

[Lesson 1]

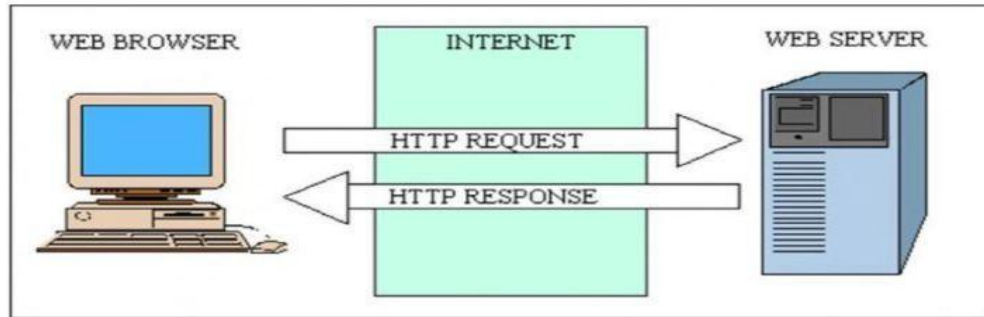
Roi Yehoshua 2018

[Our targets for today]

- Get familiar with Front-end basics
- Learn how HTML and CSS work
- Learn about JavaScript - what it is and what it is used for
- Learn basic HTML tags
- Create a simple web page

[Basic Web Architecture]

- **World Wide Web (WWW)** – an information space where documents and other web resources and can be accessed via the Internet
- **Web browser** – a software application for retrieving, presenting and traversing information resources on the World Wide Web, such as web pages, images, etc.
- **Web server** – a server software that can serve contents to the World Wide Web



[HyperText Transfer Protocol (HTTP)]

- **Hypertext** is structured text that uses hyperlinks between documents containing text
- HTTP is the protocol to exchange or transfer hypertext
 - **Simple** - has a Request and Response (header & body)
 - **Stateless** - each request is independent from the others



HTTP Request

```
GET /folder/page.html HTTP/1.0  
User-agent: Mozilla/4.0
```



HTTP Response

```
HTTP/1.0 200 OK  
Server: Microsoft-IIS/5.0 Content-Length: 6821  
Content-Type: text/html  
data data data ...
```

[Uniform Resource Locator (URL)]

- Specifies the location of a web resource on a computer network
- The structure of a URL:

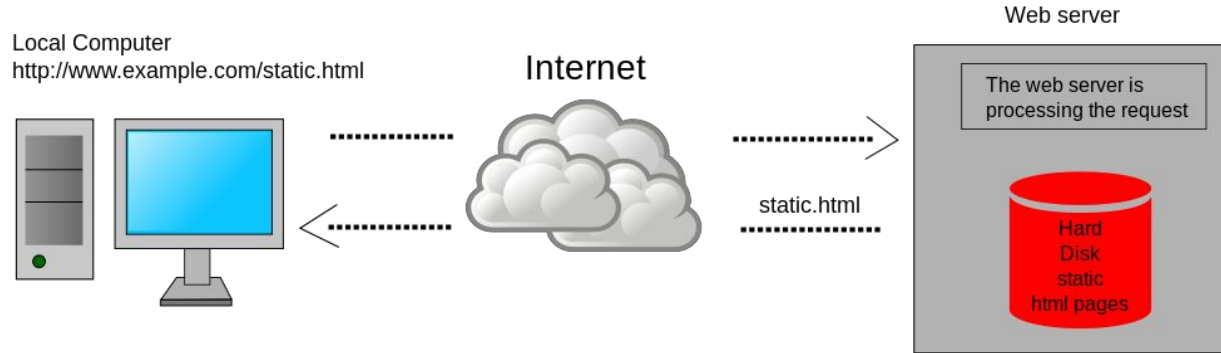
1 2 3 4 5 6 7 8
`https://www.example.com:3000/path/resource?id=123#section-id`

- 1 Scheme - defines how the resource will be obtained.
- 2 Subdomain - www is most common but not required.
- 3 Domain - unique value within its top-level domain.
- 4 Top-level Domain - hundreds of options now exist.
- 5 Port - if omitted HTTP will connect on port 80, HTTPS on 443.
- 6 Path - specify and perhaps find requested resource.
- 7 Query String - data passed to server-side software, if present.
- 8 Fragment Identifier - a specific place within an HTML document.

[Static Web Pages]

- A static web page is a web page that is delivered to the user exactly as stored

```
<html>
<head>
  <title>This is a static page</title>
</head>
<body>
  <h1>Hello world</h1>
</body>
</html>
```



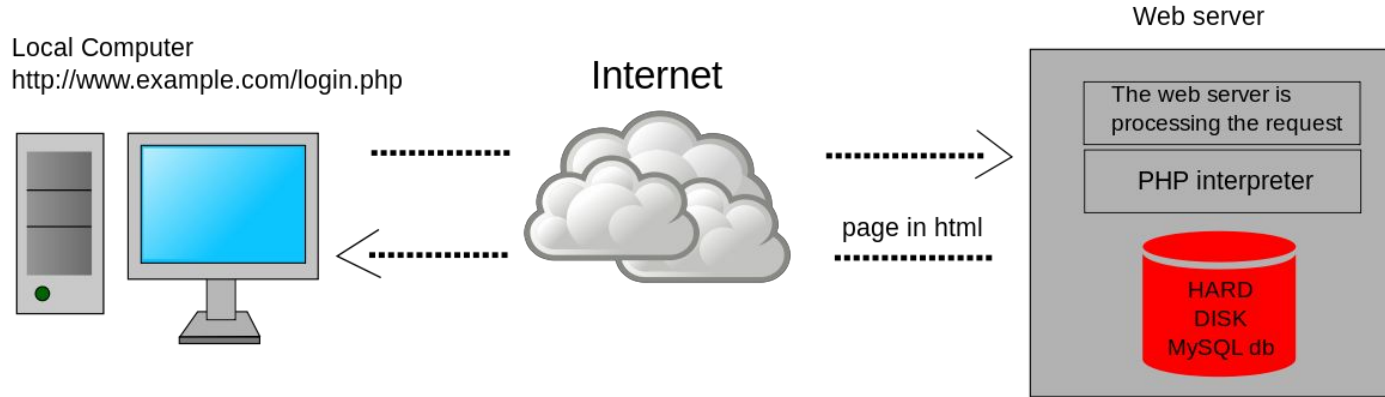
[Sample Dialog Between the Browser and the Server]

- The user enters the following URL in his browser: `http://www.example.com/path/file.html`
- The browser translates it into a connection to `www.example.com` with the following HTTP request:

`GET /path/file.html HTTP/1.1
Host: www.example.com`
- The web server on `www.example.com` appends the given path to its root directory
 - e.g., on Apache server, this is commonly `/home/www`
- The result is the local file system resource: `/home/www/path/file.html`
- The web server then reads the file, and sends a response to the client's web browser
- The response describes the content of the file and contains the file itself or an error message if the file does not exist or is unavailable

[Server Side Dynamic Web Pages]

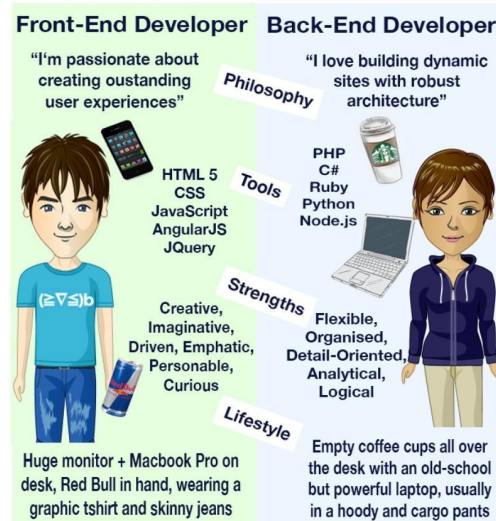
- Server-side processing allows dynamic page creation
- There are many server-side languages for creating dynamic pages such as PHP, Perl, ASP, ASP.NET, JSP, and others



Source: Wikipedia

[Front-End vs Back-End Development]

- The front-end of a website is everything the user sees, touches and experiences
- The back-end of a website works behind the scenes to enable the front-end



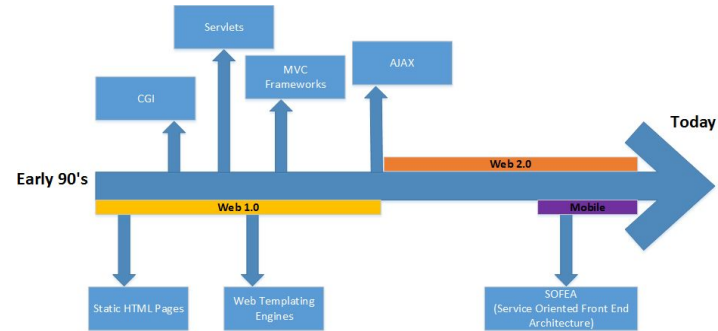
Source: <https://www.pinterest.com/pin/541628292667889162>

[The Big Three]

- While the server can process information in many different languages, the file that it serves to the client must be some combination of the following 3 languages:
- **HTML** – content
 - Different HTML tags describe the structure of content, like paragraphs, blocks, lists, images, tables, forms, etc.
- **CSS** – styling and positioning
 - Tells the browser how each type of element should be displayed, which may vary for different media (like screen, print or mobile device)
- **JavaScript** – application logic
 - Tells the browser how to change the web page in response to events that happen (like clicking on something, or changing the value in a form input)

[Web Development Evolution]

- Static content
- Dynamic content using server side processing
 - Servlets, ASP.NET, PHP
- Desktop like development
 - Rich client libraries, e.g., jQuery, Dojo
 - AJAX - Asynchronous HTTP requests
- MVC frameworks
 - KnockoutJS, EmberJS, AngularJS, Backbone, etc.
- Mobile Web apps
 - HTML5, CSS3, jQuery Mobile
- SOA – service oriented architecture
 - Server delivers data, not content

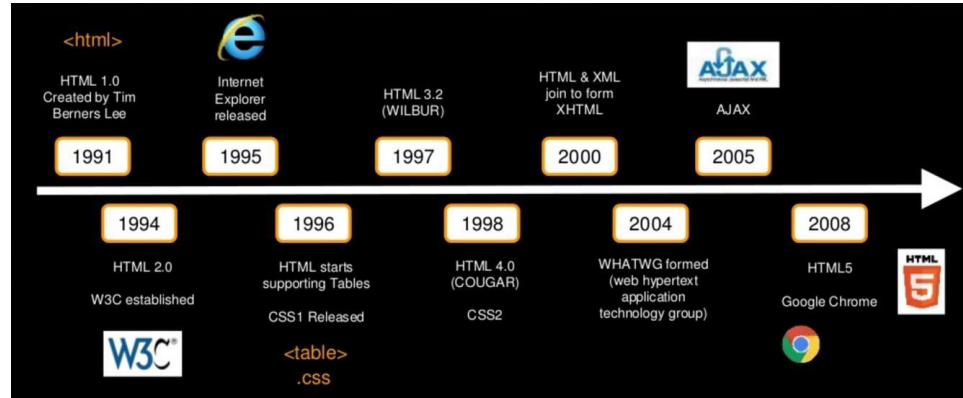


[HyperText Markup Language (HTML)]

- The standard markup language for creating Web pages
- HTML describes the structure of web pages using markup
- The purpose of a web browser is to read HTML documents and display them
- Latest version of the standard is HTML5
 - Completed and standardized on October 2014

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML Tutorial</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>My first webpage!</p>
  </body>
</html>
```

[HTML Versions]



→ XHTML – HTML written as XML

→ XML is a markup language where documents must be marked up correctly (be "well-formed")

→ XHTML is almost identical to HTML but stricter than HTML

→ ensures consistency between browsers

[HTML Editors]

- Web pages can be created and modified by using professional HTML editors
- However, for learning HTML we recommend a simple text editor like Notepad (PC) or TextEdit (Mac)
- We believe using a simple text editor is a good way to learn HTML
- In Windows:
 - Open the **Start Screen** (the window symbol at the bottom left on your screen)
 - Type **Notepad**
- In Mac:
 - Open **Finder > Applications > TextEdit**
 - Also change some preferences to get the application to save files correctly. In **Preferences > Format > choose "Plain Text"**
 - Then under "Open and Save", check the box that says "Display HTML files as HTML code instead of formatted text"

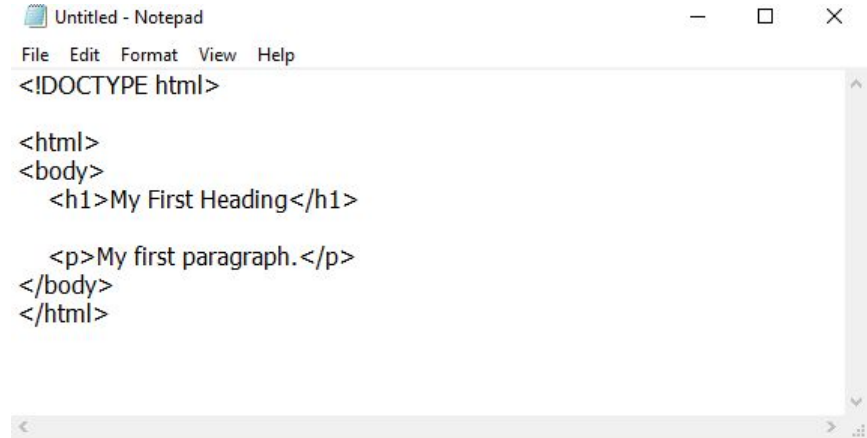
[Write Some HTML]

→ Write or copy some HTML into Notepad

```
<!DOCTYPE html>

<html>
<body>
  <h1>My First Heading</h1>

  <p>My first paragraph.</p>
</body>
</html>
```



The screenshot shows a standard Windows Notepad application window. The title bar reads 'Untitled - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The text area contains the following HTML code:

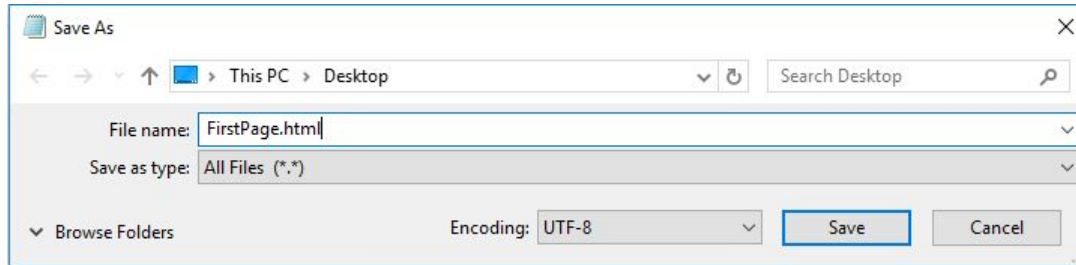
```
<!DOCTYPE html>

<html>
<body>
  <h1>My First Heading</h1>

  <p>My first paragraph.</p>
</body>
</html>
```

[Save the HTML Page]

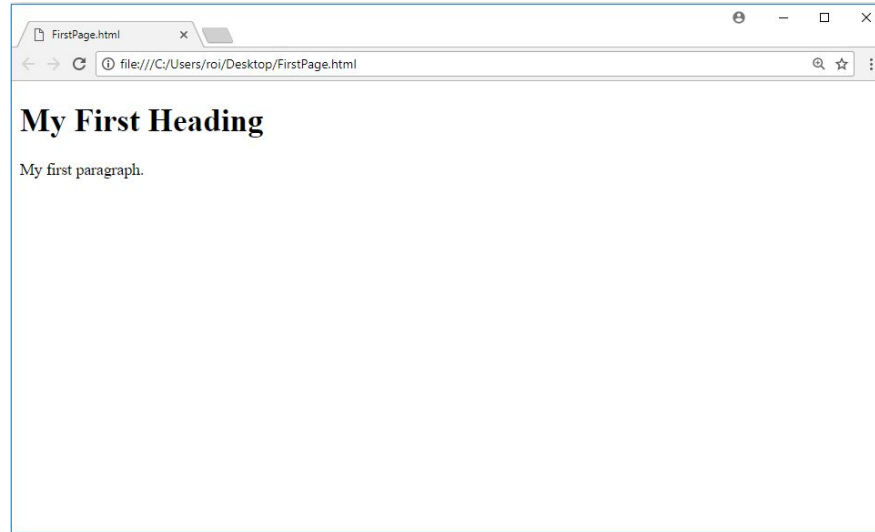
- Save the file on your computer. Select **File > Save as** in the Notepad menu.
- Name the file "**FirstPage.html**" and set the encoding to **UTF-8** (which is the preferred encoding for HTML files)



- You can use either .htm or .html as file extension
- There is no difference, it is up to you

[View the HTML Page in Your Browser]

- Open the saved HTML file in your favorite browser (double click on the file, or right- click - and choose "Open with")
- The result will look much like this:



[HTML Tags]

→ HTML tags are element names surrounded by angle brackets:

`<tagname>content goes here...</tagname>`

→ Example: `<p>` tag creates a paragraph

```
<p>This is a paragraph.</p>
```

→ HTML tags normally come in **pairs** like `<p>` and `</p>`

→ The first tag in a pair is the **start tag**, the second tag is the **end tag**

→ The end tag is written like the start tag, but with a **forward slash** inserted before the tag name

→ HTML tags are not case sensitive

→ `<P>` means the same as `<p>`

→ The HTML5 standard does not require lowercase tags, but W3C **recommends** lowercase tags ,
and **demands** lowercase for XHTML document types

→ A complete list of HTML tags is available at <https://www.w3schools.com/tags/>

[HTML Attributes]

- All HTML elements can have **attributes**
- Attributes provide additional information about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value"**

```
<p title="Tooltip text">This is a paragraph.</p>
```

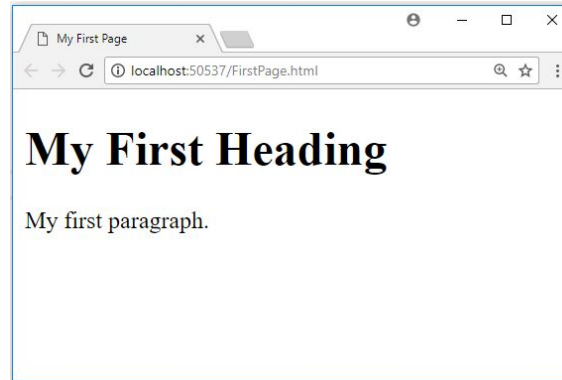
- The HTML5 standard does not require lowercase attribute names
 - The title attribute can be written with uppercase or lowercase like **title** or **TITLE**
 - W3C **recommends** lowercase in HTML, and **demands** lowercase for XHTML document types
- Double quotes around attribute values are the most common in HTML
 - but single quotes can also be used

```
<p title='John "ShotGun" Nelson'>This is a paragraph.</p>
```

[HTML Documents]

- All HTML documents must start with a document type declaration: `<!DOCTYPE html>`
- The HTML document itself begins with `<html>` and ends with `</html>`
- The `<head>` element contains meta information about the document
- The `<title>` element specifies a title for the document
- The visible part of the HTML document is between `<body>` and `</body>`

```
<!DOCTYPE html>
<html>
<head>
  <title>My First Page</title>
</head>
<body>
  <h1>My First Heading</h1>
  <p>My first paragraph.</p>
</body>
</html>
```



[HTML Comments]

- You can add comments to your HTML source by using the following syntax:

```
<!-- This is a comment -->  
<p>This is a paragraph.</p>
```

- Comments are not displayed by the browser, but they can help document your HTML source code
- Comments are also great for debugging HTML, because you can comment out HTML lines of code, one at a time, to search for errors:

```
<!-- Do not display this at the moment  
  
-->
```

[Paragraphs]

- The HTML `<p>` element defines a **paragraph**
- Browsers automatically add some white space (a margin) before and after a paragraph

```
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>
```

This is a paragraph.

This is another paragraph.

- Use `
` if you want a line break (a new line) without starting a new paragraph
 - The `
` tag is an empty tag, which means that it has no end tag

```
<p>This is<br/>a paragraph<br/>with line breaks.</p>
```

This is
a paragraph
with line breaks.

[Headings]

- Headings are defined with the `<h1>` to `<h6>` tags
 - `<h1>` defines the most important heading
 - `<h6>` defines the least important heading
- Search engines use the headings to index the content of your web pages
- Use HTML headings for headings only. Don't use headings to make text **BIG** or **bold**.

`<h1>`Heading 1

`<h2>`Heading 2

`<h3>`Heading 3

`<h4>`Heading 4

`<h5>`Heading 5

`<h6>`Heading 6

Heading 5

Heading 6

[Links]

→ Links are defined with the `<a>` tag:

```
<a href="url">link text</a>
```

→ Example:

```
<a href="http://www.google.com">To google</a>
```

[To google](http://www.google.com)

→ The href attribute specifies the destination address (https://www.google.com) of the link

→ The **link text** is the visible part (To google)

→ Clicking on the link text will send you to the specified address

→ A local link (link to the same web site) is specified with a relative URL (without http://www....)

```
<a href="Page2.html">To Page2</a>
```


[Links]

- The **target** attribute specifies where to open the linked document
 - `_blank` - Opens the linked document in a new window or tab
 - `_self` - Opens the linked document in the same window/tab as it was clicked (this is default)

```
<a href="http://www.google.com" target="_blank">To google</a><br />
```

- By default, a link will appear like this (in all browsers):
 - An unvisited link is underlined and blue
 - A visited link is underlined and purple
 - An active link is underlined and red
- You can change the default colors, by using CSS
 - More on this in the CSS module

[Images]

- Images are defined with the `` tag
- The `` tag is empty, it contains attributes only:
 - The `src` attribute specifies the URL (web address) of the image
 - The `alt` attribute provides an alternate text for an image
 - If the user for some reason cannot view it (because of slow connection, an error in the `src` attribute, or if the user uses a screen reader)
 - The `alt` attribute is required. A web page will not validate correctly without it.
- You can use the `style` attribute to specify the width and height of an image

```

```



[Control questions]

1. What is Front-end development?
2. What Front-end is responsible for?
3. Where does Front-end code gets executed?
4. What aspects of web page are done with HTML, CSS and JavaScript?
5. Which HTML tags do you know?
6. How to create a link in HTML?
7. How do you include and image in HTML page?

[Materials]

Core materials:

https://www.w3schools.com/tags/ref_byfunc.asp

<https://dan-it.gitlab.io/fs-book/projects/landing/basic.html#html>

Additional materials:

<https://ru.wikipedia.org/wiki/HTML>

<https://ru.wikipedia.org/wiki/CSS>

<https://ru.wikipedia.org/wiki/JavaScript>

<http://htmlbook.ru/html/%21doctype>

<http://htmlbook.ru/html/attr/lang>

Video materials:

https://www.youtube.com/watch?v=LICSA6iJd6w&list=PLAKxGhxbBWw_jeD7pBLK8-V_ehxGgy11Z

https://www.youtube.com/watch?v=I8Hx4BFBUY0&list=PLAKxGhxbBWw_jeD7pBLK8-V_ehxGgy11Z&index=2

https://www.youtube.com/watch?v=ckIrbSyIkCI&list=PLAKxGhxbBWw_jeD7pBLK8-V_ehxGgy11Z&index=5

https://www.youtube.com/watch?v=hGeozYVBKpU&index=9&list=PLAKxGhxbBWw_jeD7pBLK8-V_ehxGgy11Z