label>
22           <input type="checkbox"> Remember access
23         </label>
24       </div>
25     </div>
26   </div>
27   <div class="form-group">
28     <div class="offset-sm-2 col-sm-10">
29       <button type="submit" class="btn btn-secondary">
30         Log In
31       </button>
32     </div>
33   </div>
34 </form>

```

Forms

In Figure 5-12 the labels are to the left of the elements. When you are working with layouts with narrow widths (in the figure the width is > 768 px), you must go back to the previously shown scheme about placed labels.



The figure shows a login form with the following elements:

- An "eMail" label to the left of an input field containing the text "E-Mail".
- A "Password" label to the left of an input field containing the text "Password".
- A checkbox labeled "Remember".
- A "Logon" button.

Figure 5-12. *A form with a label next to the field*

Forms

Buttons

Buttons can be found in one form or another in each web application. Bootstrap buttons are seen as a design element that is separate from the technical framework. Technically, a “submit” appears only once in a form, which almost always triggers a POST request. If, however, you need access to the server via a GET request, a hyperlink is used. In design terms Bootstrap now provides both on the same level. Buttons can be created with `<a>`, `<button>`, or `<input>` (see Listing 5-27 and see Figure 5-27).

Forms

Listing 5-27. Buttons (Form_Buttons.html)

```
1 <a class="btn btn-secondary" href="#" role="button">
2   Link
3 </a>
4 <button class="btn btn-secondary" type="Submit">
5   Button
6 </button>
7 <input class="btn btn-secondary"
8         type="button" value="Input">
9 <input class="btn btn-secondary"
10        type="submit" value="Submit">
```



Figure 5-27. Buttons—several variants of the same effect

Forms

If `<a>` tags are used as a button, they should only serve the navigation and not trigger as a hyperlink because this is not the expected behavior. Moreover, it should be emphasized by the attribute `role="button"`. Even then it is questionable whether this is a good idea as not all browsers can ensure that the function will behave as expected. So take advantage of the distinct buttons and whenever possible it is always preferable to use `<button>`.

Forms

Semantic Buttons

Buttons are available in seven variants with the following semantic expressions (see Listing 5-28 and Figure 5-28).

- Primary: Raises the main function, visual effect by blue color amplified
- Secondary: Standard, gray, general function
- Success: Success, green, positive, or affirmative action
- Info: Information, azure, clarified highlighting critical or special actions
- Warning: Warning, orange, critical or complex action, should be taken with caution
- Danger: Danger, red, dangerous or irreversible action, negative
- Link: Formats a button as a link, often referred to as reduced action, additional behavior

Forms

Listing 5-28. Semantic buttons

```
1 <button type="button" class="btn btn-secondary">Std.</button>  
2  
3 <button type="button" class="btn btn-primary">Primary</button>  
4  
5 <button type="button" class="btn btn-success">Success</button>  
6  
7 <button type="button" class="btn btn-info">Info</button>  
8  
9 <button type="button" class="btn btn-warning">Warning</button>  
10  
11 <button type="button" class="btn btn-danger">Danger</button>
```



Figure 5-28. Semantic buttons

Forms

Size and Appearance

For the larger or smaller button classes `.btn-lg`, `.btn-sm`, or `.btn-xs` should be used (see Figure 5-29). They complement the base class `btn` and are independent of color. The standard size is ready for touch operation. Mouse controlled sites will look better with the smaller buttons.

```

1  <p>
2    <button type="button" class="btn btn-primary btn-lg">
3      Big
4    </button>
5    <button type="button" class="btn btn-secondary btn-lg">
6      Big
7    </button>
8  </p>
9  <p>
10   <button type="button" class="btn btn-primary">
11     Standard
12   </button>
13   <button type="button" class="btn btn-secondary">
14     Standard
15   </button>
16 </p>
17 <p>
18   <button type="button" class="btn btn-primary btn-sm">
19     Small
20   </button>
21   <button type="button" class="btn btn-secondary btn-sm">
22     Small
23   </button>
24 </p>
25 <p>
26   <button type="button" class="btn btn-primary btn-xs">
27     Tiny
28   </button>
29   <button type="button" class="btn btn-secondary btn-xs">
30     Tiny
31   </button>
32 </p>

```

Forms



Figure 5-29. *Size of the buttons*

Forms

Sometimes buttons will completely fill their containers. This is the purpose of the class `.btn-block` (see Listing 5-29).

Listing 5-29. Buttons horizontally extended (Form_ButtonsContainer.html)

```
1 <button type="button"  
2     class="btn btn-primary btn-lg btn-block">  
3   Block  
4 </button>  
5 <button type="button"  
6     class="btn btn-secondary btn-lg btn-block">  
7   Block  
8 </button>
```

Note here that this only affects the width. Appropriate height must be reached separately with `.btn-lg` or other size as necessary (see Figure 5-30).



Figure 5-30. Buttons horizontally expanded

Additional Modules

Responsive Images

Images typically have natural expansion. On large screens this is not critical, however shown on small devices they may leave the viewing area. With `.img-fluid` you can present better images.

This class enforces three actions:

- `max-width: 100%`
- `height: Auto`
- `display: Block`

Thus, the image is scaled to the device width and remains stable. Should it be extremely reduced compared to the natural size, you may want to consider scaling with a server-side solution. By not doing so you waste a lot of bandwidth and have little effect on the user.

If `.img-fluid` is used, there are other options available. The class `.center-block` centers the image in the container.

Additional Modules

It is easy to put images in special forms when using following class (see Listing 6-3 Figure 6-6):

- .img-thumbnail: Thumbnails

Listing 6-3. Effects on pictures (Image_Effects.html)

```
1 
```



Figure 6-6. Effects on pictures

Additional Modules

Text Color

Text colors are again some classes that have a slightly semantic meaning (see Listing 6-4 and Figure 6-7). The primary classes are as follows:

- `text-muted`: Suppressed, bright, and gray
- `text-primary`: Primary, main action, or primary statement, blue,
- `important`
- `text-success`: Success, green, positive, or success
- `text-info`: Information, azure, highlight, attention needed
- `text-warning`: Warning, orange, action is consequential or message is critical
- `text-danger`: Danger, red, errors or irreversible, serious warning

Additional Modules

Listing 6-4. Colors for text (Text_Colors.html)

```
1 <p class="text-muted">...</p>  
2 <p class="text-primary">...</p>  
3 <p class="text-success">...</p>  
4 <p class="text-info">...</p>  
5 <p class="text-warning">...</p>  
6 <p class="text-danger">...</p>
```



Figure 6-7. Text colors

Additional Modules

Background Color

Available background colors are similar classes that have a slightly semantic meaning (see Figure 6-8). The primary criteria are as follows:

- Primary: Primary, main action or primary statement, blue
- Success: Success, green, positive or success
- Info: Information, violet, highlight, attention needed
- Warning: Warning, orange, action is consequential or message is critical
- Danger: Danger, red, errors or irreversible, serious warning

Additional Modules

```
1 <p class="bg-faded">...</p>  
2 <p class="bg-primary">...</p>  
3 <p class="bg-success">...</p>  
4 <p class="bg-info">...</p>  
5 <p class="bg-warning">...</p>  
6 <p class="bg-danger">...</p>
```

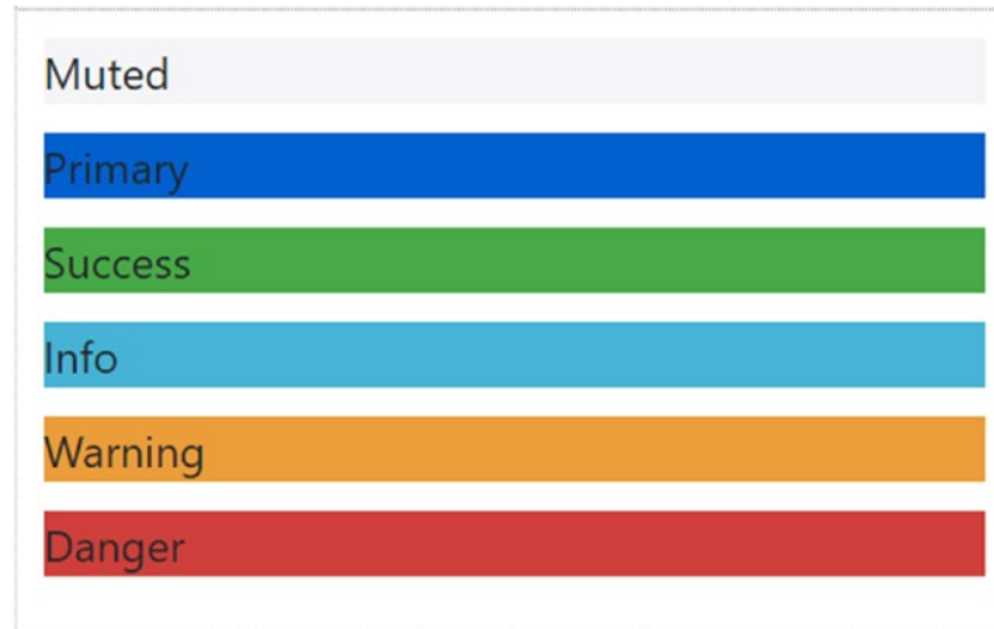


Figure 6-8. Background colors

Components

Drop-Down Menus

Drop downs—context or pop-up menus—are an integral part of many forms. The interaction is complex and partly based on JavaScript. It must not be activated separately, the data-attributes attach here, namely `data-toggle="dropdown"`. Even menus benefit from symbols.

In Bootstrap 4 menus are not created with `` any longer, but with `<div>` and regular buttons or anchor tags that match `.dropdown-menu` or `.dropdown-item` and information can be added to them (see Listing 7-1 and Figure 7-1).

Components

Listing 7-1. Menu folds downward (Icons_DropDown.html)

```

1  <div class="dropdown">
2    <button class="btn btn-default dropdown-toggle"
3          type="button"
4          id="dropdownMenu1"
5          data-toggle="dropdown"
6          aria-haspopup="true" aria-expanded="true">
7      Fold down
8      <span class="caret"></span>
9  </button>
10 <div class="dropdown-menu" aria-labelledby="dropdownMenu1">
11   <a class="dropdown-item" href="#">Edit</a>
12   <a class="dropdown-item" href="#">Delete</a>
13   <a class="dropdown-item" href="#">Details</a>
14   <a class="dropdown-item" href="#">Lock</a>
15 </div>
16 </div>

```

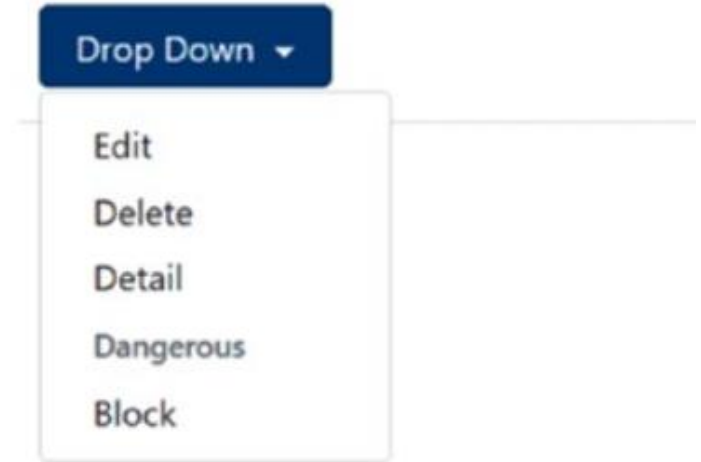


Figure 7-1. Menu folds downward

Components

Listing 7-2. Menu tilted upward (Icons_DropUp.html)

```
1  <div class="dropup">
2    <button class="btn btn-default dropdown-toggle"
3          type="button"
4          id="dropdownMenu2" data-toggle="dropdown"
5          aria-haspopup="true" aria-expanded="false">
6      Fold Up
7      <span class="caret"></span>
8  </button>
9  <div class="dropdown-menu" aria-labelledby="dropdownMenu2">
10     <a class="dropdown-item" href="#">Edit</a>
11     <a class="dropdown-item" href="#">Delete</a>
12     <a class="dropdown-item" href="#">Details</a>
13     <a class="dropdown-item" href="#">Lock</a>
14  </div>
15 </div>
```



Figure 7-2. Menu tilted upward

Components

Navigation

The navigation elements are all initiated by a common base class `.nav`. When navigating by tabulators, JavaScript is required. To support the accessibility attribute `role="navigation"` should be used in the logic overlying containers. This gives the navigation the necessary semantic meaning. The important thing is not to do so on the ``- element, but on the surrounding `<nav>` or `<div>` element.

Basically distinction should be made in navigation between content navigation and navigation action. Content can be accessed via links, buttons, or tabs. However navigation action takes place over buttons, menus, toolbars, menu strips, and so forth. Forms are among the content. Only the submit button, for example, is an action.

Components

Listing 7-12. Tabs (Nav_tab.html)

```
1 <ul class="nav nav-tabs">
2   <li class="nav-item" role="presentation">
3     <a data-toggle="tab" class="nav-link active" href="#">
4       Start
5     </a>
6   </li>
7   <li class="nav-item" role="presentation">
8     <a data-toggle="tab" class="nav-link" href="#">
9       Profile
10    </a>
11  </li>
12  <li class="nav-item" role="presentation">
13    <a data-toggle="tab" class="nav-link" href="#">
14      News
15    </a>
16  </li>
17 </ul>
```

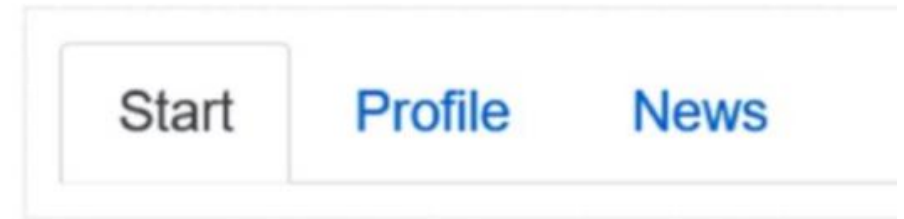


Figure 7-12. Tabs

Components

Deactivation

It may happen that the content in question or part of the form does not need to be reached. In this case the item is disabled (see Listing 7-14 and Figure 7-14). It is then grayed out and does not respond to the mouse or touch actions.

Listing 7-14. Disabled navigation buttons (Nav_Pills_Deactivated.html)

```
1 <ul class="nav nav-pills">
2   ...
3   <li role="presentation" class="nav-item">
4     <a class="nav-link disabled" href="#">Deactivated Link</a>
5   </li>
6   ...
7 </ul>
```

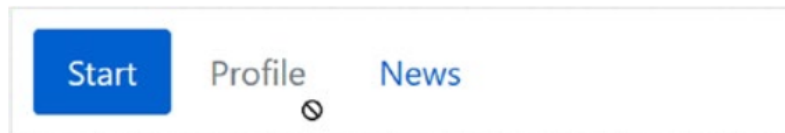


Figure 7-14. Disabled navigation buttons

Components

Page Scrolling

Even page-by-page scrolling belongs to the navigation elements. Usually it occurs in the context of data tables. But even multipage content pages can be browsed.

A typical setup for a few pages or a reasonably constant number of pages looks like this:

« 1 2 3 4 5 »

Components

Listing 7-19. Page scrolling (Nav_Pagination.html)

```

1  <nav>
2    <ul class="pagination">
3      <li class="page-item">
4        <a href="#" aria-label="Previous">
5          <span aria-hidden="true">&laquo;</span>
6        </a>
7      </li>
8      <li class="page-item"><a href="#">1</a></li>
9      <li class="page-item"><a href="#">2</a></li>
10     <li class="page-item"><a href="#">3</a></li>
11     <li class="page-item"><a href="#">4</a></li>
12     <li class="page-item"><a href="#">5</a></li>
13     <li class="page-item">
14       <a href="#" aria-label="Next">
15         <span aria-hidden="true">&raquo;</span>
16       </a>
17     </li>
18   </ul>
19 </nav>

```



Figure 7-20. Page scrolling

Thank You

Alfred Tenggono, S.Kom., M.Kom.

alfred.tenggono@jiu.ac

Reference

- Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, ISBN: 978-1-491-96020-2
- <https://www.figma.com/>