# HTML

Introduction to HTML

# Course Content

Its highly recommended that you try the code yourself, we've provided a zip file of all the source code and exercises.  You can copy and paste them into your editor to try the code out.  Please feel free to adjust and try it out, make changes and get a feel for how HTML works.

Editor used within the course is brackets.io (http://brackets.io/) - its open source free download.  Adobe product with great features for coding.

This PDF is part of the Introduction to HTML Complete Beginner to Expert Course  @
https://www.udemy.com/course/1926012

# Editor

An HTML editor is a program for editing HTML, the markup of a webpage. ... Many word processing, graphic design and page layout programs that are not dedicated to web design, such as Microsoft Word or notepad, also have the ability to function as HTML editors.

In the course we use Brackets

http://brackets.io/

# Brackets

Brackets is a lightweight, yet powerful, modern text editor.

Get a real-time connection to your browser. Make changes to CSS and HTML and you'll instantly see those changes on screen.

# Try it

1. Open Editor
2. Type some text
3. Save the file as index.html
4. Open in browser
5. Live preview in Brackets

# First Website Ever

http://info.cern.ch/

HTML is backwards compatible

No styling which is introduced later

Purpose was to add some structure to plain text.

# index.html

Why we use index.html

Originally the idea was that navigating to a particular directory would simply list the files in it, just like clicking on a folder on your desktop, but if you preferred you could provide your own custom index of the files.
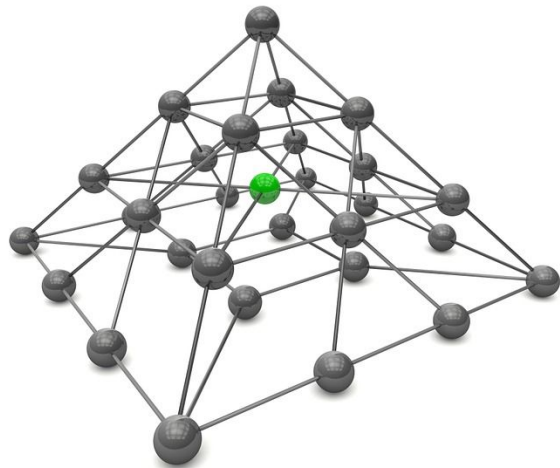
# Build Structure

An Hyper Text Markup Language element is an individual component of an HTML document or web page, once this has been parsed into the Document Object Model.

HTML is composed of a tree of HTML nodes, such as text nodes. Each node can have HTML attributes specified.

Nodes can also have content, including other nodes and text.

Many HTML nodes represent semantics, or meaning. For example, the <title> node represents the title of the document.

# Elements

There are a lot of elements but you may only need to know about 15 that are commonly used.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element

# What is an Element

Tag that wraps text and lets the browser know what to do with it.



Element

Start tag

Content

End tag

`<p class="foo">This is a paragraph.</p>`

Attribute name

value
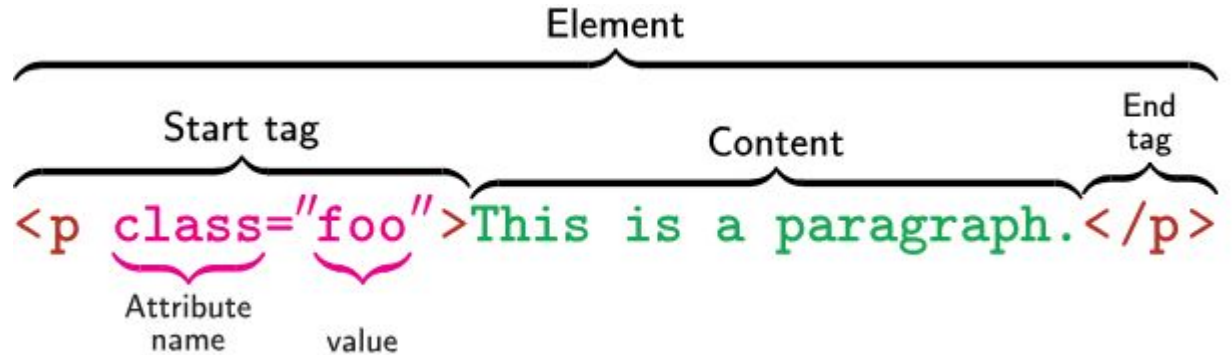
# Exercise #1

```
1  I need a paragraph here.
2  I need bolded text here.
3  New paragraph.
4  This is just a bunch of words, should all
   be in the same paragraph.  This is more
   content in the same paragraph.
```

I need a paragraph here. I need bolded text here. New paragraph.
This is just a bunch of words, should all be in the same paragraph.
This is more content in the same paragraph.

1.  Open exercise 1 start.html
2.  Update the tags to make it look correct using <p> and <b>

# Exercise #1 - solution

```
1  <p>I need a paragraph here.</p>
2  <b>I need bolded text here.</b>
3  <p>New paragraph.</p>
4  <p>This is just a bunch of words, should
   all be in the same paragraph.  This is more
   content in the same paragraph.</p>
```

I need a paragraph here.

**I need bolded text here.**

New paragraph.

This is just a bunch of words, should all be in the same paragraph. This is more content in the same paragraph.

# Basic HTML structure

**&lt;html&gt;...&lt;/html&gt;**

The root element of an HTML document; all other elements are contained in this.  The HTML element delimits the beginning and the end of an HTML document.

**&lt;head&gt;...&lt;/head&gt;**

Container for processing information and metadata for an HTML document.

**&lt;body&gt;...&lt;/body&gt;**

Container for the displayable content of an HTML document.

# Document metadata

<meta>      The HTML <meta> element represents metadata that cannot be represented by other HTML meta-related elements, like <base>, <link>, <script>, <style> or <title>.

<style>      The HTML <style> element contains style information for a document, or part of a document.

<title>      The HTML Title element (<title>) defines the document's title that is shown in a browser's title bar or a page's tab.

<link>      For external files, mainly for styling.

# DocType

A document type declaration, or DOCTYPE, is an instruction that associates a particular webpage with a document type definition (DTD)

HTML5 uses a DOCTYPE declaration which is very short, due to its lack of references to a DTD in the form of a URL or FPI. All it contains is the tag name of the root element of the document, HTML.[4] In the words of the specification draft itself:

<!DOCTYPE html>, case-insensitively.

# Try it

1. Open Editor
2. Create basic structure
3. Save the file as index.html
4. Open in browser
5. Live preview in Brackets
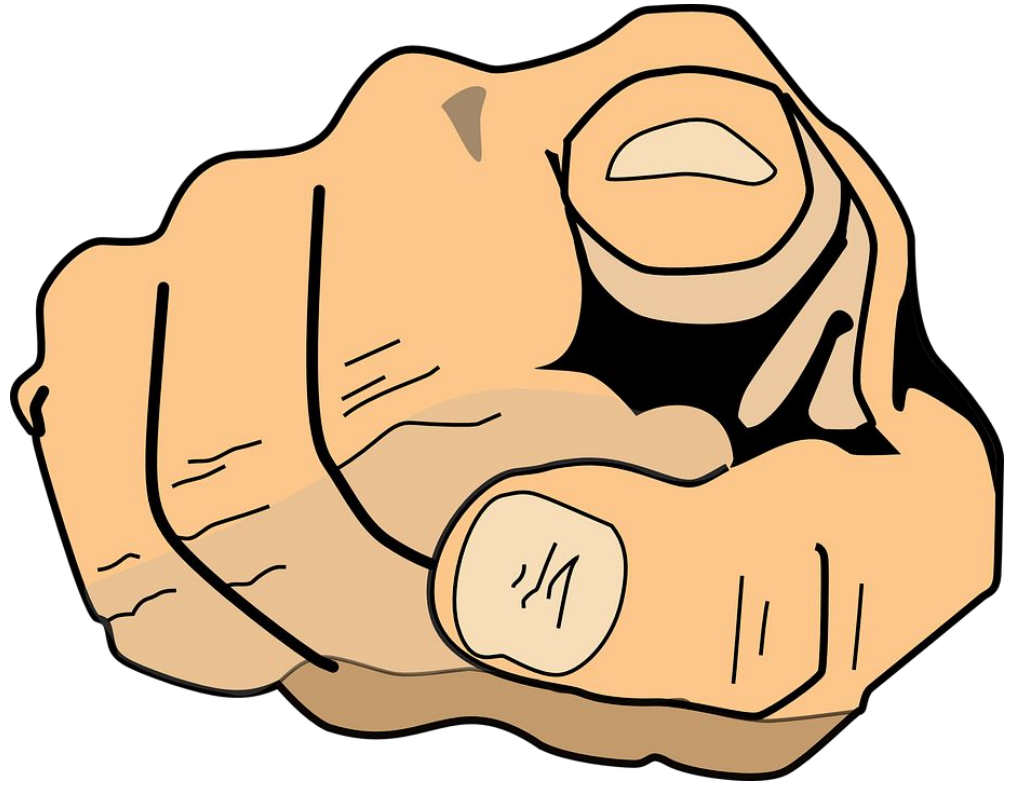
# Solution #1

You should have something like this. Notice the output, only Hello World. example 1.html

Why do you think this is?

```
1 ▼ <html>
2
3 ▼     <head>
4
5             <title>WebPage</title>
6
7       </head>
8
9 ▼     <body>
10
11            Hello World
12
13       </body>
14  |
15  </html>
```

# Creating an HTML boilerplate template

DocType -
https://en.wikipedia.org/wiki/Document_type_declaration

Title tag

example 2.html

```
example 2.html
 1    <!DOCTYPE html>
 2 ▼  <html>
 3
 4 ▼  <head>
 5        <title>My Title</title>
 6    </head>
 7
 8    <body> My Content </body>
 9
10    </html>
```
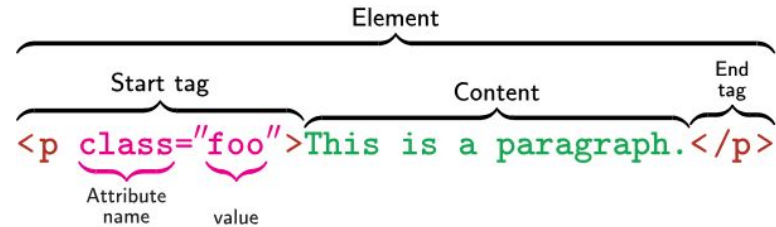
# Element formatting

Tags <tag>

Elements

Attributes <tag id="value"></tag> Add additional information

# Online Resources

HTML (Hypertext Markup Language) is not a programming language; it is a markup language used to tell your browser how to structure the web pages you visit.

```
1   My cat is very grumpy
```

If we wanted the line to stand by itself, we could specify that it is a paragraph by enclosing it in a paragraph (`<p>`) element:

```
1   <p>My cat is very grumpy</p>
```

Opening tag       Closing tag

`<p>`My cat is very grumpy`</p>`

Content
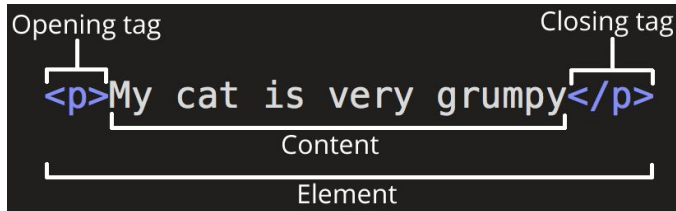
Element

# The main parts of our element are:

**The opening tag:** This consists of the name of the element (in this case, p), wrapped in opening and closing angle brackets. This states where the element begins or starts to take effect — in this case where the start of the paragraph is.

**The closing tag:** This is the same as the opening tag, except that it includes a forward slash before the element name. This states where the element ends — in this case where the end of the paragraph is. Failing to include a closing tag is a common beginner error and can lead to strange results.

**The content:** This is the content of the element, which in this case is just text.

**The element:** The opening tag plus the closing tag plus the content equals the element.

# Headings

<h1>...</h1> <h2>...</h2> <h3>...</h3>
<h4>...</h4> <h5>...</h5> <h6>...</h6>

example 3.html

Section headings at different levels. h1 delimits the highest-level heading, h2 the next level down (sub-section), h3 for a level below that, and so on to h6.

# Self closing tags

Line Breaks <br> and <hr> are self closing tags

Previously <br /> now the slash is not needed.

# Exercise #2

```
exercise 2 start.html
1    First paragraph text
2
3    Second paragraph text
4
5    Third paragraph text
6    on a new line using line break
```

First paragraph text

Second paragraph text

Third paragraph text
on a new line using line break

1. Open exercise 2 start.html
2. Update the tags to make it look correct using <p> and <br>

# Exercise #2 solution

```
<p>First paragraph text</p>

<p>Second paragraph text</p>

<p>Third paragraph text <br> on a
new line using line break</p>
```

1. How is whitespace handled in the editor?
2. What happens when you add more spaces between words?

# Comments

<!-- A Comment → Comments can appear anywhere in a document, as the HTML parser is supposed to ignore them no matter where they appear so long as they are not inside other HTML tag structures.  Comments are represented in HTML and XML as content between '<!--' and '-->'

example 4.html

Unlike most HTML tags, comments do not nest.

```
this is a comment
<!-- comment content -->
```

```
this is a comment
<!-- comment content -->

this is NOT a comment
<p <!-- comment content -->></p>
```

# Formatted Text

Whitespace handling

HTML code and viewable content Whitespace spacing in content

Preformatted text <pre> </pre>

# HTML Entities

common character entity used in HTML is the non-breaking space: ** **

[https://dev.w3.org/html5/html-author/charref](https://dev.w3.org/html5/html-author/charref)

**<p>What does this say &lt;p&gt;, &lt;br&gt;. </p>**

**&copy;**

# HTML Text Formatting

<b> - Bold text → <strong> - Important text
<i> - Italic text → <em> - Emphasized text
<mark> - Marked text
<small> - Small text
<del> - Deleted text
<ins> - Inserted text
<sub> - Subscript text
<sup> - Superscript text

# HTML Text Formatting - Little More

<mark>Marked</mark>
<blockquote>Welcome to blockquotes</blockquote>
<u>underlined text</u>
<strike>Strike text</strike>
<big>Big text</big>
 <acronym> and </acronym>
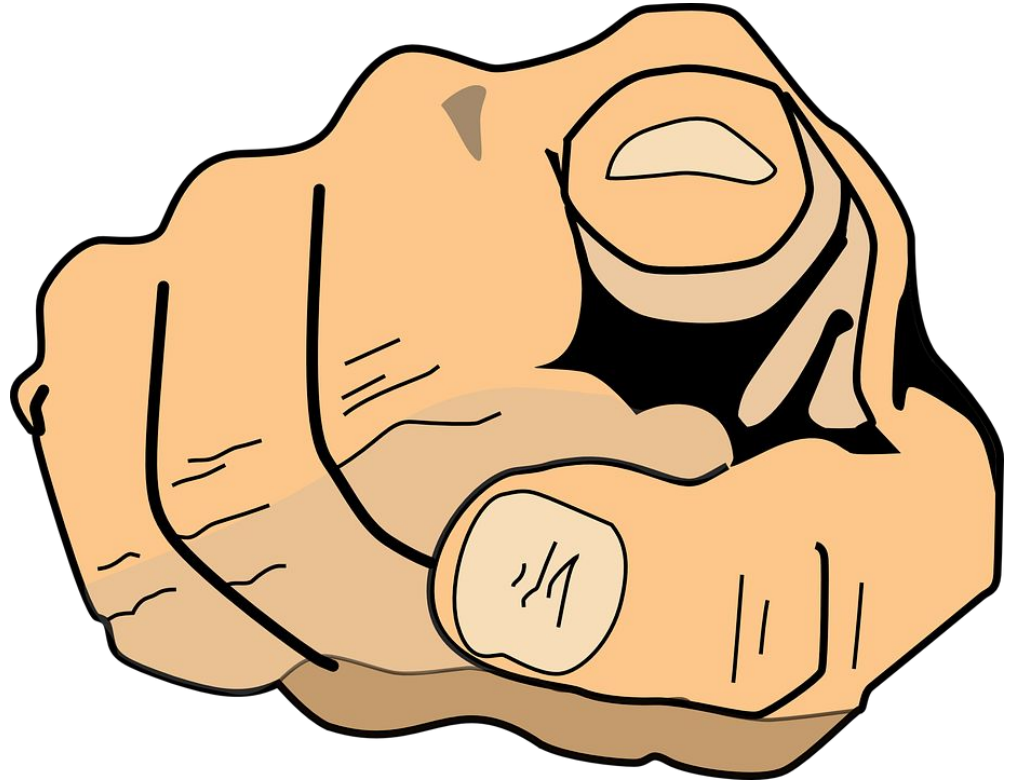 <q>Quoted</q>
 <cite> text within cites </cite>
<code>This is code.</code>

# Try it

1. Open Editor
2. Write some text
3. Apply formatting
4. Add paragraphs and line breaks
5. Create some text add bold and italic to at least one word in your text.

# Group and organize content

The HTML Content Division element (<div>) is the generic container for flow content. It has no effect on the content or layout until styled using CSS.

The <div> element should be used only when no other semantic element (such as <article> or <nav>) is appropriate.

The HTML <span> element is a generic inline container for phrasing content, which does not inherently represent anything.

Both can be used to group elements for styling purposes, common attributes.

<div> is a block-level element whereas a <span> is an inline element.
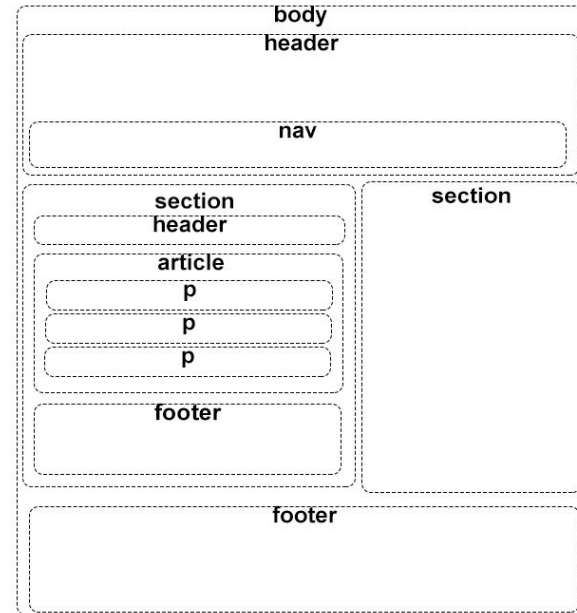
# Semantic - elements with a meaning HTML5

https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Using_HTML_sections_and_outlines

- <article>
- <aside>
- <footer>
- <header>
- <nav>
- <section>

Previously done with divs and adding attributes to the elements.
These are just like divs with meaningful names, they don't do anything for the styling.

# Exercise #3



1. Open exercise 3 start.html
2. Apply content sectioning, head info and all we've learned in the earlier lessons.

# Lorem Ipsum

Lorem Ipsum

Lorem ipsum is placeholder text commonly used in the graphic, print, and publishing industries for previewing layouts and visual mockups

https://htmlstrip.com/lorem-ipsum-generator

```
<!-- Main content area -->
<h1>Quid de Platone aut de Democrito loquar?</h1>
<p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Occultum facinus esse potuerit,
gaudebit; Duo Reges: constructio interrete. <b>An
eiusdem modi?</b> Sed ego in hoc resisto; Eam
stabilem appellas. <b>At hoc in eo M.</b> Tria
genera bonorum; Sed fortuna fortis; </p>
<p>Odium autem et invidiam facile vitabis. Sedulo,
inquam, faciam. Quis istud possit, inquit, negare?
Certe, nisi voluptatem tanti aestimaretis. </p>
<p>Second paragraph text </p>
<p>Third paragraph text
    <br> on a new line using line break </p>
section>
```

# HyperLinks Anchor

The HTML <a> element (or anchor element) creates a hyperlink to other web pages, files, locations within the same page, email addresses, or any other URL.

Target _self default opens in same window _blank - New window or tab _parent opens in parent frame

Link Bookmarks <a href="id">id</a>  <h2 id="id">blah  blah blah</h2>

example 5.html

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a

# Anchor Attributes

**_self:** Load the URL into the same browsing context as the current one. This is the default behavior.

**_blank:** Load the URL into a new browsing context. This is usually a tab, but users can configure browsers to use new windows instead.

**_parent:** Load the URL into the parent browsing context of the current one. If there is no parent, this behaves the same way as _self.

**_top:** Load the URL into the top-level browsing context

example 6.html

```
header>
    <h1>My Heading</h1>
    <nav> <a href="#output1">Home Page</a> <a
    href="https://www.google.com"
    target="_blank">Website</a> <a
    href="mailto:spamaddress@example.com">Email</a> <a
    href="tel:+123456789">Phone</a> </nav>
/header>
```

# Absolute and relative paths

An absolute or full path points to the same location in a file system, regardless of the current working directory.

By contrast, a relative path starts from some given working directory, avoiding the need to provide the full absolute path.
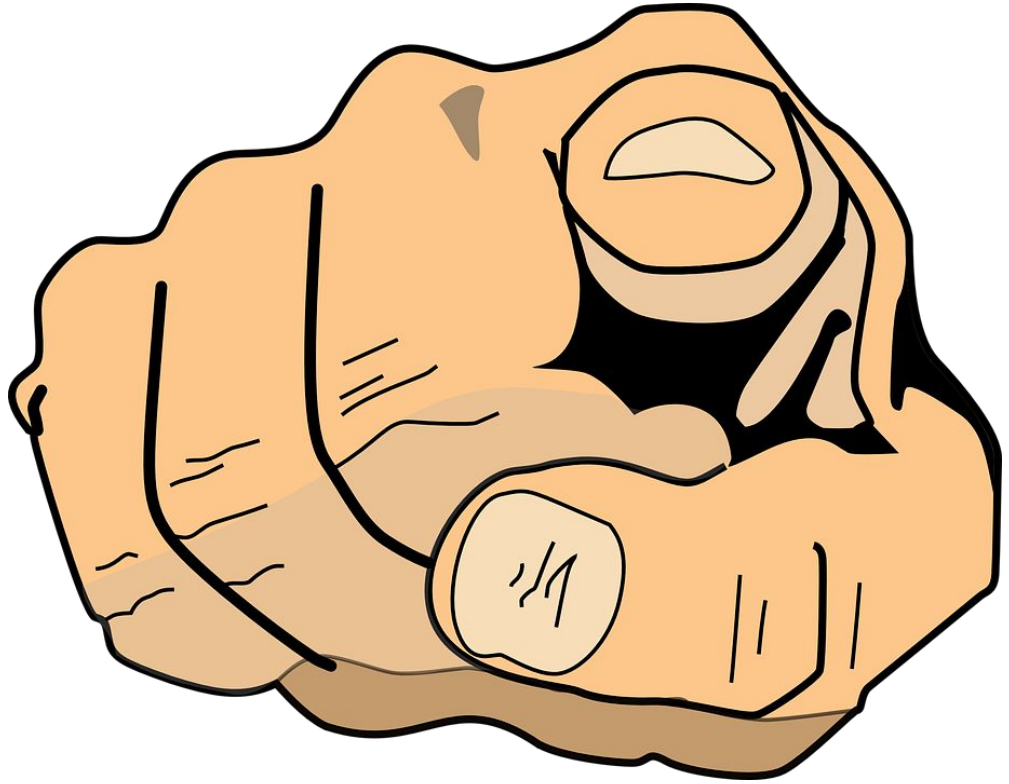
example 7.html

```html
<a href="example%201.html">Example 1</a>
<a href="#output1">Home Page</a>
<a href="https://www.google.com"
target="_blank">Website</a>
```

# Try it

1. Open Editor
2. Add a link to another part of your webpage using ID attribute.
3. Add new hyperlink to the page in the nav area, link to google or your favorite website

*Add more text so that the page has a scroll to see it move down.  Add ID to be able to select the grouped off content.

# Images

Placeholder images - https://placeholder.com/

`<img src="https://via.placeholder.com/350x150">`

# Exercise #4



1. Add 2 images one with a relative path the other with an absolute path.
2. Add a hyperlink to click out to the image with the absolute path.

# Lists

Lists unordered <ul><li>
Lists ordered <ol><li>
Nested Lists inside lists
<li><ul><li></li></ul></li>
List styles
<ul style="list-style-type:disc">
<ol type="1">
List options <ol start="5">
Description Lists <dl><dt><dd>

```
My Heading</h1>
>
<ul>
    <li><a href="example%201.html">Example 1</a>
    <li><a href="#output1">Home Page</a></li>
    <li><a href="https://www.google.com"
    target="_blank">Website</a> </li>
    <li> <a
    href="mailto:spamaddress@example.com">Email<
    </li>
    <li> <a href="tel:+123456789">Phone</a> </li
</ul>
```

## My Heading

- Example 1
- Home Page
- Website
- Email
- Phone

## Quid de Platone aut d loquar?

# Exercise #5

```
<h1>My Heading</h1>
<nav>
    <a href="example%201.html">Example 1</a>
    <a href="#output1">Home Page</a>
    <a href="https://www.google.com"
target="_blank">Website</a>


    <a href="mailto:spamaddress@example.com">Email</a> <a
href="tel:+123456789">Phone</a> </nav>
```

1. Select all the hyperlinks in the nav section
2. Create an underordered list
3. exercise 5 start.html

# Exercise #6

## http://info.cern.ch - home of the first website

From here you can:

- Browse the first website
- Browse the first website using the line-mode browser simulator
- Learn about the birth of the web
- Learn about CERN, the physics laboratory where the web was born

1. Use the start file to build your own copy of the first website ever
2. Apply lists, and hyperlinks, and heading
3. Use exercise 6 start.html

# Nested Lists

Lists allow you to create lists inside of list.

```
<ul>
    <li>HTML website</li>
    <li>Tag structure
        <ul>
            <li>
                <h2>My Website</h2>
                <ol>
                    <li>DOCTYPE</li>
                    <li>HTML</li>
                    <li>Head
                        <ol>
                            <li>H1</li>
                            <li>Paragraph</li>
                            <li>Paragraph</li>
                            <li>Paragraph</li>
                        </ol>
                    </li>
                </ol>
            </li>
        </ul>
    </li>
</ul>
```

# Exercise #7

```
<!--<!DOCTYPE html>
<html>

<head>
    <title>My Title</title>
</head>

<body>
    <h1>My Content</h1>
    <p>First paragraph text</p>
    <p>Second paragraph text</p>
    <p>Third paragraph text
        <br> on a new line using line b
    <br>next line
    <hr>new line with horizontal rule b
    comment </body>

</html>-->
```

- HTML website
- Tag structure

  ○ **My Website**

     1. DOCTYPE
     2. HTML
     3. Head
          1. H1
          2. Paragraph
          3. Paragraph
          4. Paragraph

1. Take a typical website structure from before
2. Represent it in a nested list format using lists in html
3. example 10.html
4. Use exercise 7 start.html

# Tables

The HTML <table> element represents tabular data — that is, information presented in a two-dimensional table comprised of rows and columns of cells containing data.

<table><tr><td>
<th>
<thead><tbody>

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table

**The table header**

| First Name | Last Name | Favorite Fruit |
|---|---|---|
| Laurence | Svekis | Apple |

The table footer

# Tables with Style

With HTML5 we use style instead of the table attributes.
example 8.html

```html
<thead>
    <tr>
        <th colspan="3">The table header</th>
    </tr>
</thead>
<tbody style="border: 1px solid black">
    <tr>
        <td style="width: 100px;border: 1px solid black">First
        Name</td>
        <td style="width: 100px;border: 1px solid black">Last
        Name</td>
        <td style="width: 100px;border: 1px solid
        black">Favorite Fruit</td>
    </tr>
    <tr>
        <td>Laurence</td>
        <td>Svekis</td>
        <td>Apple</td>
    </tr>
```

# HTML Forms

The HTML <form> element represents a document section that contains interactive controls for submitting information to a web server.

Create a simple form using input and form tags.

example 9.html

```
<form> First name:
    <br>
    <input type="text"
    name="firstname" value="Laurence">
    <br> Last name:
    <br>
    <input type="text" name="lastname"
    value="Svekis">
    <br>
    <br>
    <input type="submit"
    value="Submit"> </form>
```

First name:
Laurence
Last name:
Svekis

Submit

# Forms action and method

- The `action` attribute defines the location (URL) where the form's collected data should be sent when it is submitted.
- The `method` attribute defines which HTTP method to send the data with (it can be "get" or "post").

https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form

Name:

E-mail:

Message:

Send your message

# Forms - textarea - labels - placeholder

https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form

```html
<form action="/serverPage.php" method="get">
  <div>
    <label for="name">Name:</label>
    <input type="text" id="name" name="user_name">
  </div>
  <div>
    <label for="mail">E-mail:</label>
    <input type="email" id="mail" name="user_mail">
  </div>
  <div>
    <label for="msg">Message:</label>
    <textarea id="msg" name="user_message"></textarea>
  </div>
</form>
```

# Try it

1. Take simple form and instead of line breaks use labels
2. Add method and attributes and placeholder

example11.html

Name: `User Name`
E-mail: `Email`
Message: `my message`

# Forms input types

Button, checkbox, color, date, datetime-local, email, file, hidden, image, month, number, password

example12.html

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

```html
<form>
  <input type="color"><br>
  <input type="date"><br>
  <input type="email"><br>
  <input type="number" min="10" max="50"><br>
  <input type="url"><br>
<input type="file"><br>
<input type="range" min="10" max="50"><br>
<input type="datetime-local"><br>
<input type="submit">
</form>
```
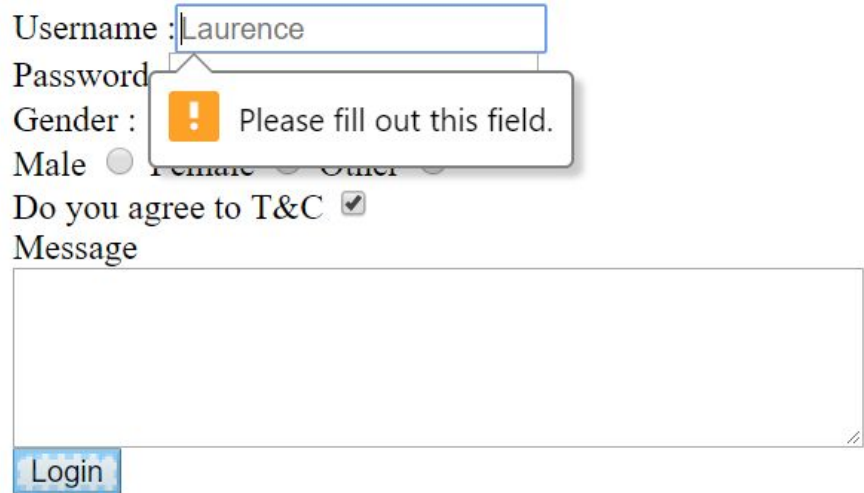
# Forms Advanced validations & constraints

required attribute — if you want to make an input mandatory

All text fields created by (<input> or <textarea>) can be constrained in size using the minlength and maxlength attributes.

example14.html

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

# Try it

1. Try some more form elements

example13.html

Subject : Support ▼
```
Text that the user adds
```
submit

Username : Laurence
Password : ••••••
Gender :
Male ⦿ Female ○ Other ○
Do you agree to T&C ☑
Login

# Exercise #8

Name :

First name | Last name

Contact :

Mobile number or email

Birthday :

May, 2018

Gender :

Male ○ Female ○ Other ●

Submit

1. Let's create a typical signup form.
2. Add some styling for the fields ;)
3. Solution is in exercise 8 done.html

# FieldSet for grouping

The HTML <fieldset> element is used to group several controls as well as labels (<label>) within a web form.

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/fieldset

example 16.html

# Exercise #9

┌─ Like HTML ─────────────────────────┐
│  ⚪ Yes  ⚪ No                        │
└─────────────────────────────────────┘

How old are you? [_____]

What's your favorite fruit? [_____]

What's your e-mail? [_____]

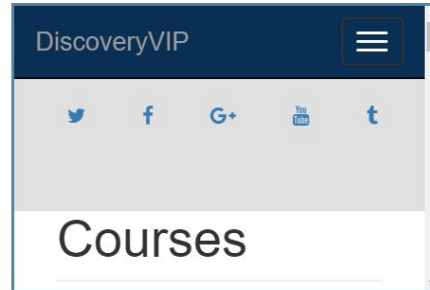Leave a short message [_____]

[ Submit ]

1. Build the above form
2. Add validation
3. example 15.html

# iFrames

The HTML Inline Frame element (<iframe>) represents a nested browsing context, effectively embedding another HTML page into the current page. You can include any number of <iframe> elements within a document, each of which embeds another document inside <body> of a page. example17.html

https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe

```html
<iframe id="inlineFrameExample"
    title="Inline Frame Example"
    width="300"
    height="200"
    src="http://www.discoveryvip.com">
</iframe>
```
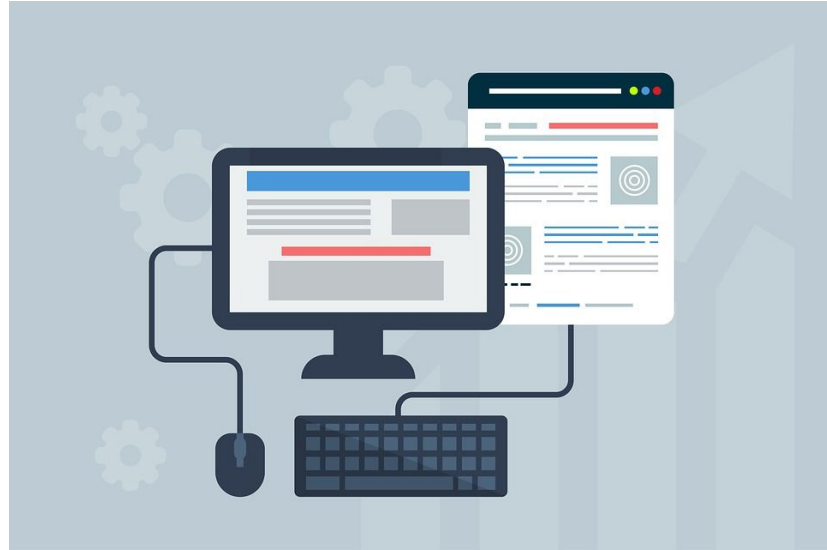
# Exercise #10

1. Build a mini website with at least 3 pages (home) (about) (contact)
2. Link them together

You can add images, use a form for the contact, use list for the about.  Use table on the home page.

Use the placeholder generators for each content.

# Thank you

Thank you for taking the course, and reading this PDF. If you have any questions of suggestions please connect with me on Udemy.

https://www.udemy.com/user/lars51/

Laurence Svekis