HTML and CSS

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University
https://softuni.bg

Have a Question?

Software University

sli.do

#QA-Auto-FrontEnd

Table of Contents

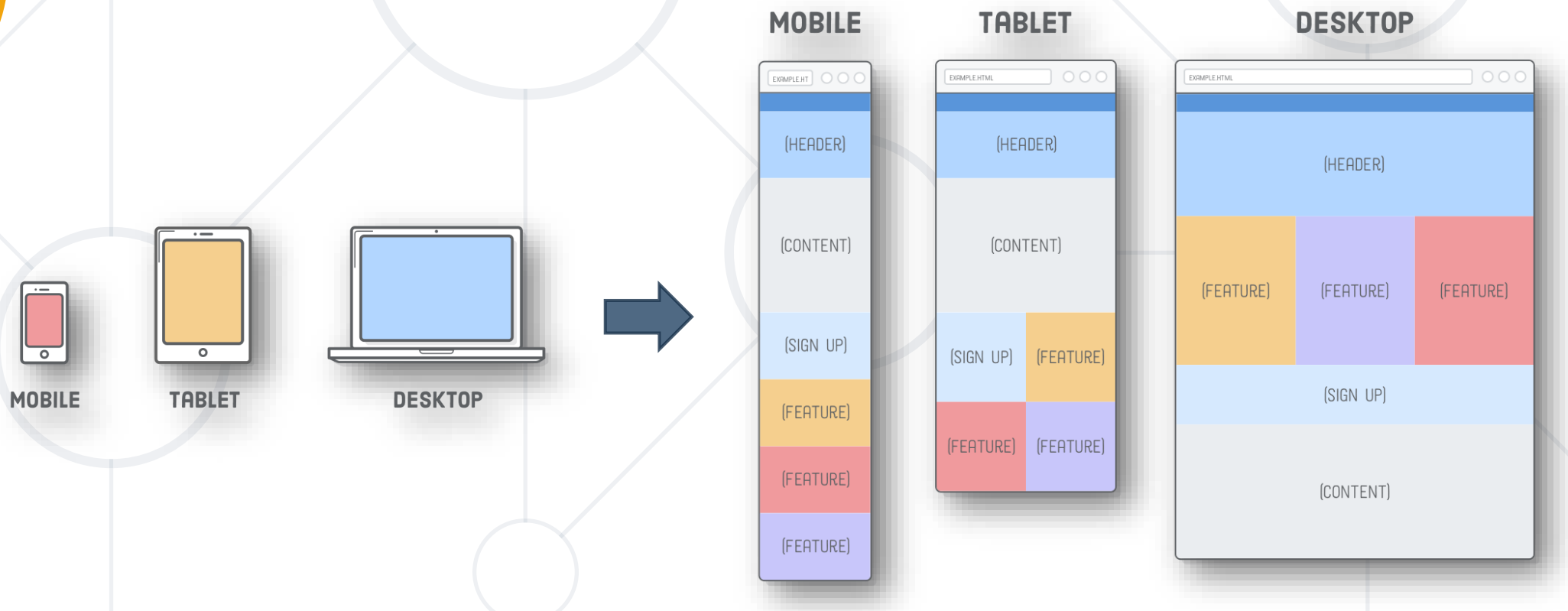
1. Responsive Web Design
2. Media Queries
3. HTML Forms

Responsive Web Design

What is Responsive Web Design?

Software University

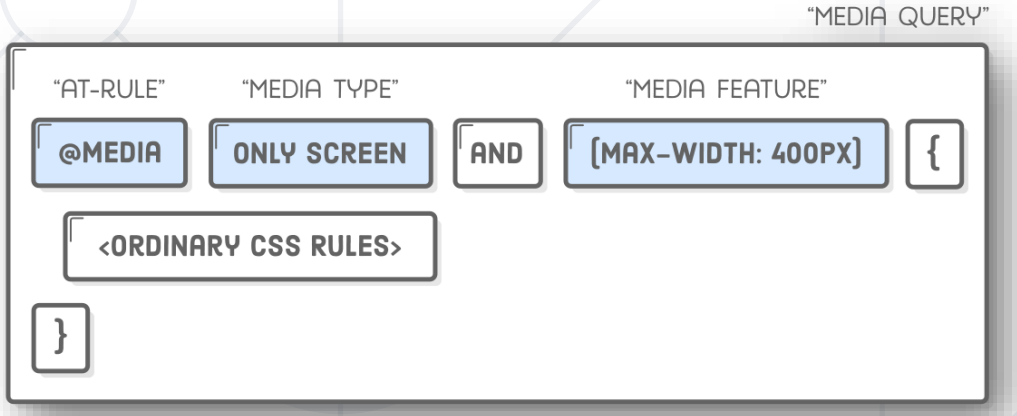
- It is an approach to make web pages render well on **all screen sizes** and **resolutions** while ensuring good usability

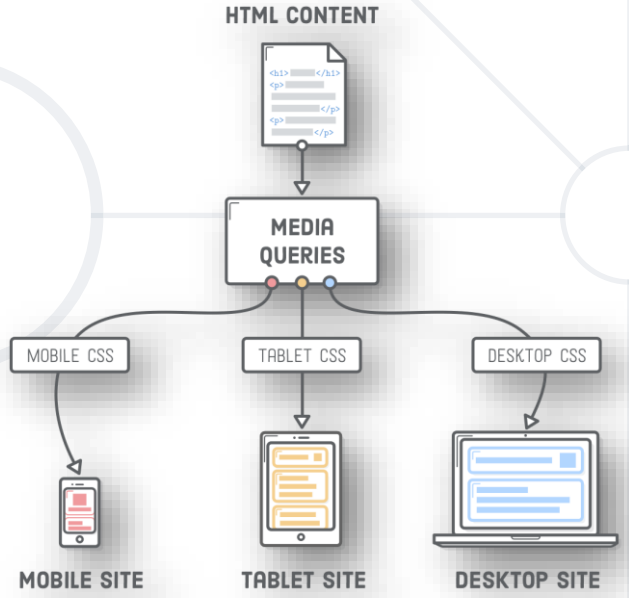


Media Queries

Media Queries

- Allow for running a **series of tests** (e.g., whether the user's screen is greater than a certain width, or a certain resolution) and **apply CSS selectively** to style the page **appropriately** for the user's needs





Media Queries Application

- **Link element's media attribute**
  - Define the media to which a linked resource (typically CSS) should be applied
- **Source element's media attribute**
  - Define the media to which that source should be applied
    - Valid only inside picture elements
- **Style element's media attribute**
  - Define the media to which the style should be applied

Media Types

- Describe the **general category** of a given **device**
  - Although websites are commonly designed with screens in mind, it may be needed to create styles that target special devices such as printers or audio-based screen readers

```
@media print { ... }
@media screen, print { ... }
```

Rules

- `@import`
  - Used to import style rules from other style sheets
  - These rules must precede all other types of rules, except `@charset` rules
  - It is not a nested statement
    - `@import` cannot be used inside conditional group at-rules

```
@import url('styles.css') screen;
```

Media Features

Software University

- Describe a specific characteristic of the user agent, output device, or environment

```
@media screen and (min-width: 600px) {
   …
}
```

```
@media screen and (orientation: landscape) {
   …
}
```

Media Features

- `min-width`
  - Specify a minimal width for the viewport in order to apply the rules included

```
@media screen and (max-width: 600px)
```

- `max-width`
  - Specify a maximum width for the viewport in order to apply the rules included

```
@media screen and (max-width: 600px)
```

Media Queries

- `orientation`
  - Specify a device orientation for which it will apply the rules included

```
@media screen and (orientation: landscape)
```

- `hover` | `pointer`
  - Check for a device ability for which it will apply the rules included

```
@media screen and (hover: hover)
@media screen and (pointer: fine)
```

Logical Operators

- The logical operators **not**, **and**, and **only** can be used to compose a complex media query
- They can also combine **multiple** media queries into a single rule by separating them with **commas** (if more than two)

```
@media screen and (pointer: fine) {
   …
}
```

Logical Operators

- and
  - Used for combining multiple media features together into a single media query, requiring each chained feature to return true in order for the query to be true
  - Also used for joining media features with media types

```
@media screen and (pointer: fine) {
   …
}
```

Logical Operators

- `not`
  - Used to **negate** a media query
  - If present in a comma-separated list of queries, it will only negate the specific query to which it is applied
  - If used, a media type must be specified, too

```
@media screen not (pointer: fine) {
   …
}
```

Logical Operators

- only
  - Used to apply a style only if an entire query matches and is useful for preventing older browsers from applying selected styles
  - If used, a media type must be specified, too

```
@media only screen and (pointer: fine) {
   …
}
```

Logical Operators

- , (comma)
  - Used to combine multiple media queries into a single rule
  - Each query in a comma-separated list is treated separately from the others
    - If any of the queries in a list is true, the entire media statement returns true

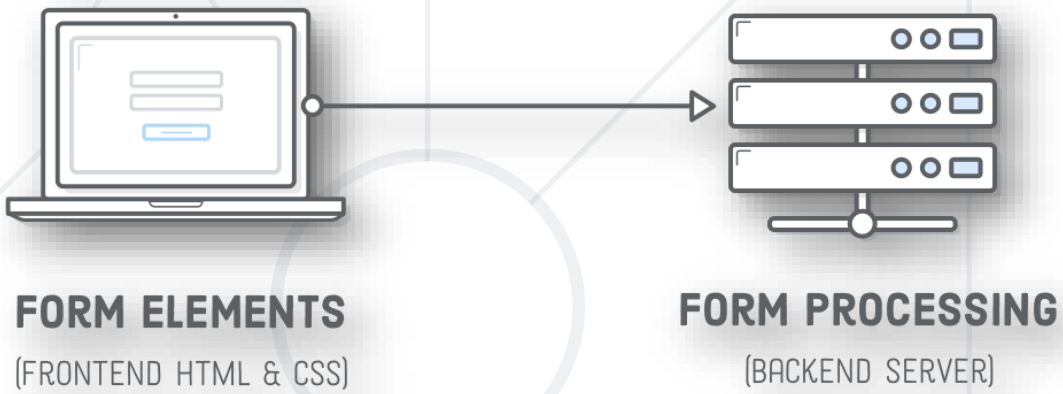
```
@media (min-height: 680px), only screen and (orientation: portrait) {
   ...
}
```

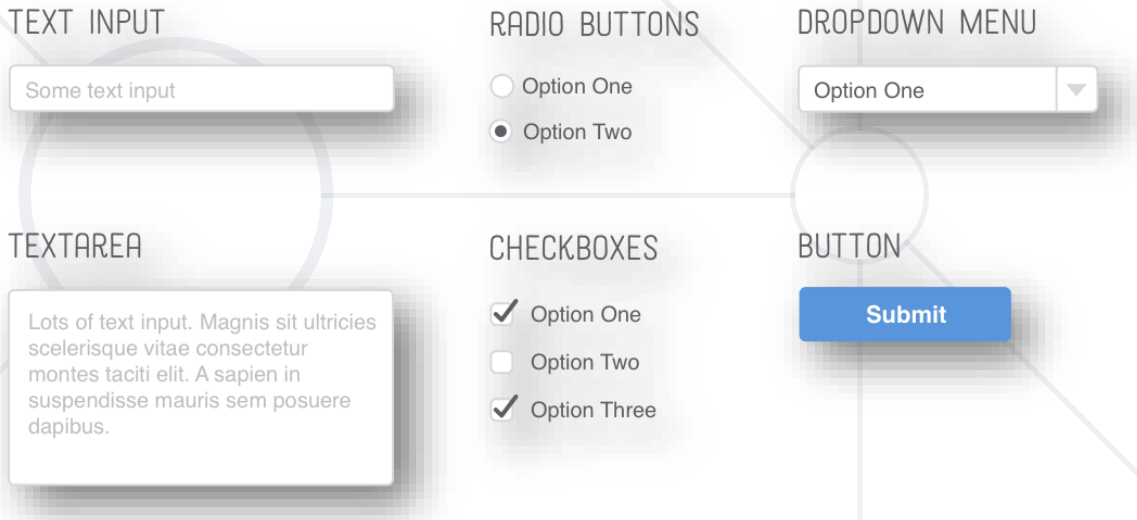
HTML Forms

HTML Forms

- Represent a **document section** that contains **interactive controls** for **submitting information** to a web **server**

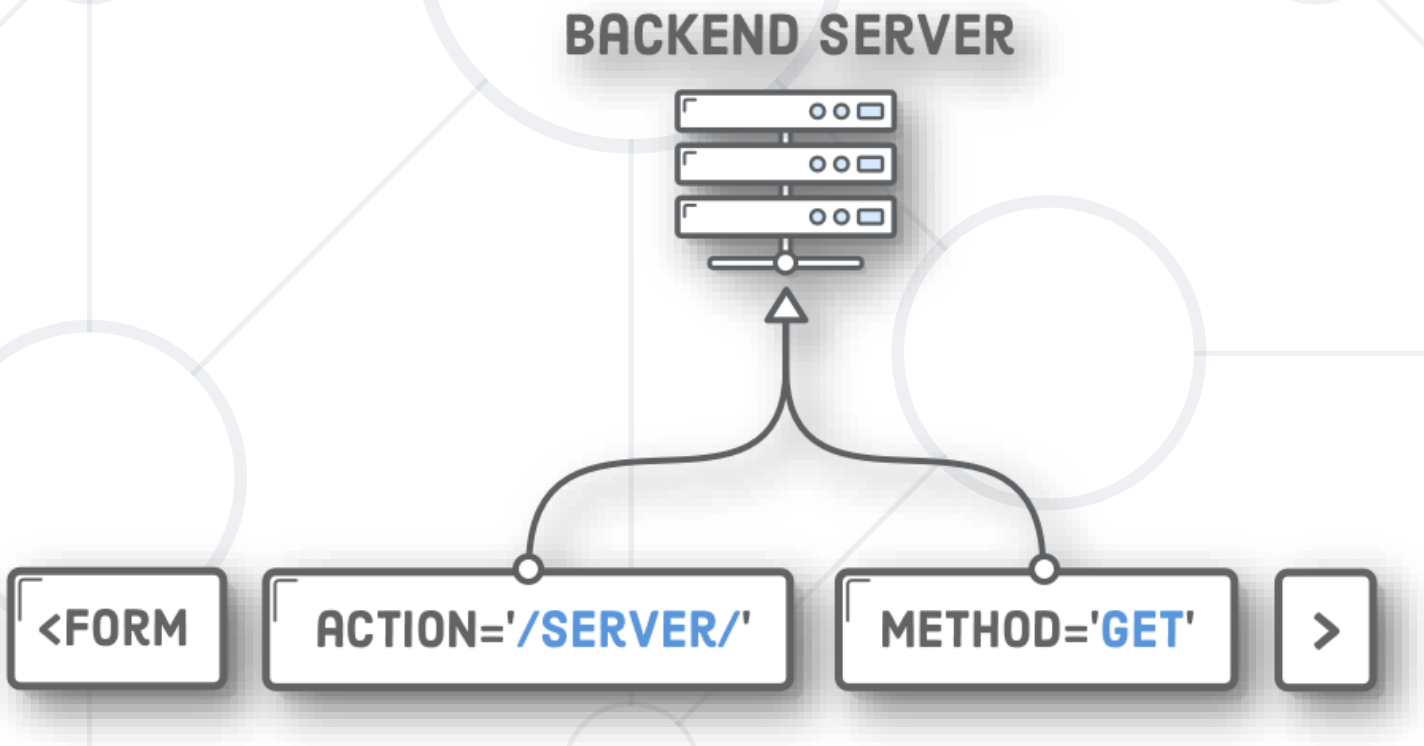
```
<form> … </form>
```





Attributes

- `method`
  - Specify the **HTTP** method that the browser uses to submit the form

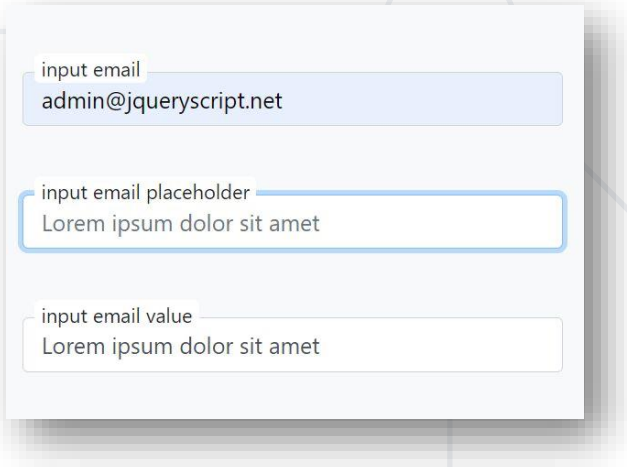


Method Types

- `Post`
  - Corresponds to the HTTP `POST` method
  - Form data is included in the body of the form and sent to the server
- `Get`
  - Corresponds to the HTTP `GET` method
  - Form data is appended to the action attribute `URI` with a `?` as separator
    - The resulting URI is sent to the server

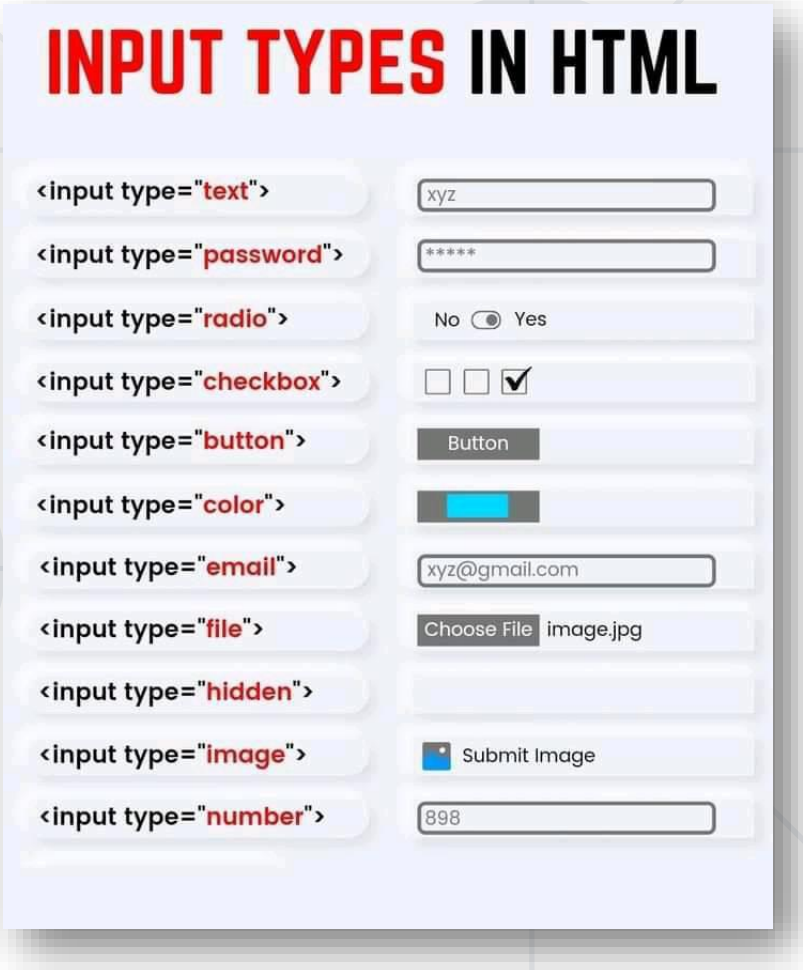
Elements

- `<label></label>`
  - Represent a caption for an item in a user interface
  - The label text is not only visually associated with its corresponding text input
    - It is programmatically associated with it, too



Elements

- `<input></input>`
  - Used to create interactive controls for web-based forms in order to accept data from the user
  - A wide variety of types of input data and control widgets are available, depending on the device and user agent
  - The `<input>` element is one of the most powerful and complex in all of HTML due to the sheer number of combinations of input types and attributes

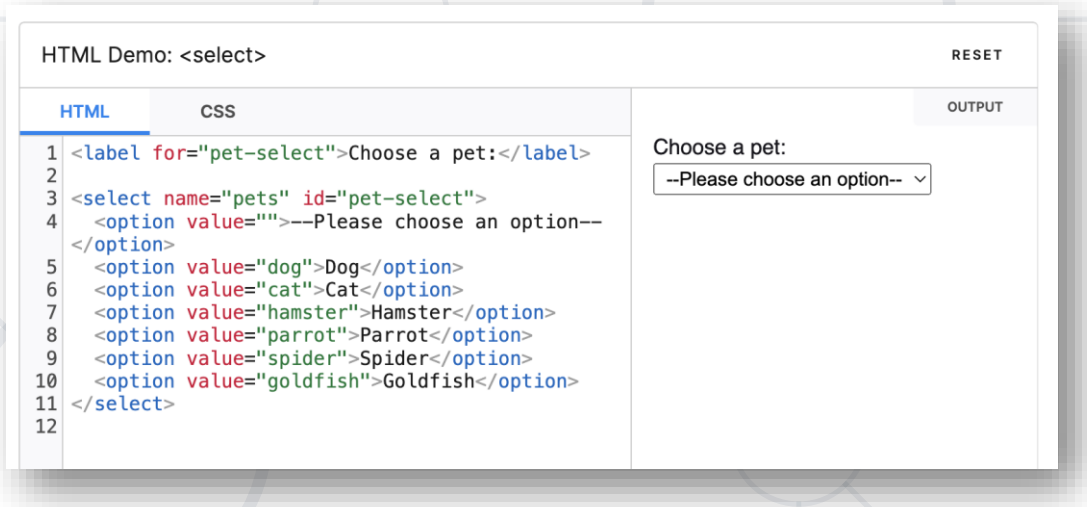


Elements

- `<textarea></textarea>`
  - Represents a multi-line plain-text editing control
  - Useful when you want to allow users to enter a sizeable amount of free-form text

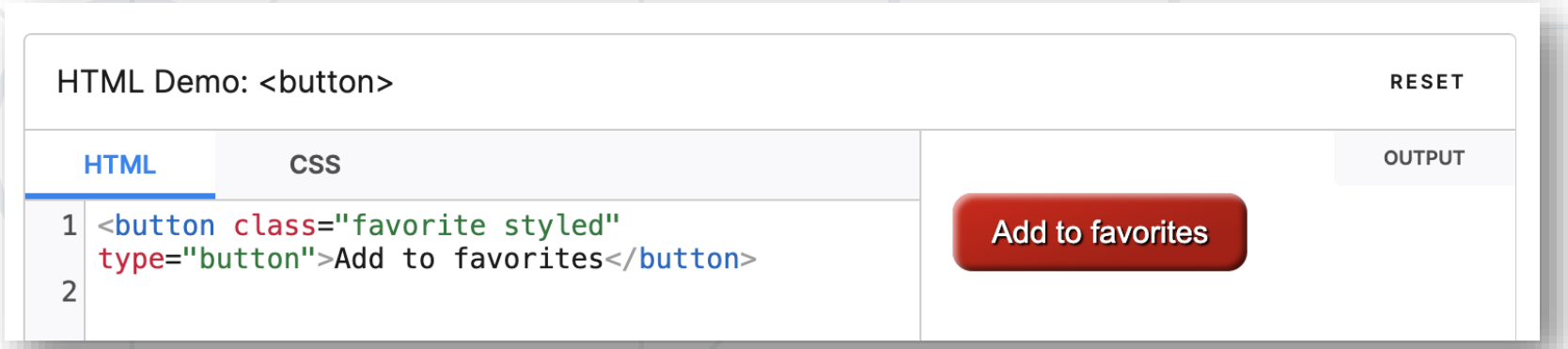
Elements

- `<select></select>` & `<option></option>`
  - The `<select>` HTML element represents a control that provides a menu of options
  - The `<option>` HTML element is used to define an item contained in a `<select>`, an `<optgroup>`, or a `<datalist>` element
    - Can represent menu items in popups and other lists of items in an HTML document



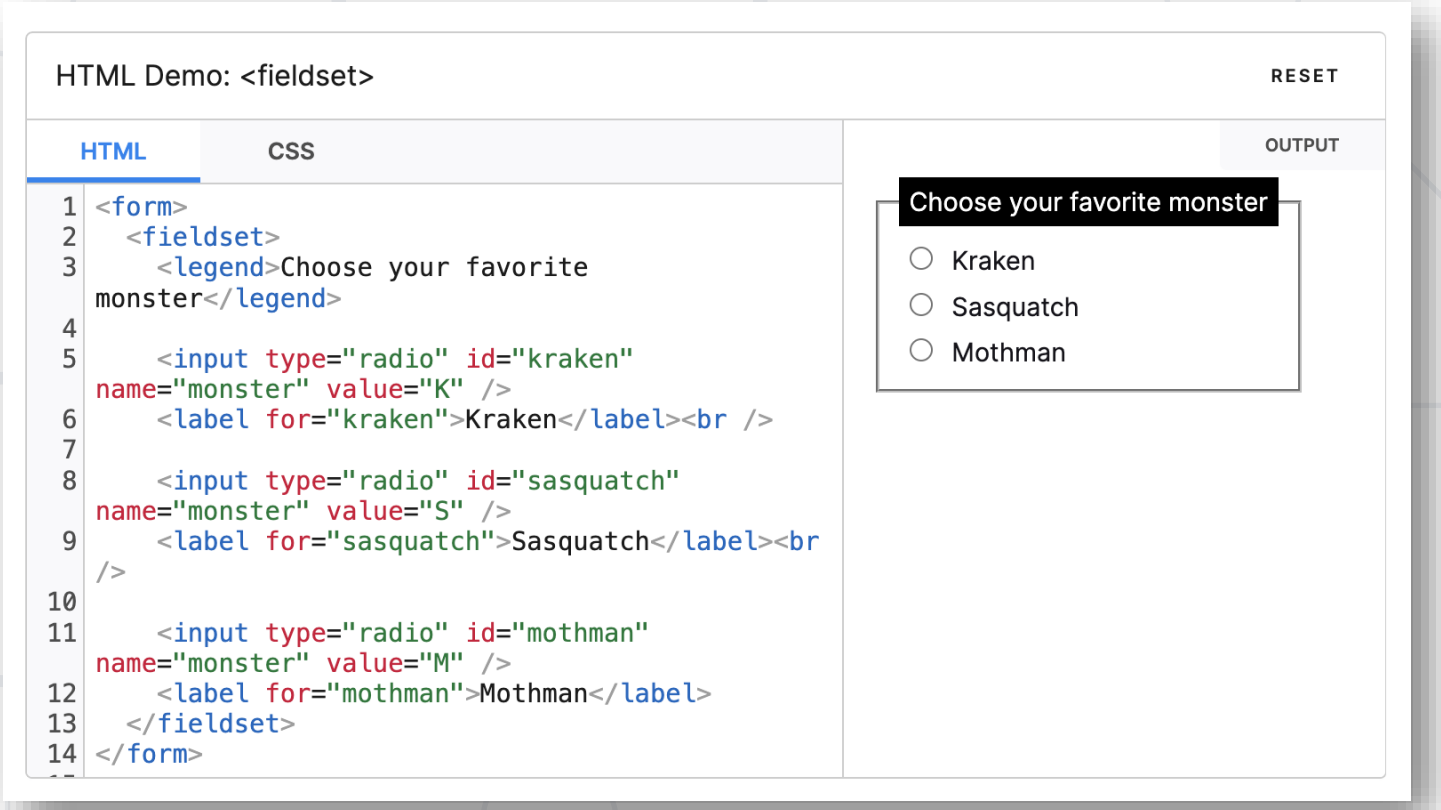
Elements

- `<button></button>`
  - Interactive element
    - Activated by a user with a mouse, keyboard, etc.
    - Once activated, it then performs an action, e.g., submitting a form or opening a dialog



Elements

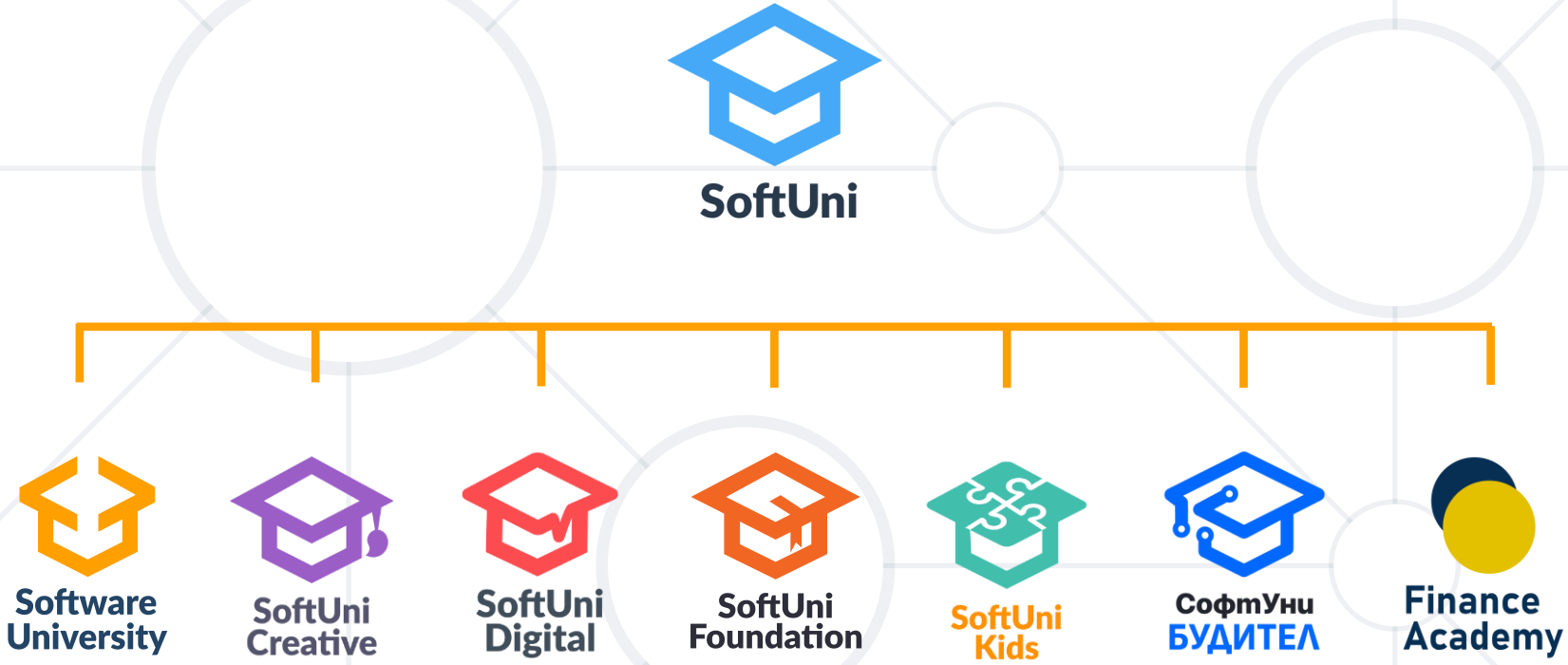
- `<fieldset></fieldset >` & `<legend></legend>`
  - Used to group several controls as well as labels within a web form



Summary

- **Responsive Web Design** == important concept in web applications
  - **Automatically** adjusts content for various screens and viewports
  - Controlled using **rules**, **queries**, **logical operators**
- **HTML forms** == section from a web page for sending data to a web server
  - Controlled using **method types** and HTML **elements**

Questions?



© SoftUni – https://about.softuni.bg. Copyrighted document. Unauthorized copy, reproduction or use is not permitted.

Диамантени партньори

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

SOFTWARE GROUP

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg