HTML Forms

Learning Objectives

- Understanding the purpose and structure of forms
- Knowing most common types of form fields
- Building useable and accessible forms
- Understanding the different types of buttons

Basic form setup

`<form>` tag

The `<form>` tag must be wrapped around the complete form with all elements, that are presented as form controls to the user.

```
<form>
  <!-- All form elements inside -->
</form>
```

Form field names

Forms are created to request information from the user. Each fragment of information (each form field) requires a unique name. It can be set with the `name` attribute and pairs up with the entered data, when submitting the form.

```
<input name="first-name" />
```

Labels

The `<label>` always goes together with a form field. It provides a caption to let users understand, what kind of data they are asked to enter.

It is required to define, which label and form field belong together. Use the `for` attribute on the `<label>` and the `id` attribute on the form field. Their values needs to match.

```
<label for="first-name">First name</label>
<input name="first-name" id="first-name" />
```

> ❗ Always add a label to a form field. Otherwise users won't understand the purpose of a field, which makes it unusable.

> ❗ Never use the `placeholder` attribute instead of a label.

Different types of form fields

Text

The default `type` for `<input>` elements is `text`. Choose the `type` based on the kind of data the user is requested to enter. Use `type="text"` when none of the other types is a better fit.

```
<label for="first-name">First name</label>
<input type="text" name="first-name" id="first-name" />
```

Email

Use `type="email"` to let the user enter an email address. The browser can check automatically, whether the entered text is a valid email address. .

```
<label for="email-address">Email address</label>
<input type="email" name="email-address" id="email-address" />
```

Number

Use `type="number"` to let the user enter a number.

```
<label for="age">Age</label> <input type="number" name="age" id="age" />
```

Date

Use `type="date"` to let the user enter a date with the help of a date picker (calendar) provided by the browser.

```
<label for="date-of-birth">Date of birth</label>
<input type="date" name="date-of-birth" id="date-of-birth" />
```

Color

Use `type="color"` to let the user enter a color with the help of a color picker tool provided by the browser.

```
<label for="favorite-color">Favorite color</label>
<input type="color" name="favorite-color" id="favorite-color" />
```

Multi-line text

Use the tag `<textarea>` to let the user enter longer text with multiple lines.

```
<label for="personal-message">Personal Message</label>
<textarea name="personal-message" id="personal-message"></textarea>
```

> ❗ Please be aware that the `<textarea>` tag is not a self-closing tag like `<input>`.

Select / dropdown menu

The `<select>` field lets the user choose between different options wrapped into `<option>` tags that are nested into their parent `<select>` tag - this renders a dropdown menu. Each `<option>` has a `value` attribute defining the data to be submitted. The option's text presented to the user is defined between the opening and closing tag.

```
<label for="billing-plan">Billing plan</label>
<select name="billing-plan" id="billing-plan">
  <option value="weekly">Weekly billing</option>
  <option value="monthly">Monthly billing</option>
  <option value="yearly">Monthly billing</option>
</select>
```

Radio elements

The `<input type="radio" />` element is another way of presenting a choice with different options to the user. In many situations it can be used as an alternative to `<select>`.

```
<input
  type="radio"
  name="billing-plan"
  id="billing-plan-weekly"
  value="weekly"
/>
<label for="billing-plan-weekly">Weekly billing</label>

<input
  type="radio"
  name="billing-plan"
  id="billing-plan-monthly"
  value="monthly"
/>
<label for="billing-plan-monthly">Monthly billing</label>

<input
  type="radio"
  name="billing-plan"
  id="billing-plan-yearly"
  value="yearly"
/>
<label for="billing-plan-yearly">Yearly billing</label>
```

> ❗ The `name` attribute must be equal among all radio elements that refer to the same choice. The browser groups them together and ensures only one radio element can be selected at the same time.

Checkboxes

In contrast to the radio element, `<input type="checkbox" />` presents individual choices, that are not related to each other. Each choice can either be "on" ("true") or "off" ("false").

```
<input type="checkbox" name="accept-data-privacy" id="accept-data-privacy"
/>
<label for="accept-data-privacy">I accept the data privacy agreement
</label>

<input
  type="checkbox"
  name="accept-terms-conditions"
  id="accept-terms-conditions"
/>
<label for="accept-terms-conditions">I accept the terms and conditions
</label>
```

> ❗ The `name` attribute must not be equal among the checkbox elements. They are used to represent individual choices.

More form field types

The different types of `<input>` elements described above is just a small selection. Please refer to the MDN web docs to see a complete list of all types with examples.

Buttons

Submit button

The default `type` for a `<button>` element is `submit`. It is used to let users submit the form's data after they filled out all fields.

```
<button type="submit">Send</button>
```

> 💡 Since it's the default, it would work the same without the type attribute: `<button>Send</button>`.

Reset button

A button with `type="reset"` lets the user reset all form fields to the their default value on click.

```
<button type="reset">Reset all fields</button>
```

Other buttons

Since `type="submit"` is the default for `<button>` elements, buttons outside of a `form` element should be defined with `type="button"` to be semantically correct.

```
<button type="button">Click here for more information</button>
```

This also applies to buttons with diverging functionality within a form.

Form structure and a11y

Fieldset and Legend

The `<fieldset>` element is used to group multiple fields together. Use the `<legend>` element to provide a caption for such a group.

```
<fieldset>
  <legend>Personal information</legend>

  <label for="first-name">First name</label>
  <input type="text" name="first-name" id="first-name" />

  <label for="email">Email address</label>
  <input type="email" name="email" id="email" />
</fieldset>
```

`aria` labels

aria-label

The `aria-label` attribute defines a label for an interactive element. Use it when the accessible name is missing and there is no content visible in the DOM that can be referenced via the `aria-labelledby` attribute, e.g. a button with no text but only an icon:

```
<button aria-label="Close form" onClick="...">
  <svg ...><path ... /></svg>
</button>
```

aria-labelledby

The `aria-labelledby` attribute identifies which element labels the element it is applied to. Use the `id` attribute to create the connection:

```
<h2 id="title">Personal Information Form</h2>
<form aria-labelledby="title">...</form>
```

aria-describedby

The `aria-describedby` attribute allows more verbose information than a label. Use the `id` attribute to create the connection:

```
<p id="description">
  We need some personal information about you in order to proceed. Please
fill
  in this form so that we can help you.
</p>
<fieldset aria-describedby="description">...</fieldset>
```

Resources

- `<form>`: The Form element
- `<input>`: The Input (Form Input) element
- Forms Tutorial W3C
- Forms (Web Accessibility Guidelines)