# HTML5 and CSS3

Margit Tennosaar

# Last week

Links // <a>, external (http://...) and internal (.../about.html or #footer)

Images in HTML and CSS // external and internal. Part of content to HTML and others to CSS

Lists // <ul> <li></li></ul>

CSS combinators // main p, main > p, main + p, main ~ p

CSS colors // #ff0000, rgb(255, 0, 0), rgba(255, 0, 0, 0.3), green

# Last week

HTML tables // <table>, <thead>, <tbody>, <tfoot>, <tr>, <th>, <td>

CSS borders // 1px solid green

CSS backgrounds  // background: #333 url("http://example.com/image.png") no-repeat center center/cover;

Box model // top, right, bottom, left; padding inside, margin outside

Universal reset // * { margin: 0; padding: 0; box-sizing: border-box; }

Float // use clear to end float in next elements

# Today

Fonts

Form

Input attributes

Input types

- Radiobuttons

- Checkboxes

Select

Textarea

Labels

Submitting

# Fonts

# Types

- **Serif fonts** have a small stroke at the edges of each letter. They create a sense of formality and elegance.

- **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.

- **Monospace fonts** - here all the letters have the same fixed width. They create a mechanical look.

- **Cursive** fonts imitate human handwriting.

- **Display** fonts are decorative/playful fonts.

# Font in CSS

font-family

font-size

font-weight // normal, bold, 100-900

line-height // 1; 1.2; 2;

# Fallback font

A fallback font is a reserve typeface containing symbols for as many Unicode characters as possible. When a display system encounters a character that is not part of the repertoire of any of the other available fonts, a symbol from a fallback font is used instead.

```
font-family: 'Lato', sans-serif;
```

# Forms

# Form

The HTML <form> element defines a form
that is used to collect user input

Form elements are different types of input elements:

- text fields,

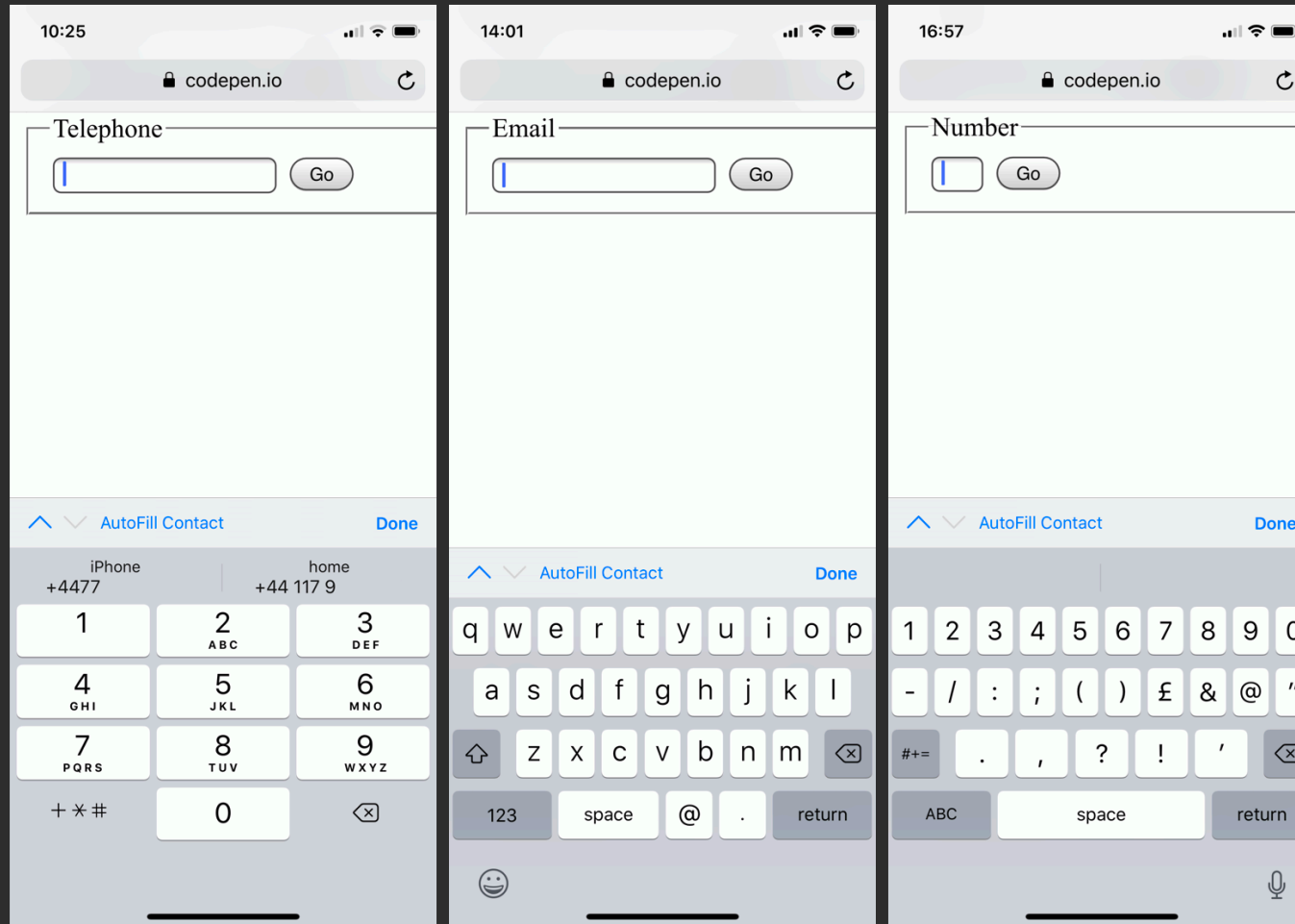- checkboxes,

- radio buttons,

- submit buttons,

- and more

*https://www.w3schools.com/tags/tag_form.asp*

# Input / text fields

The <input> element can be displayed in several ways, depending on the type attribute.

Most used input attributes are: type, value, readonly, disabled, size, maxlength, pattern, required, placeholder

Note: name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted.

*https://www.w3schools.com/html/html_form_attributes.asp*

# Input types

# Input types

<input type="button">

<input type="checkbox">

<input type="color">

<input type="date">

<input type="email">

<input type="file">

<input type="hidden">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="month">

<input type="week">

# 4th task

Make yourself a webpage, where you have in table examples with different input types

| Description | Input example | Notes |
|---|---|---|
| type="button" | I am button | Needs value attribute for button text |
| type="checkbox" | ☐ Checkbox | Checkbox text comes after input tag |
| type="color" | 🟥 | With value you can set default color. type="color" is not supported in Internet Explorer 11 and earlier versions or Safari 9.1 and earlier versions. |
| type="date" | dd.mm.yyyy | type="date" is not supported in Safari or Internet Explorer 11 and earlier versions. |
| type="email" | | type="email" is not supported in IE9 and earlier. |

# Can I use?

# Radiobuttons and checkboxes

Radio buttons let a user select ONLY ONE of a limited number of choices:

```
<form>
  <input type="radio" name="gender" value="male" checked> Male<br>
  <input type="radio" name="gender" value="female"> Female<br>
  <input type="radio" name="gender" value="other"> Other
</form>
```

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

# Other form elements

The <select> element defines a drop-down list

  - The <option> elements defines an option that can be selected. To define

   a pre-selected option, add the selected attribute to the option

The <textarea> element a multi-line input field (a text area)

The <datalist> element specifies a list of pre-defined options for

an **input** element (similar to <select>).

# 4th task continues

Add <Select><Textarea> to your form elements webpage

# Label

The <label> tag defines a label for a <button>, <input>, <meter>, <output>, <progress>, <select>, or <textarea> element.

The <label> element does not render as anything special for the user. However, it provides a usability improvement for mouse users, because if the user clicks on the text within the <label> element, it toggles the control.

# Clickable inputs

```
<label>Click me <input type="text" /></label>
```

```
<label for="username">Click me</label>
<input type="text" id="username" />
```

# Submit or button?

```
<form>
 <input type="submit" value="Submit">
</form>
```

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

The difference is that the <button> element can have content

Note: Always specify the type attribute for the button element. Different browsers may use different default types for the button element.

# Form validation

Validation done in the browser is called client-side validation, while validation done on the server is called server-side validation.

If the information is correctly formatted, the application allows the data to be submitted to the server and (usually) saved in a database; if the information isn't correctly formatted, it gives the user an error message explaining what needs to be corrected, and lets them try again.

*https://developer.mozilla.org/en-US/docs/Learn/Forms/Form_validation*

# Why validate?

- **We want to get the right data**, in the right format. Our applications won't work properly if our users' data is stored in the wrong format, is incorrect, or is omitted altogether.

- **We want to protect our users' data**. Forcing our users to enter secure passwords makes it easier to protect their account information.

- **We want to protect ourselves**. There are many ways that malicious users can misuse unprotected forms to damage the application.

- **required**: Specifies whether a form field needs to be filled in before the form can be submitted.

- **minlength** and **maxlength**: Specifies the minimum and maximum length of textual data (strings)

- **min** and **max**: Specifies the minimum and maximum values of numerical input types

- **type**: Specifies whether the data needs to be a number, an email address, or some other specific preset type.

- **pattern**: Specifies a regular expression that defines a pattern the entered data needs to follow.

# 5th task

## Sign Up

It's free and only takes a minute

First name

Last name

Email

Password

Confirm Password

SIGN UP

My clicking the Sign Up button, you agree to
our Terms & Conditions and Privacy Policy

Already have an account? Login Here

# Portfolio 3rd week – a static contact form to the footer

HTML

- Contact form in footer

- Contact me link to menu (scrolls to bottom to footer

- Inputs for name, email and textarea to message

- Submit button

CSS

- Form styling