

# HTML and CSS

Margit Tennosaar

# Forms

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

```
<form action="/submit" method="POST">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

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

# Form elements

- <form>
- <label>
- <input>
- <textarea>
- <select>
- <option>
- <fieldset>
- <legend>

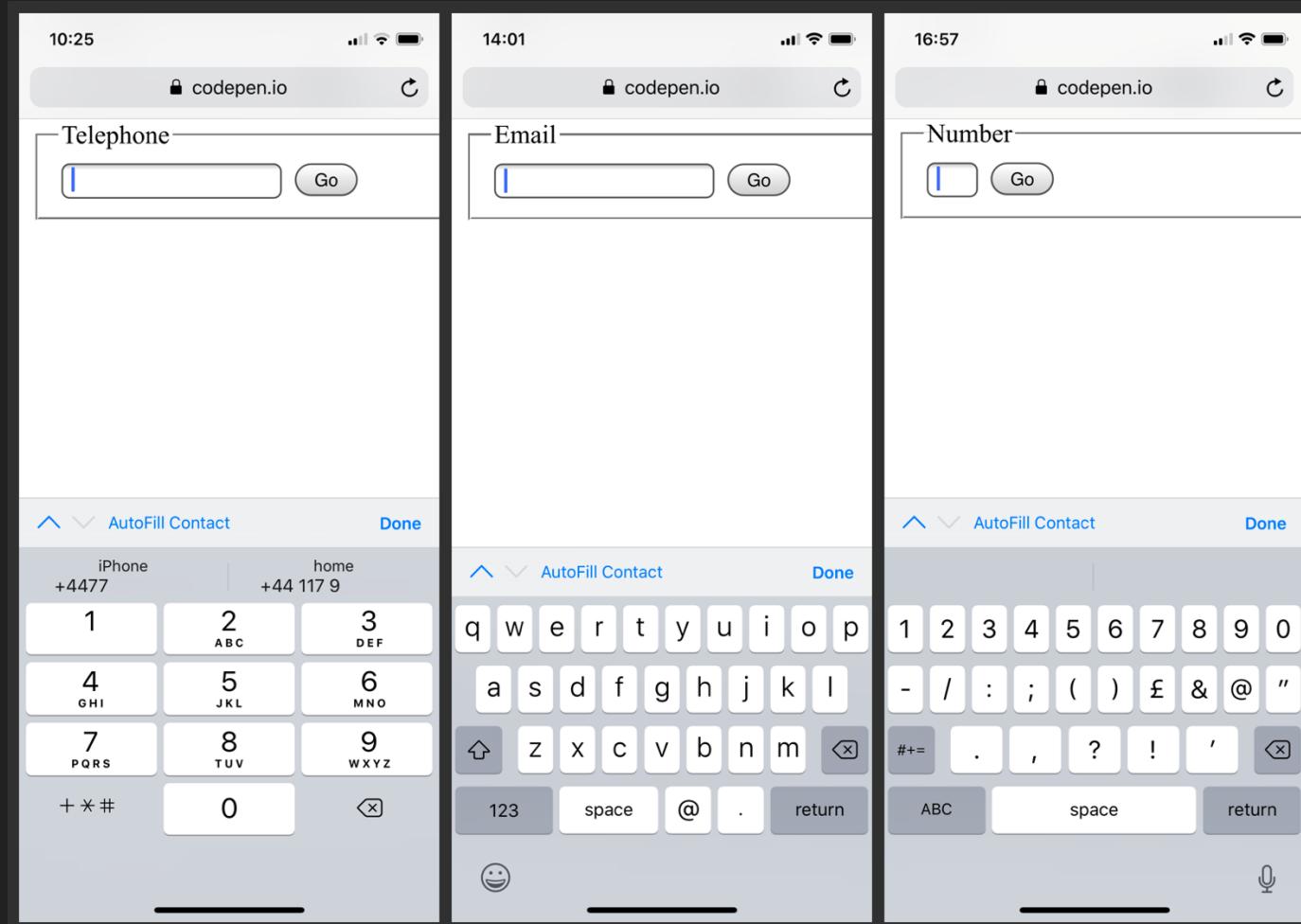
# Input / text fields

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.

# Input types



# Input types

<input type="checkbox">

<input type="color">

<input type="date">

<input type="email">

<input type="file">

<input type="number">

<input type="password">

<input type="radio">

<input type="range">

<input type="search">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

```
<form>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="age">Age:</label>
  <input type="number" id="age" name="age" min="18">

  <label for="dob">Date of Birth:</label>
  <input type="date" id="dob" name="dob">

  <label>Choose an option:</label>
  <input type="radio" id="option1" name="choice" value="1"> Option 1
  <input type="radio" id="option2" name="choice" value="2"> Option 2

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

# 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>`).

# 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.

# Can I use?

Can I use  ? Settings

1 result found

Email, telephone & URL input types - LS

Usage Global % of all users 93.11% + 0.03% = 93.14%

Text input fields intended for email addresses, telephone numbers or URLs. Particularly useful in combination with form validation

IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android	IE Mobile	UC Browser for Android	Sams Intern
6-9		2-3.6	4	3.1-4				1 2.1-2.3							
10	12-17	4-63	5-70	5-11.1	10-56	3.2-11.4		3-4.4.4	1 7	12-12.1			10		4-6
11	18	64	71	12	57	12.1	all	67	10	46	70	63	11	11.8	7.1
		65-66	72-74	TP											

Notes Known issues (0) Resources (2) Feedback

- Specification [html.spec.whatwg.org]
- Article on usage [htmlgoodies.com]

# 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.

# 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.

# Task

## Sign Up

It's free and only takes a minute

First name

Last name

Email

Password

Confirm Password

SIGN UP

By clicking the Sign Up button, you agree to our [Terms & Conditions](#) and [Privacy Policy](#)

Already have an account? [Login Here](#)

# Forms summary

**Form** // container for forms

**Input attributes** // name, type, value etc.

attribute makes input what it is

**Input types**

- **Radiobuttons** // one choice

- **Checkboxes** // multible choices

**Select** // dropdown with options

- **Textarea** // text input for longer texts

- **Labels** // <label for="input\_id">  
<input id="input\_id">

- **Submitting** // button, input – type="submit"