HTML 101

The start of your journey

Agenda

- Quiz
- Overview
- Basics
- Document Structure
- Elements
- Tables
- Forms
- Semantic Elements

Overview

Web Page Building Blocks

HTML

describes and defines the structure and content of a web page

CSS

describes the appearance or presentation of the content on a web page

JavaScript

adds interactivity and dynamic features to a web application

HyperText Markup Language

HyperText

Text displayed on a display or other electronic devices with references (**hyperlinks**) to other text that the reader can immediately access

Markup Language

A system for the annotation of documents in a way that is syntactically distinguishable from the content

HTML Characteristics

The **current** standard is **HTML5**

- Published in October 2014 by W3C

We call it simply **HTML.** It brings semantics and a host of new APIs to the spec.

It's a **declarative** language (**WHAT**, does NOT control the flow)

- As opposed to **imperative** (**HOW**, controls the flow of a program)

Obeys Postel's Law / Robustness Principle:

- Be conservative in what you do, be liberal in what you accept from others
- Browsers will not throw errors in case of incorrect HTML code, they will ignore it or try to correct it automatically

Basics

The Code

Written in **files** with **.html** or **.htm** extensions

Created and edited using:

- Text editors: Visual Studio Code, Brackets, Atom, Sublime, Notepad++
- HTML editors (WYSIWYG): Macromedia Dreamweaver

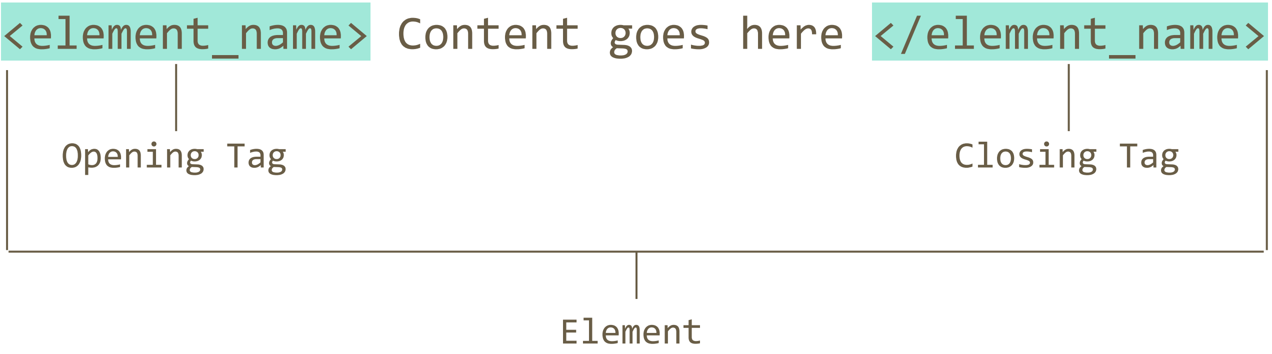
Runs in the **browser** (is parsed and displayed)

The Building Blocks

HTML is made up of **elements**

Elements consist of:

- Opening tag
- Closing tag (/ before the element name)
- Content (text and/or other elements)



```
<element_name> Content goes here </element_name>
```

Opening Tag

Closing Tag

Element

Elements

- Can be nested

```
<p>
    This is a paragraph about <strong>HTML</strong>
</p>
```

- Can be empty
- Some elements consist only of a single (opening) tag
    - Usually used to insert/embed something in the document

```
<img src=“logo.jpg”>
```

Nesting

The practice of putting elements **inside** other elements

```
<p>Some <strong>cool</strong> text!</p>
```

Nested <strong>
element

Proper nesting takes into account the order elements are opened and closed in. Elements should be closed in **reverse** order of opening.

```
<p>Some <strong>cool text!</p></strong>
```

Nesting

Block vs. Inline Elements

Block

- Form a visible block on a page: appear on **a new line** and any content that goes after them will **also appear on a new line**
- Tend to be **structural** elements
- e.g.: **paragraph, header, footer, div**

Inline

- Usually represent **small parts** of the document's content
- Will **not** cause a **new line** to appear in the document
- e.g.: **strong, a, em, span**

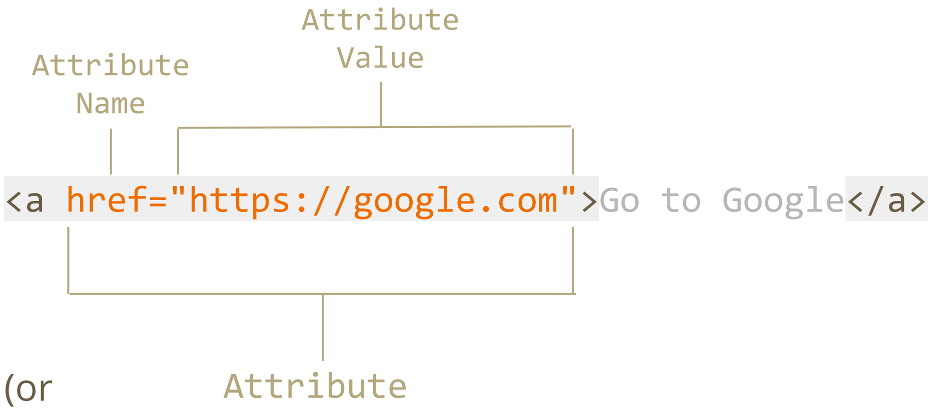
Attributes

Elements can have attributes

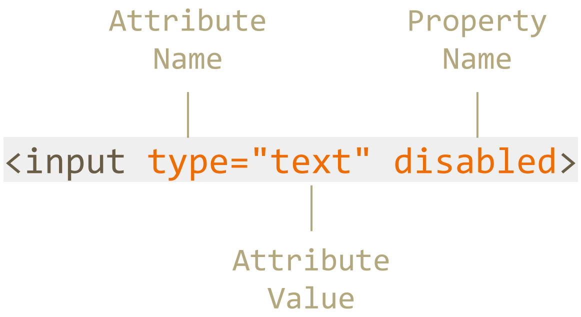
Attributes have:

- Space between them and the tag name (or previous attributes, if there are any)
- **Name**, optionally followed by an = sign
- Optional **value** enclosed in quotation marks (“ ”, preferably double quotes)

Attributes which normally have boolean values (true or false) have their **values omitted** from HTML and are commonly called **properties**. They toggle effects by simply (not) being present on the element.



```
<a href="https://google.com">Go to Google</a>
```



```
<input type="text" disabled>
```

Whitespace and Comments

The HTML parser reduces **multiple spaces** and **line breaks** to a **single space** when rendering the page

Use **as much** whitespace **as necessary** for **readability**!

Recommendations:

- Every block element should start on a **new line** as well as **its content**
- Every nested (block) element should be indented exactly one time
- Line length should not cause the editor to scroll

Comments:

```
<!-- some code here -->
```

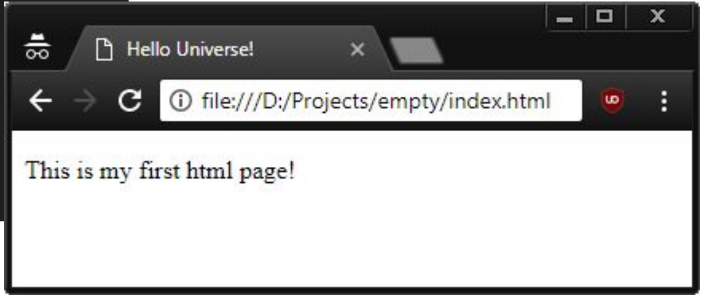
Special Characters

Symbol Name	HTML Entity	Rendered
Copyright Sign	&copy;	©
Registered Trademark Sign	&reg;	®
Less Than	&lt;	<
Greater Than	&gt;	>
Ampersand	&amp;	&
Non-breaking space	&nbsp;	
Euro	&#8364;	€
Em Dash	&mdash;	—

Document Structure

My First HTML Page

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Hello Universe!</title>
</head>
<body>
   <p>This is my first html page!</p>
</body>
</html>
```



This is my first html page!

The <!DOCTYPE> Declaration

- HTML documents **must** start with a document type definition (**DTD**)
- Its initial goal was to link to a **set of rules** that the HTML page had to follow to be considered **valid** HTML
- Old style:

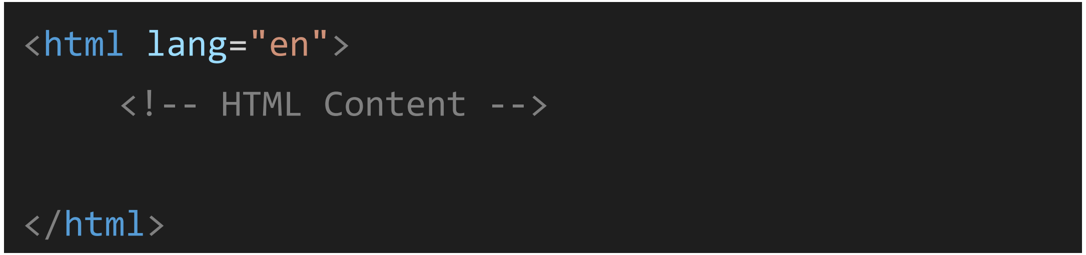
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- HTML 5:

```
<!DOCTYPE html>
```

The <html> Element

- Is the **root** element
- Wraps **all the content** of the page

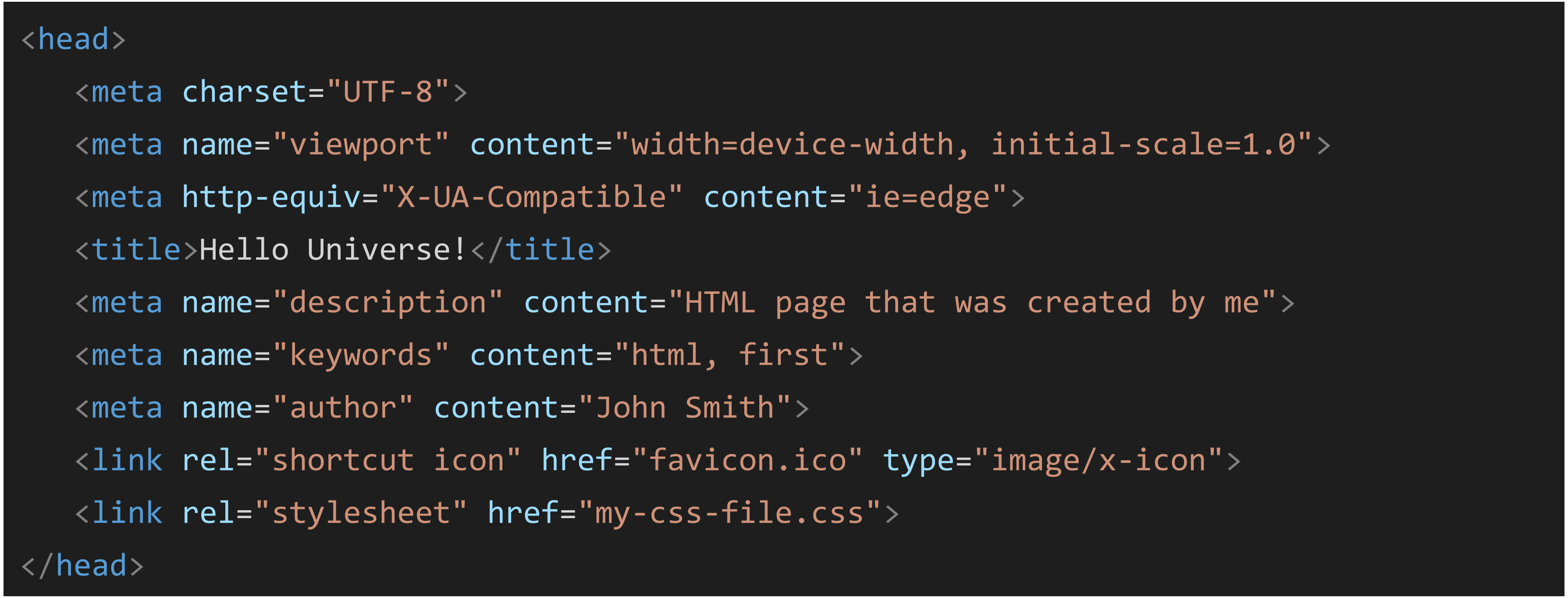


```
<html lang="en">
    <!-- HTML Content -->


</html>
```

The <head> Section

- Its content is not displayed on the page
  - Contains the page’s metadata



```html
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Hello Universe!</title>
    <meta name="description" content="HTML page that was created by me">
    <meta name="keywords" content="html, first">
    <meta name="author" content="John Smith">
    <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
    <link rel="stylesheet" href="my-css-file.css">
</head>
```

The <body> Section

- Wraps all the content that is displayed to the users like:
    - Text
    - Images
    - Videos
    - Audio
    - etc.

Basic Elements

<div>, <p> and <span>

<div>

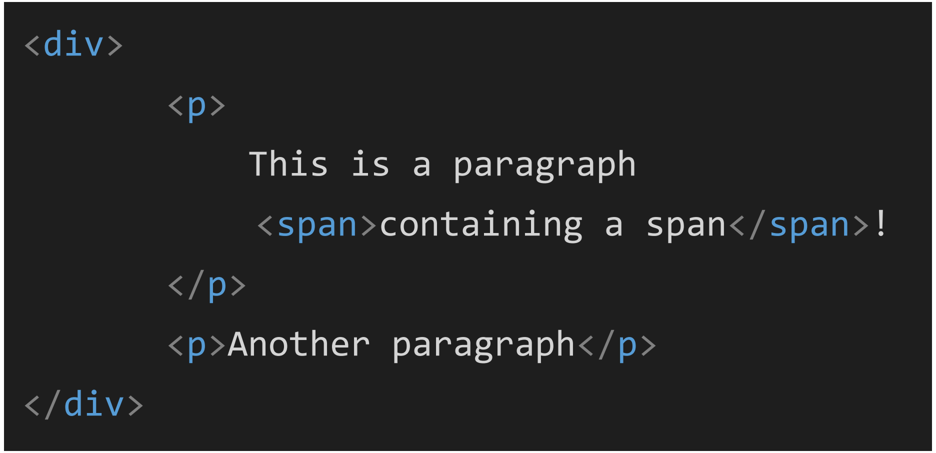
- Creates a **logical division** within the page
- **Block**-level element
- **No semantic value**

<span>

- **Inline** element
- Useful for **targeting** a specific portion of text from CSS
- **No semantic Value**

<p>

- Represents a **paragraph** of text
- **Block**-level element
- **Has** implied **semantics**

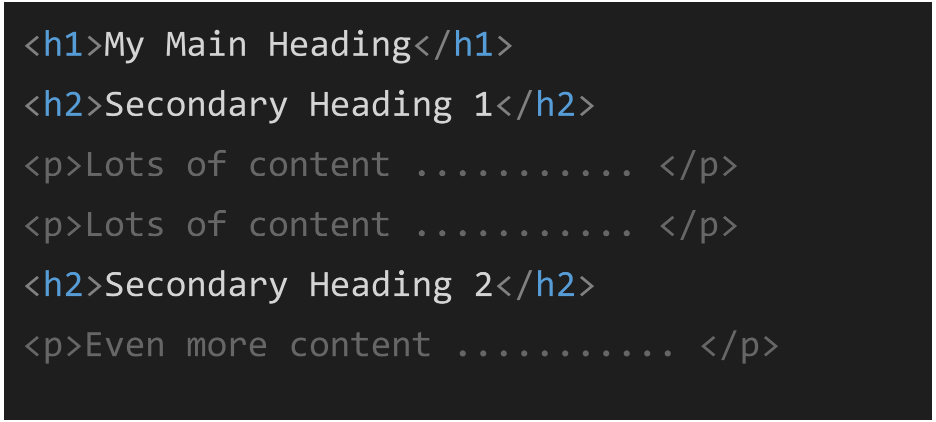


```html
<div>
    <p>
        This is a paragraph
        <span>containing a span</span>!
    </p>
    <p>Another paragraph</p>
</div>
```

Headings

`<h1> - <h6>`

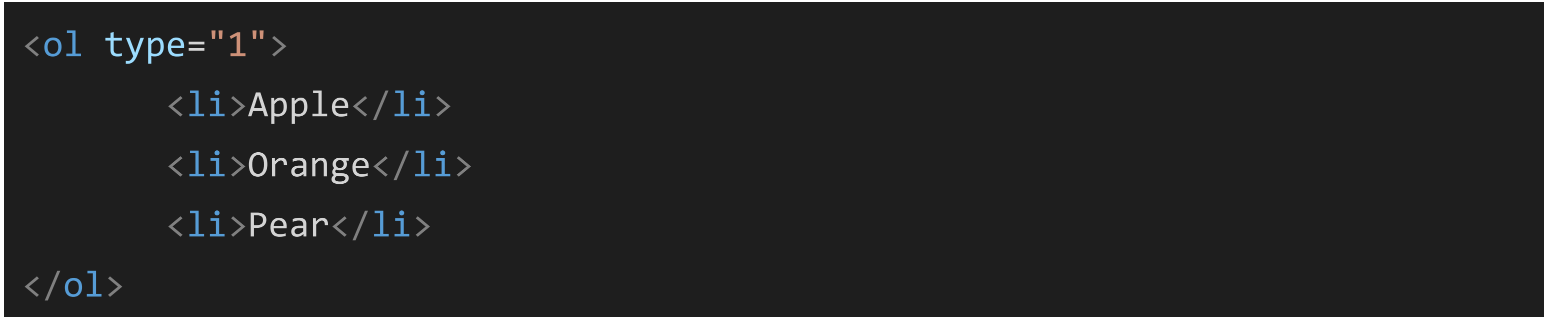
- Denote **headings** in the page
- **Have** implied **semantics**
- There are **6 levels** of headings
- Should appear in **numerical order** inside the content of the page
- A rule of thumb is to never use more than **3 different levels** on the same page



```html
<h1>My Main Heading</h1>
<h2>Secondary Heading 1</h2>
<p>Lots of content ........... </p>
<p>Lots of content ........... </p>
<h2>Secondary Heading 2</h2>
<p>Even more content ........... </p>
```

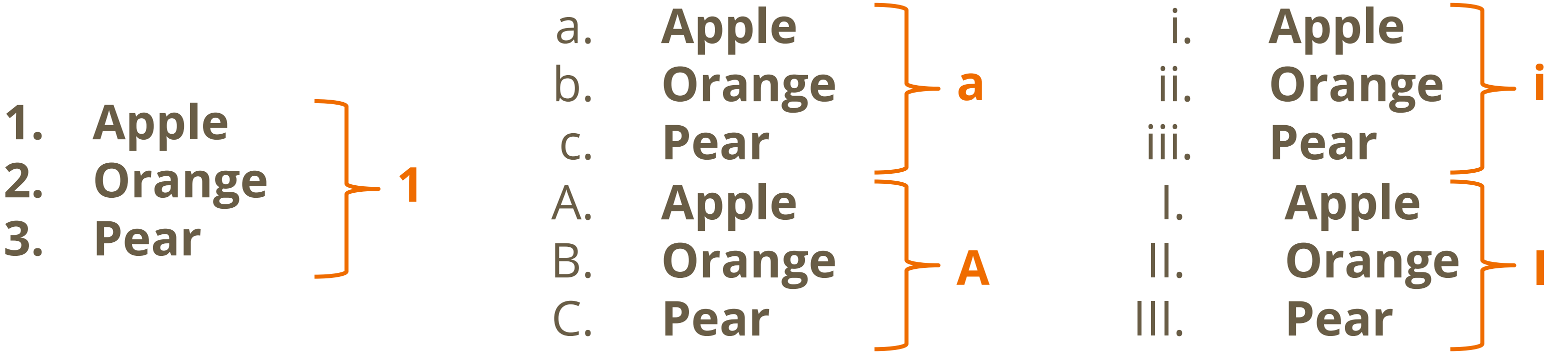
Lists

Ordered Lists: **<ol>**



```
<ol type="1">
        <li>Apple</li>
        <li>Orange</li>
        <li>Pear</li>
</ol>
```

Attribute values for type:



1. **Apple**
2. **Orange**
3. **Pear**

1

a. **Apple**
b. **Orange**
c. **Pear**

a

A. **Apple**
B. **Orange**
C. **Pear**

A

i. **Apple**
ii. **Orange**
iii. **Pear**

i

I. **Apple**
II. **Orange**
III. **Pear**

I

Lists

Unordered Lists: **`<ul>`**

```
<ul type="disc">
        <li>Apple</li>
        <li>Orange</li>
        <li>Pear</li>
</ul>
```

Attribute values for type:

● **Apple**
● **Orange**
● **Pear**

disc

○ **Apple**
○ **Orange**
○ **Pear**

circle

■ **Apple**
■ **Orange**
■ **Pear**

square

Lists

- Description Lists: **`<dl>`**
- Pairs of terms (**`<dt>`**) and associated descriptions (**`<dd>`**)

```
<dl>
    <dt>Firefox</dt>
    <dd>
        The Red Panda also known as the Firefox, is a mostly
        herbivorous mammal 60 cm long.
    </dd>
</dl>
```

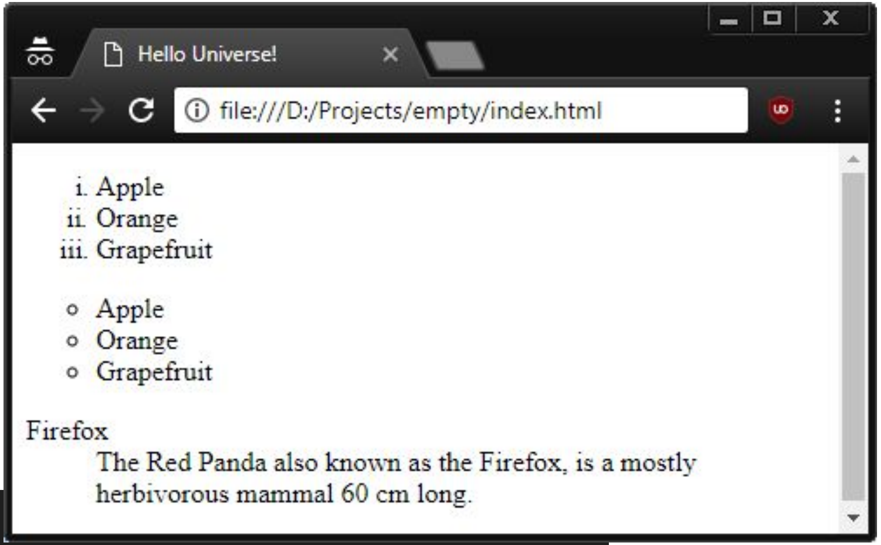
- Renders without bullets
- Description is indented

Lists

```html
<ol type="i">
    <li>Apple</li>
    <li>Orange</li>
    <li>Grapefruit</li>
</ol>

<ul type="circle">
     <li>Apple</li>
     <li>Orange</li>
     <li>Grapefruit</li>
</ul>
```

```html
<dl>
   <dt>Firefox</dt>
   <dd>
       The Red Panda also known
       as the Firefox ...
   </dd>
</dl>
```



```
   i. Apple
  ii. Orange
 iii. Grapefruit

   ○ Apple
   ○ Orange
   ○ Grapefruit

Firefox
      The Red Panda also known as the Firefox, is a mostly
      herbivorous mammal 60 cm long.
```

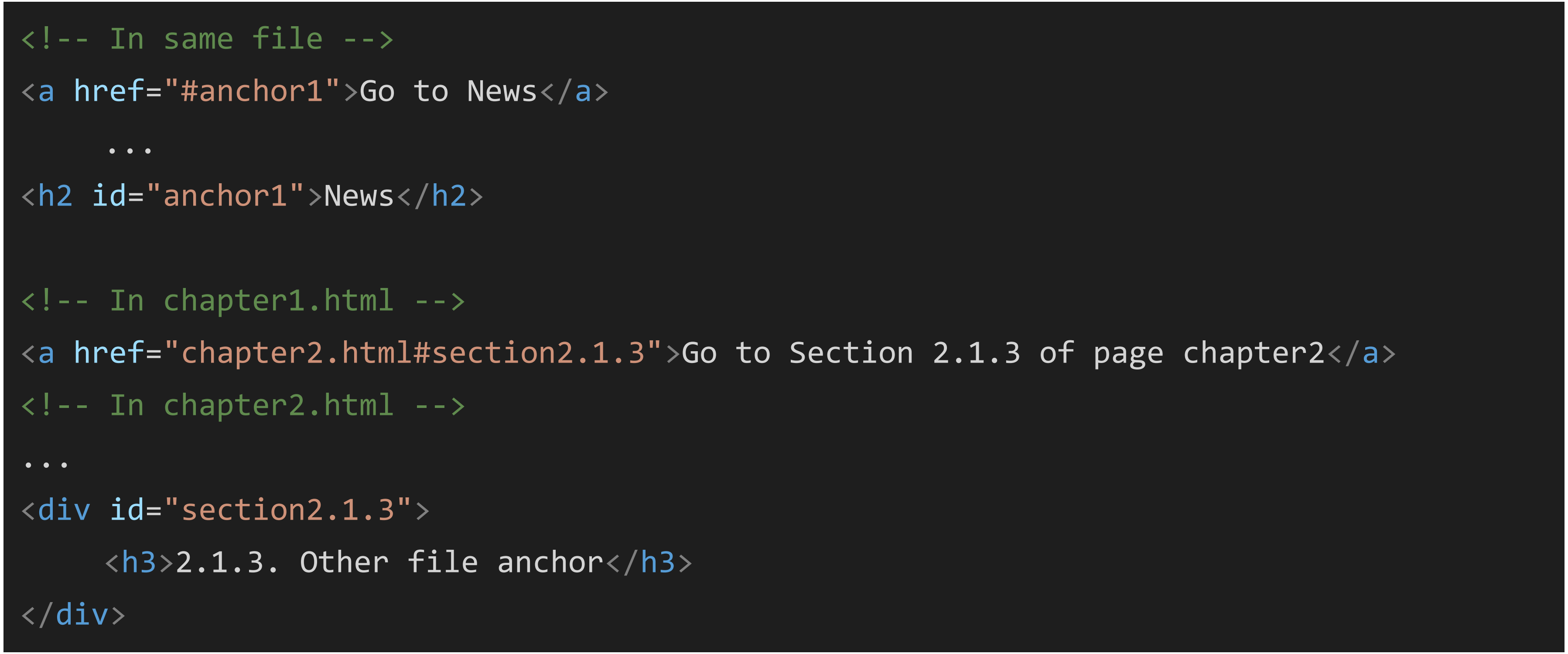
Hyperlinks

```
<a href="about.html">To about.html in the same folder</a>
<a href="../in-parent.html">To in-parent.html in parent folder</a>
<a href="child/in-child.html">To in-child.html in a subfolder</a>

<a href="https://google.com/" title="Google">To other site w/ more information</a>
<a href="https://google.com/" target="_blank">Opens in a new window</a>
<a href="https://google.com/" target="frameName">Opens in an iframe</a>

<a href="mailto:me@example.com">Opens default email app with “to” filled in</a>
<a href="mailto:me@example.com?cc=someone@else.com&subject=Hello World">
    Email link with cc and subject
</a>
```

Hyperlinks



```html
<!-- In same file -->
<a href="#anchor1">Go to News</a>
     ...
<h2 id="anchor1">News</h2>

<!-- In chapter1.html -->
<a href="chapter2.html#section2.1.3">Go to Section 2.1.3 of page chapter2</a>
<!-- In chapter2.html -->
...
<div id="section2.1.3">
     <h3>2.1.3. Other file anchor</h3>
</div>
```

Images

<img> tags are empty elements:

```
<img src="images/logo.png">
<img src="images/logo.png" alt="Website Logo">
```

Attributes:

src	Path to image file (relative or absolute) or base64
alt	Alternate text (e.g. in text mode, image missing)
height	Height in number of pixels
width	Width in number of pixels

iFrames

iFrames (inline frames) are almost like a browser inside your web page, they can be used to display other html pages inside your own page.

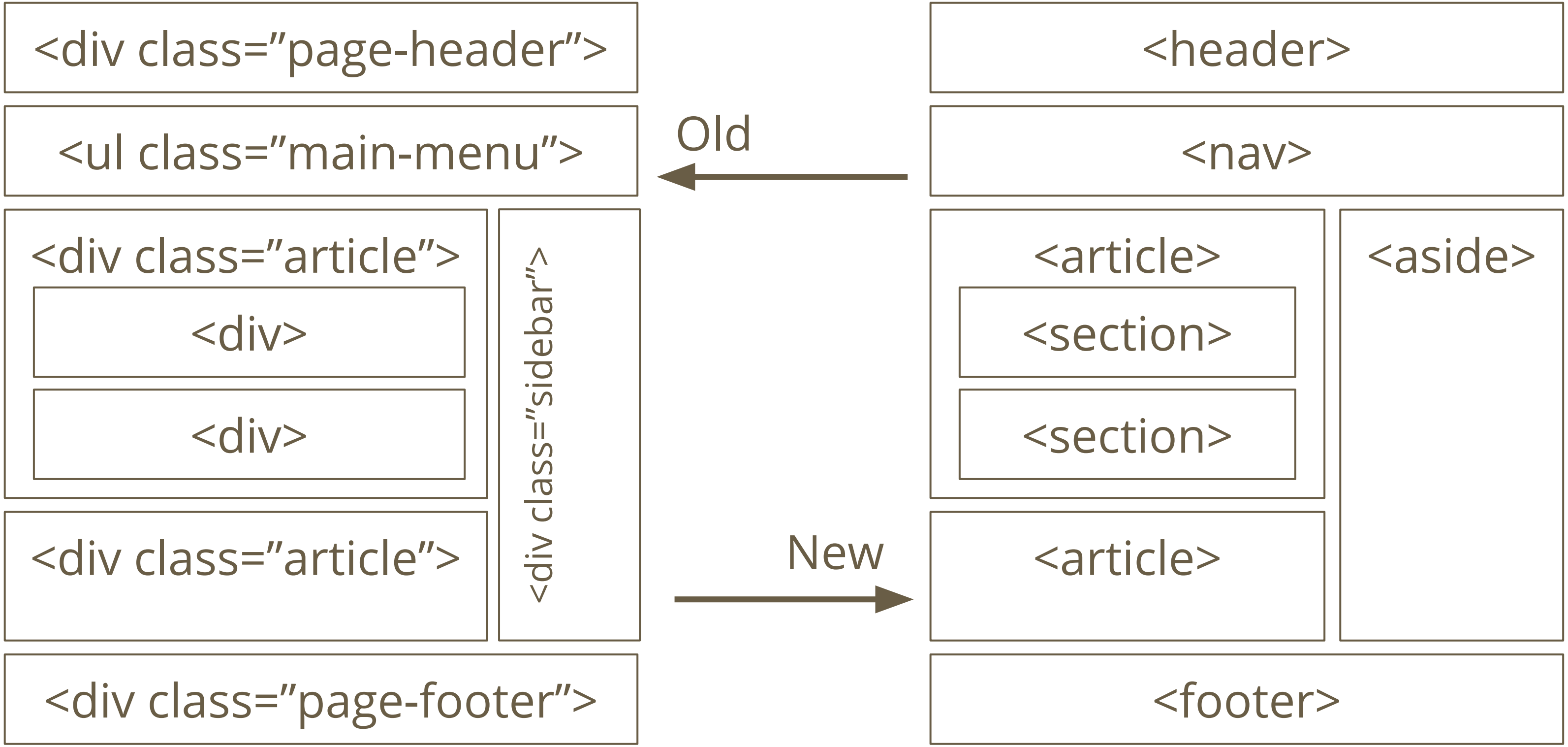
Frames (not iFrames but similar) were very popular in the early ages of web development in order to compose a webpage out of multiple parts, some of which rarely changed. You shouldn’t use frames anymore!

iFrames are still used today in order to display certain types of content inside the main html page, e.g.: ads coming from an ad server, embedded applications (like a YouTube video)

iFrame Example

```html
<iframe
     src="https://mdn-samples.mozilla.org/snippets/html/iframe-simple-contents.html"
     width="400"
     height="300">
    <p>Your browser does not support iframes.</p>
</iframe>
```

Semantic Elements



Tables

Tables

Used **only** to represent **tabular data**, never layout!

A table consists **rows** (<tr>) and **cells/data** (<td>)

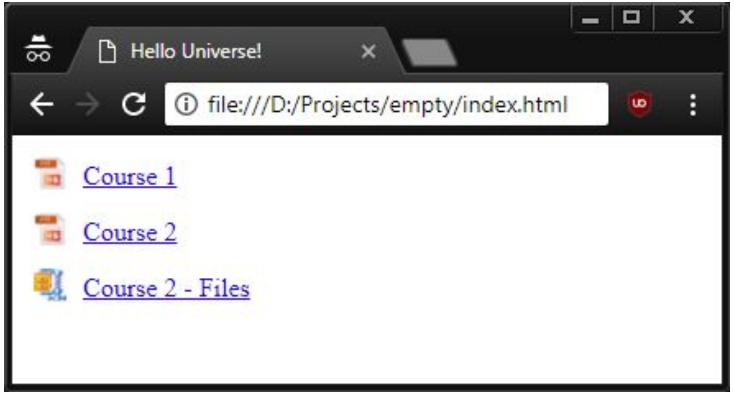
Table elements:

- **<table>** - actual table
- **<thead>, <tbody>, <tfoot>** - table sections
- **<tr>** - table row
- **<th>** - table header
- **<td> -** tabular data (cell)

For layout use CSS positioning, float, display (flex/grid) etc. styles instead!

Simple Tables

```
<table cellspacing="0" cellpadding="5">
      <tr>
            <td><img src="pptx.png"></td>
            <td><a href="course1.pptx">Course 1</a></td>
      </tr>
      <tr>
            <td><img src="pptx.png"></td>
            <td><a href="course2.pptx">Course 2</a></td>
      </tr>
      <tr>
            <td><img src="zip.png"></td>
            <td><a href="course2-files.zip">
                  Course 2 - Files</a></td>
      </tr>
</table>
```



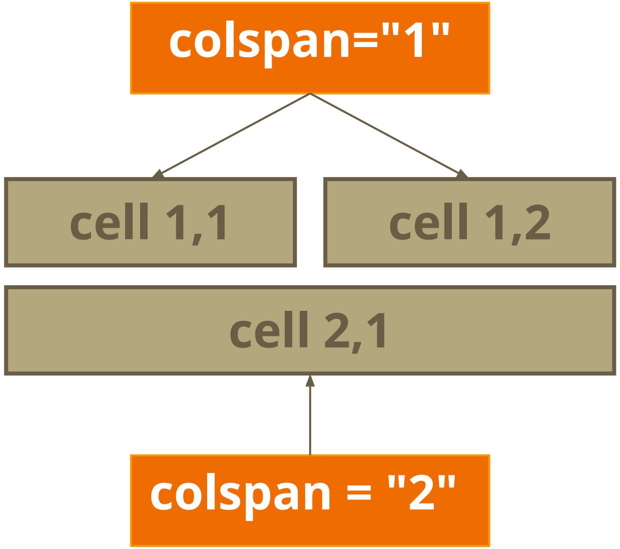
Course 1

Course 2

Course 2 - Files

Cells Spanning Rows and Columns

colspan: how many columns does one cell occupy starting from it’s current position and looking to the right hand side



colspan="1"

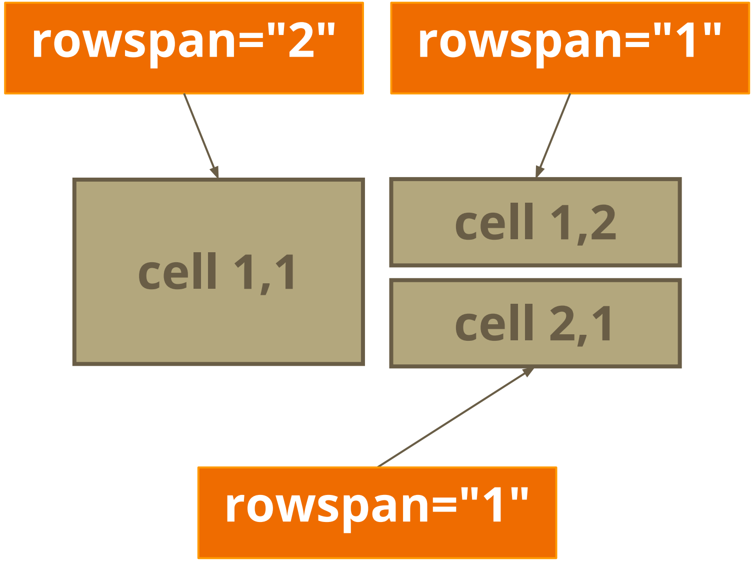
cell 1,1

cell 1,2

cell 2,1

colspan = "2"

rowspan: how many rows does the cell occupy starting from the current position and looking down.



rowspan="2"

rowspan="1"

cell 1,1

cell 1,2

cell 2,1

rowspan="1"

Forms

Forms

The **primary** method for accepting **user input and data**

Data is usually **sent to** a web **server**

JavaScript can also **intercept** it to use it in a **different** way

```html
<form method="get" action="/url/to/send/data/to">
    <!-- Form Elements -->
</form>
```

What Makes a Form?

- **Labels** - annotate input fields for the user
- **Input fields** - text fields, password fields, checkboxes, radio buttons, etc.
- **Actions** - usually buttons that perform an action (e.g. submit, reset)
- **Help** - provide assistance with form submission
- **Messages** - give feedback based on user input and validation
- **Validation** - ensure the data is acceptable

Labels

Labels are used to associate an **explanatory text** to a form field using the field's **ID**

```
<label for="username">Username</label>
<input type="text" id="username" name="username" />
```

Clicking on a label shifts focus to its associated field (checkboxes are toggled, radio buttons are checked)

Labels are both a usability and accessibility feature and are required in order to pass accessibility validation

Input Fields

Single-line text field:

```html
<input type="text" name="fieldName" value="This is a text field" />
```

Multi-line text field:

```html
<textarea name="fieldName">This is a multi-line text field</textarea>
```

Password field – masks entered text

```html
<input type="password" name="password" />
```

Hidden fields contain data not shown to the user, but sent to the server:

```html
<input type="hidden" name="fieldName" value="This is a hidden text field" />
```

- Often used by JavaScript code

Form Controls

Checkboxes:

```
<input type="checkbox" name="fruit" value="apple" />
```

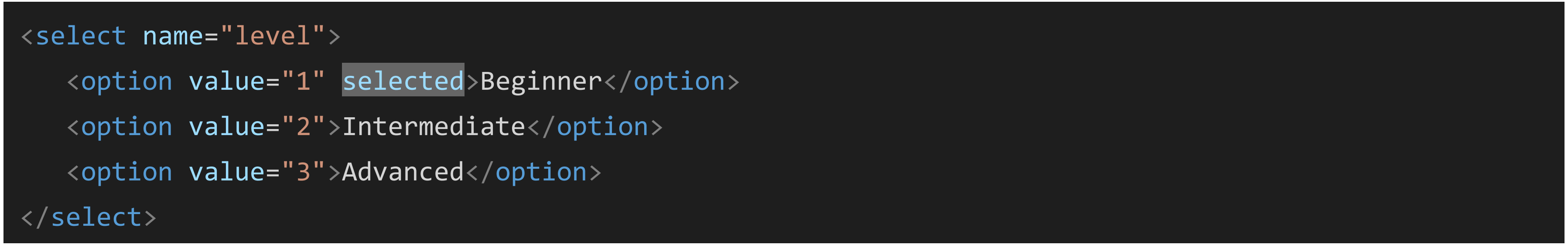
Radio buttons

- Radio buttons can be grouped, allowing only one of the group to be selected at a time, this is achieved by giving all of them the same name

```
<input type="radio" name="city" value="Brasov" />
<input type="radio" name="city" value="Bucuresti" />
<input type="radio" name="city" value="Cluj" />
```

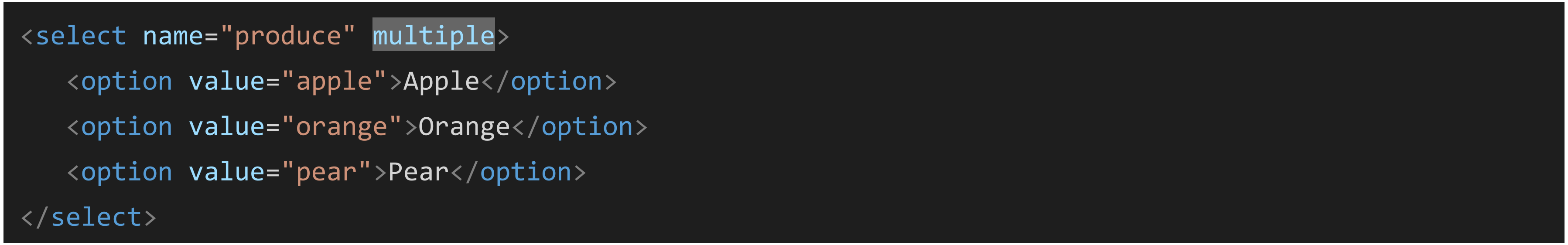
Form Controls

Dropdown menu / Select field



```html
<select name="level">
   <option value="1" selected>Beginner</option>
   <option value="2">Intermediate</option>
   <option value="3">Advanced</option>
</select>
```

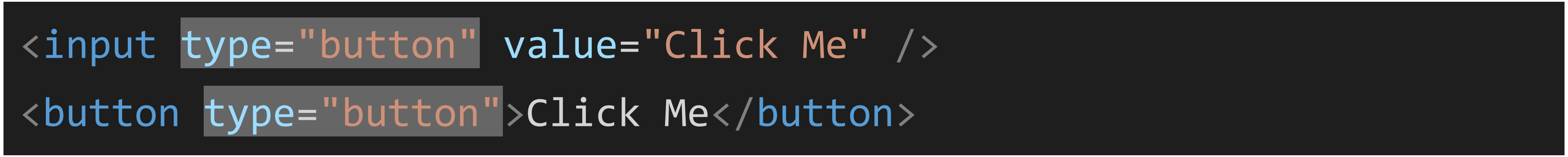
Multiple select field – displays the list of items on multiple lines and allows for more than one to be selected by holding CTRL or SHIFT



```html
<select name="produce" multiple>
   <option value="apple">Apple</option>
   <option value="orange">Orange</option>
   <option value="pear">Pear</option>
</select>
```

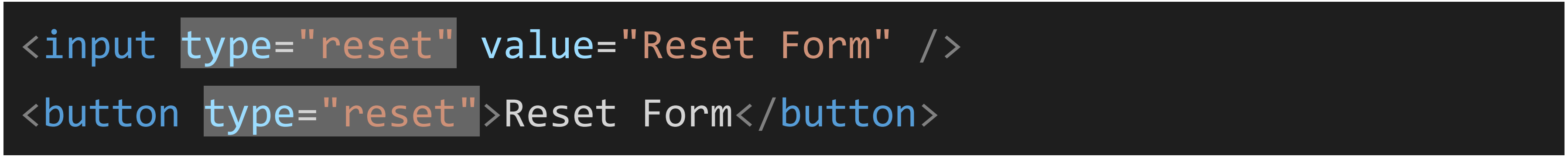
Buttons

- Buttons, used by JavaScript to add functionality, have no default action



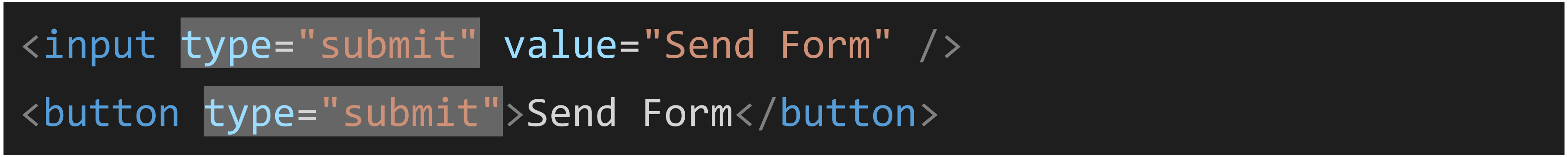
```
<input type="button" value="Click Me" />
<button type="button">Click Me</button>
```

- Reset button – returns the form to its initial state



```
<input type="reset" value="Reset Form" />
<button type="reset">Reset Form</button>
```

- Submit button – sends the form’s content to server

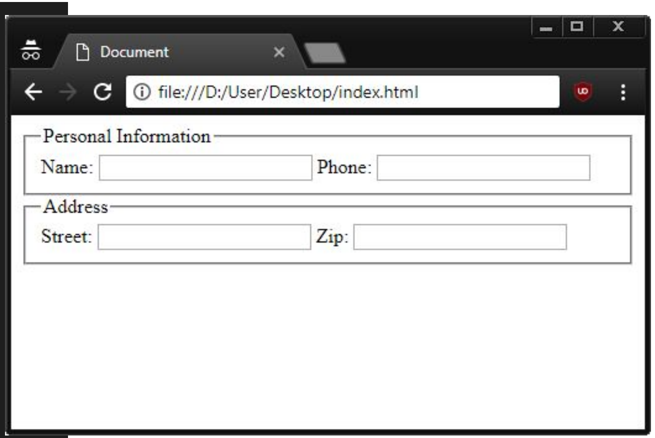


```
<input type="submit" value="Send Form" />
<button type="submit">Send Form</button>
```

Fieldsets

Used to visually and semantically group some form elements together.

```
<form>
   <fieldset>
       <legend>Personal Information</legend>
       <label>Name: <input type="text" name="name" /></label>
       <label>Phone: <input type="tel" name="phone" /></label>
   </fieldset>
   <fieldset>
       <legend>Address</legend>
       <label>Street: <input type="text" name="street" /></label>
       <label>Zip: <input type="text" name="zip"></label>
   </fieldset>
</form>
```



Personal Information

Name: Phone:

Address

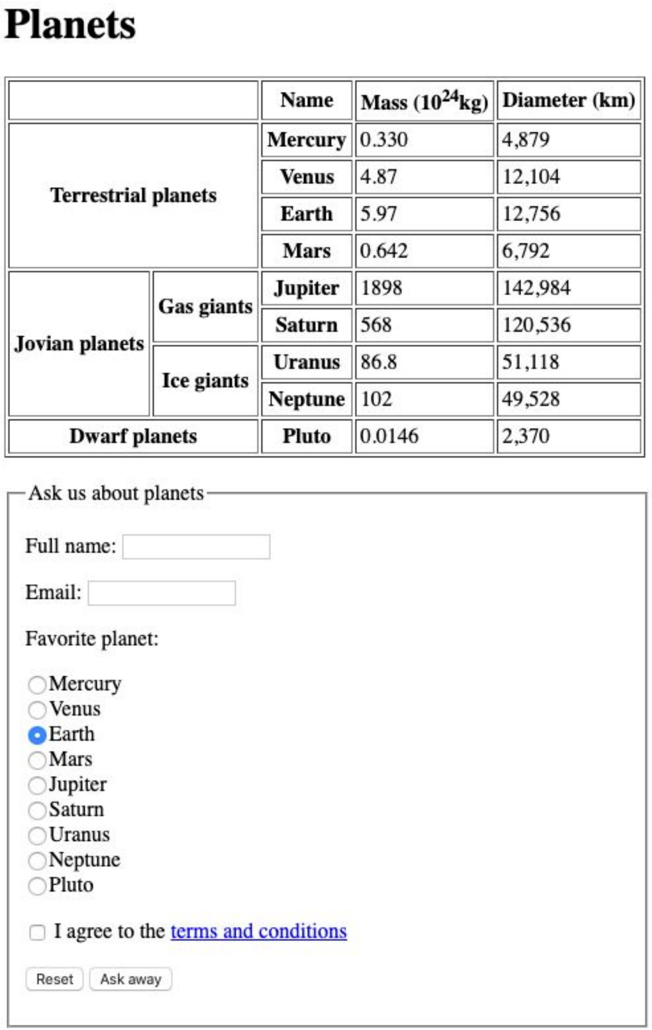
Street: Zip:

Home Assignment

Create an index.html page that looks like the image to the right.

In the form:

- When you click on a label the corresponding input is toggled or focused
- Earth is selected by default
- The reset button actually empties the form
- The “Ask away” button submits the form to the current page but the URL should now contain the data entered in the form, like this: index.html?name=Paul+Negoescu&email=paul.negoescu%40iquestgroup.com&planet=saturn&agree=on
- Name, email and the agreement are required and the email is validated



Planets

		Name	Mass (10^{24}kg)	Diameter (km)
Terrestrial planets		Mercury	0.330	4,879
		Venus	4.87	12,104
		Earth	5.97	12,756
		Mars	0.642	6,792
Jovian planets	Gas giants	Jupiter	1898	142,984
		Saturn	568	120,536
	Ice giants	Uranus	86.8	51,118
		Neptune	102	49,528
Dwarf planets		Pluto	0.0146	2,370

Ask us about planets

Full name:

Email:

Favorite planet:

○ Mercury
○ Venus
● Earth
○ Mars
○ Jupiter
○ Saturn
○ Uranus
○ Neptune
○ Pluto

□ I agree to the terms and conditions

Resources

MDN - Introduction to HTML

Smashing Magazine - Extensive Guide to Web Form Usability