Intro to Web Development

Goals

Computer & Lamp

In this lesson you will learn about:
1. The World Wide Web - WWW
2. How the web works
3. Career opportunities as a Frontend Developer and related areas

Introduction

This lesson is intended to introduce you to the basics of Web Development. The topics covered will help you understand how the web works and the advantages of pursuing a career in web development.

Web development offers good career opportunities, not only because it is the most in-demand job in technology, but also because it is one of the highest paying skilled jobs with a great deal of room to grow. There are also many advantages, such as being able to work remotely, and move up the career track quickly. There are two main career paths: full-time web developer and freelance web developer. A full-time developer generally works for a company on their web code-base. A freelance developer takes on individual projects as a contractor. As in any profession, there are challenges such as to keep up to date. Web development is a career for people with strong problem solving skills who enjoy continuous learning.

You may find yourself stuck on a problem. This happens to everyone! It is not uncommon for people to feel “impostor syndrome”, which means they feel like everyone has skills that they don’t have. One of the main reasons for this is that there is so much to learn, no one can know everything or

figure out how to solve every problem on their own. If you are willing to persevere, you'll get there! Web developers will often stop working and take a walk or make a cup of coffee in order to come back to a problem with fresh eyes.

Coffee

The World Wide Web

The World Wide Web - WWW

Watch this video and write your own description of what the world wide web is.

The WWW in my own words...

Exercise 1:

Curious about how many pages the Web has? What do you think is the approximate number of pages that the web currently has?

More information here!

Who invented the Web?

The World Wide Web was invented by British scientist Tim Berners-Lee in 1989 while working at CERN.

Take a look at the first Web page. At the end of our first week of learning, you'll be able to create something like this!

Wikipedia as a part of the W3

The web is currently the most extensive source of information that exists, in the same way that libraries like the one in Alexandria were in ancient times, where the source of knowledge of the time was concentrated.

In modern times, we have access to more knowledge than the ancients could have dreamed of. Who has not browsed wikipedia? Wikipedia is the collaborative encyclopedia of the World Wide Web with more than 50 million articles in 328 languages.

How the Web Works

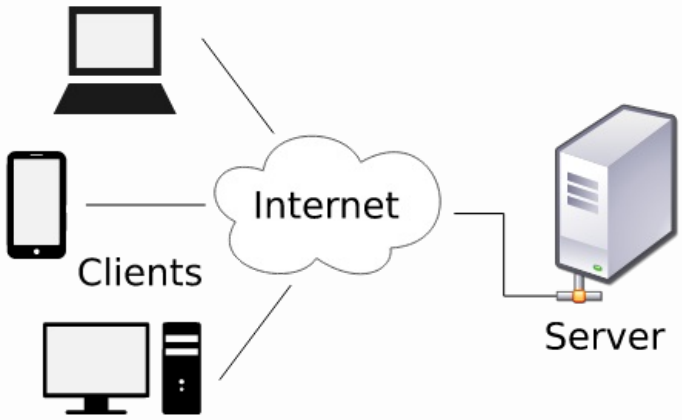
How does the Web work?

Most of us use the web without knowing how it really works. The Internet is a network of computers connected to each other by cables.

Some computers play the role of servers because they are connected directly to the internet and store web pages as files on their hard disks. There are many, many, servers, and they are where the internet “lives”. When you build your own web page, you will probably host it on a cloud server. You can even learn to use your own computer as a server and host your own page! This used to be more common than it is now, because cloud hosting has become inexpensive and is more flexible; if you hosted your content on your home computer you could never turn it off! If the power were to go out, your page would not be accessible. Each server has a unique identifier called an IP address, and it works like a postal address .

Computers that consume content or services are called clients. Clients are connected indirectly to the Internet through Internet service providers (ISP). An ISP is a company that offers infrastructure and services for accessing and using the Internet. They’re your gateway.

A Web browser is the program used for accessing websites and retrieving its files from a web server.



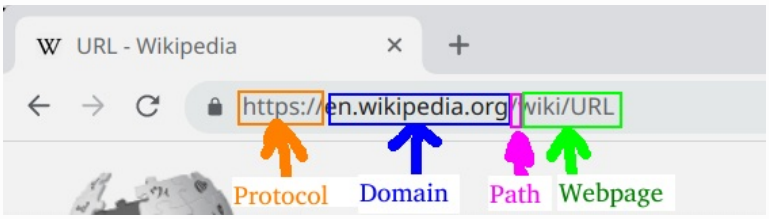
Source: https://commons.wikimedia.org/wiki/File:Client-server-model.svg

Web Resources

Identifying resources on the Web

URL - Uniform Resource Locator

Web browsers need to access a web address to locate a web resource, this is possible through the URL of the website. The image below shows Wikipedia's URL. Notice it is composed of several elements:



web address components

What is a Protocol?

A protocol can be defined as a set of rules that establish how computers will send data. The Hypertext Transfer Protocol (http) is the protocol that describes the behavior for transferring data between a server and a client. In other words `http://` and `https://` tells the web browser to expect an address.

Domain

This is the website name. It is a set of characters to identify the computer or server where the page is stored.

Path

It works like the folder structure on a computer. Each '/' indicates the subfolder where a page is located.

Webpage

The ending part of a URL is the filename of the specific page we request.

Hosting services and Domains registrars

Websites are hosted on a hosting server and you need to pay to publish your site there. It is also necessary to register the domain of your site so that no other website uses your domain name, domain registrars take care of this registration.

Consider the web address https://www.correlation-one.com/about and what the components represent.

Career Opportunities

Career opportunities as a Front-end Developer and related areas

Positions and Job titles
* Front-end Developer
* Front-end Engineer
* Web Designer Front-end QA
* Engineer Front-end [Angular/React/Vue]
* Developer Application
* Developer Front-end Software
* Engineer UI Developer HTML
* Developer Solutions Engineer Product Designer
* QA/Test Engineer
* Programmer Analyst
* Analyst
* Project Analyst
* And more! Titles are less important than the day-to-day tasks involved in the position.

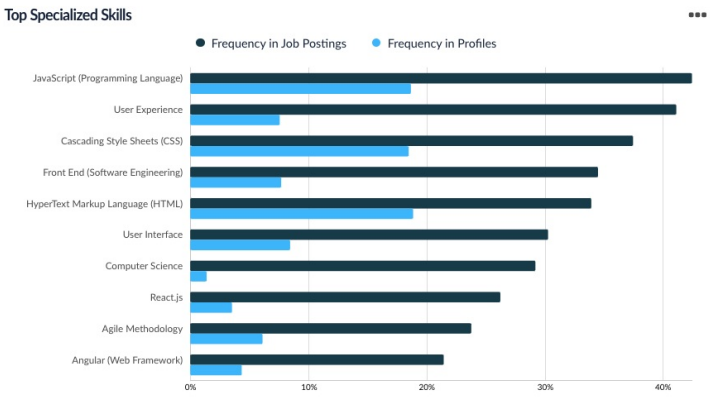
A day in the life of a front-end developer

Exercise 2:

Read this article and describe in your own words 3 things a web developer does on a day-to-day basis.

Salaries & statistics

A career in web development demands specific skills. The figure below describes these skills and their frequency in job postings and profiles.



Hide Detailed Table

Skills	Postings	% of Total Postings	Profiles	% of Total Profiles
JavaScript (Programming Language)	13,659	42%	69,192	19%
User Experience	13,229	41%	28,099	8%
Cascading Style Sheets (CSS)	12,058	37%	68,494	18%
Front End (Software Engineering)	11,105	35%	28,477	8%
HyperText Markup Language (HTML)	10,909	34%	69,856	19%
User Interface	9,738	30%	31,426	8%
Computer Science	9,387	29%	5,069	1%
React.js	8,439	26%	13,200	4%
Agile Methodology	7,653	24%	22,725	6%
Angular (Web Framework)	6,906	21%	16,187	4%

Job Data

- The data show Javascript as the main skill
- There are 24,433 total Web Dev postings in this search
- Out of those, 3,408 explicitly mention Wordpress (14%)
- 480 ask for both Wordpress and Angular (median salary 67.5k)
- 836 ask for Wordpress and React.js (median salary 73.6k)
- The remaining 2k or so are just Wordpress. (Though what that means is not clear, since there are various ways to “know” wordpress.)
- Median salary for just Wordpress is 63.9k 2,593 ask for both Wordpress and Javascript (median salary 64.9k) 2,881 ask for both Wordpress and CSS (median salary 64.4k) 2,286 ask for both Wordpress and HTML (median salary 62.6k)

Conclusion & Takeaways

- A Web Developer career is a promising and profitable career with continuous learning.
- The WWW is the greatest source of information of our times.
- The web works on the basis of a network of computers that function as servers or clients connected through the Internet.

Attribution

Free Stock Photos, PNGs, Templates & Mockups. (n.d.). Rawpixel. Retrieved October 25, 2022, from https://www.rawpixel.com/image/269818/image-rawpixelcom
Freepik. (2021, January 21). People remote working illustrated Free vector. https://www.freepik.com/free-vector/people-remote-working-illustrated_12242105.htm
Image internet, https://upload.wikimedia.org/wikipedia/commons/c/c9/Client-server-model.svg
Top specialized skills, Burning Glass report data, September 19, 2022.

Anatomy of a Webpage

Anatomy of a Webpage

Goals

Throughout this lesson you will learn about:
1. Websites and their types
2. Identify the main parts of the structure of a web page

Introduction

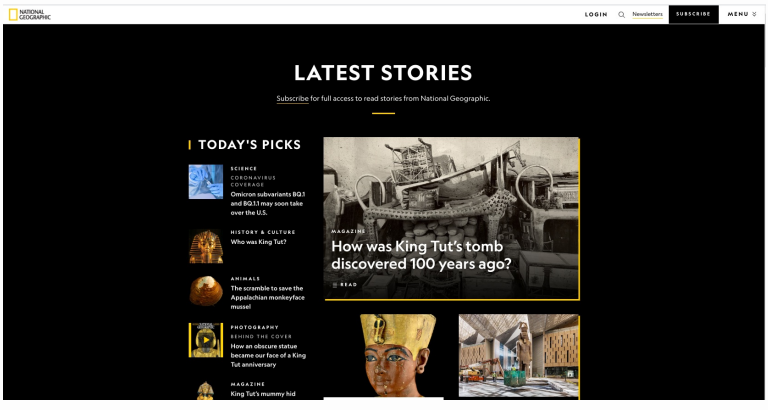
Designing and building a website is a process that basically involves three stages: (1) Identify and analyze the needs of the client and the target audience, (2) Design the user experience (UX) (a topic we will discuss in depth later in this course), and (3) Implementing the code in the syntax of a language that can be interpreted by a web browser.

The analysis stage is very important because this is where the elements that will guide the following stages are defined.

Let's begin with an exploratory analysis of famous websites. Each website has a special way of displaying its content, however, there are some elements that are common to most of them.

What is a webpage/website?

A webpage is an HTML document and a website is a collection of webpages related with a common purpose.



Nat Geo Example

The image above is from the National Geographic website, where you will find pages on animals, the environment, history, culture, science, and travel.

Types of Websites

- **Ecommerce websites**: Ecommerce websites allow users to shop for and purchase products or services online.
- **Personal websites**: People use them to display their personal interests, hobbies, and highlights of their lives.
- **Portfolio websites**: Professionals use them to display examples of their work, skills and experience.
- **Business websites**: Designed to build brand presence.
- **Blog websites**: Sites where people share their knowledge, ideas and experiences on a given topic.

Exercise 1:

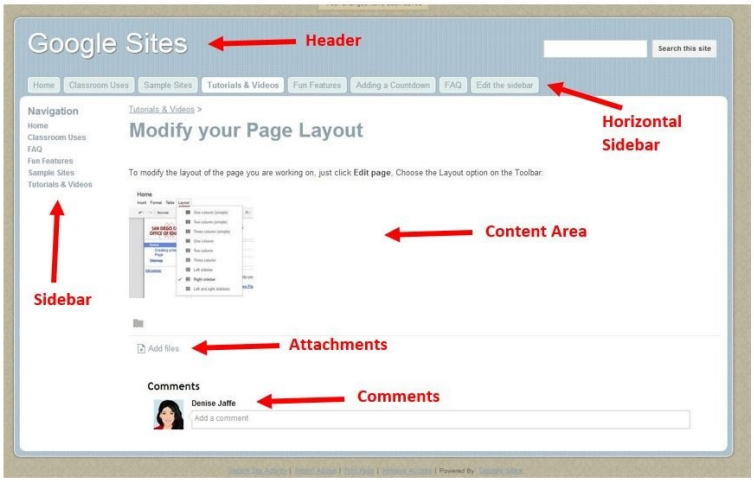
Select one of the types of web sites and find an example.

Content Structure

How is the content structured?

When we browse the web we expect to find what we are looking for located in a certain way within the website. There is an unspoken way of structuring the content of a website.

A web page typically has a basic structure composed of several elements. Look at the image below and identify the different elements.



Webpage Anatomy

- Header: It is at the top of the page and it usually has a logo and the name of the website. It is also common to find a horizontal navigation menu bar in the header.
- Content area: This is the body of the page and it is the most relevant part.
- Sidebar: It is the side menu and can be located on the left or right side of the website.
- Footer: It is located at the bottom of the page. It usually includes links to information of interest such as copyrights, terms of use and contact information.

Challenge your skills!

Exercise 2:

What is the anatomy of this website?

NASA webpage

Web History

Website Changes over Time: The Wayback Machine

What were the most famous websites like 20 years ago? Go to this link and explore its contents. It's what amazon.com looked like in 2001! The difference between that page and the current page show us how far web development has come in 20 years. You can see more website snapshots at https://archive.org/web/.

Have you ever browsed a poorly made website?

Choose a site from 10 or more years ago, and consider what you would do to update the site.

Exercise 3:

Conclusion & Takeaways

The anatomy of a website establishes the structure of its content and is very important for a good user experience.

Attribution

Image Natgeo. (n.d.). National Geographic. Retrieved October 25, 2022, from https://www.nationalgeographic.com/
Anatomy of a Google Site - Google Sites. (n.d.). Retrieved October 25, 2022, from https://sites.google.com/a/whps.org/google-sites/tutorials-videos/anatomy-of-a-google-site
National Aeronautics and Space Administration. (n.d.). NASA. Retrieved October 25, 2022, from https://www.nasa.gov/

Intro to Hypertext Markup language - HTML

HTML is the language to create web pages. It is characterized as a markup language, which means that it uses tags or markup to define and structure content. It is important to keep in mind that HTML is not a programming language, but a language to be **interpreted by a browser**.

What is a Markup Language (ML)?

Computers do not understand human language. There are different types of languages: Markup, Programming, and Scripting. ML uses tags to control the structure, format and relationships between a set of symbols. Scripting languages and programming languages do many of the same things, like performing complex mathematical operations. Scripting languages require a host, and are often used in web development, where they tell a browser what to do.

ML examples:

Hypertext Markup Language

Extensible Markup Language

Markdown Language

Each of these markup languages is used for different purposes. HTML is preferred for web development, while XML is associated with data transport and markdown is often used in non-web content development, as it is easier to read than HTML but does not have the functionality of HTML.

How does a web browser work?

Web navigators *(commonly known as web browsers)* are applications that allow the visualization of web pages. Browsers **interpret the HTML tags** *(and other web programming languages)* to render the content on the screen. Browsers allow **navigation** between documents and hyperlinks through a communications protocol called Hypertext Transfer Protocol (HTTP or HTTPS).

Which browser are you using at the moment?

- Google Chrome
- Safari
- Microsoft Edge
- Firefox
- Opera
- Other

Common elements of a webpage

Webpages have different types of elements: Headings, Paragraphs, Links, Images, Lists, Tables, Regions, Forms, etc.

Basic HTML Syntax

This code, known as the document type declaration, must be the first line of an HTML file to indicate to the browser that the page is written in HTML language: `<!DOCTYPE html>`

HTML syntax is based on tags. We use tags to separate content, and for ease of formatting. When you create a document, you don't just write a wall of text; it's difficult to read. When we learn CSS, we are going to be able to tell it to format everything with a certain type of tag in a certain way. This means we want to get comfortable with separating content with tags, and start thinking ahead about how you might want to format those things.

That is the main point of HTML! We separate and organize text by adding tags, and then we can style different parts of our HTML with CSS! This means that to start, we want to learn what the various tags are, and get comfortable organizing our text with tags. *(We can also create some very basic web pages using only HTML tags.)*

In the image below, the `<p>` tag is a tag that specifies a paragraph element. An HTML paragraph element consists of an opening `<p>` tag, the text or content that makes up the paragraph, and a closing `</p>` tag.

In HTML, a tag is a keyword surrounded by angle brackets (< and >) that specifies an HTML element. An HTML element is a complete set of opening and closing tags, attributes, and content that describes the structure and content of a web page.

Every element has:

1. **Opening part**: <> angular brackets, tag name inside, attributes can be attached.
2. **Content**: any data
3. **Closing part**: </> similar to opening part, forward slash before the tag name . No attributes, Opening and Closing tags must match.

> Attributes are used to modify the behavior, appearance, or functionality of an HTML element.

<html>

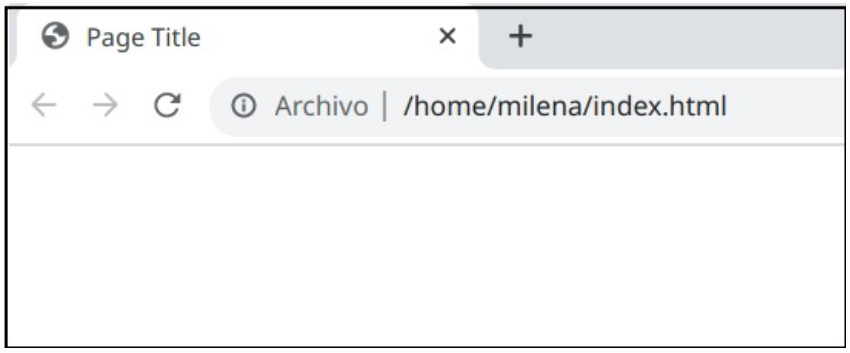
Tag to indicate the start of the html document. All following tags must be inside this tag.

```
<!DOCTYPE html>
  <html>
    ...
  </html>
```

<head>

Tag <head> is a container for data about data (Metadata). It is located after <html> tag and before <body> tag. The webpage title goes inside <head> tag.

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
  </html>
```



A webpage with only a title and no content

In the above image, the page title is “Page Title”. You can change it by changing the text between the <title> tags in the html.

<body>

Inside the <body> tag is the content or visible part of the page.

```
<!DOCTYPE html>
  <html>
    <head>
     <title>Page Title</title>
    </head>
    <body>
      ...
    </body>
  </html>
```

Exercise 1:

Right-click on the current web page and from the context menu select View source code (or something similar, like “view page source”) that the browser is interpreting for you.

What tags can you see? (Tags are generally the first thing following the < symbol, such as <body> and <p>.)

First Coding

The IDE and Coding

On the left, you should see what is referred to as an IDE: an integrated coding environment. The code is at the top, and at the bottom is a preview.

Change the page title, and add some body text to see how it looks. You may need to press the refresh button at the top of the window in the lower left of your screen. Refresh it with the button to the right in the image below:

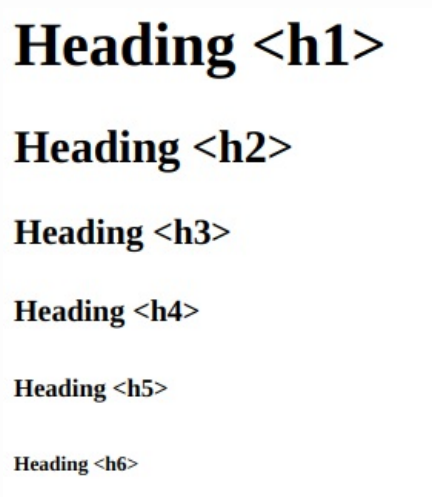
refresh

The button on the left will launch your code in a new webpage. There's nothing there, other than the tab title. Add some text to the body, and then when you refresh the webpage, you'll see your text.

As you follow along on this page, you can also try out these in the IDE on the left and refresh to see how it looks. We learn coding by doing (and making lots of mistakes and they trying to fix those mistakes).

Headings <h1>, <h2>, ..., <h6>

These are Headings used for titles and subtitles. These are useful in organizing the text on your webpage.



headings

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
    <body>
      <h1>Heading &lt;h1&gt;</h1>
      <h2>Heading &lt;h2&gt;</h2>
      <h3>Heading &lt;h3&gt;</h3>
      <h4>Heading &lt;h4&gt;</h4>
      <h5>Heading &lt;h5&gt;</h5>
      <h6>Heading &lt;h6&gt;</h6>
    </body>
  </html>
```

Paragraphs <p>

This tag is used to format text as a paragraph

paragraph

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
    <body>
      <p>This is a paragraph.</p>
    </body>
  </html>
```

Links <a>

Tag to define hyperlinks. <a> tag has an attribute href to set the link's source.

hyperlinks

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
    <body>
      <a href="https://www.google.com">This is a link to Google.
</a>
    </body>
  </html>
```

Note: 1) the value of the href attribute is the link to another web page, in this case the Google page. 2) The underlined text is placed immediately after closing the opening part of the tag <a>.

Images <img>

embedded image

```
<!DOCTYPE html>
  <html>
    <head>
      <title>Page Title</title>
    </head>
    <body>
      <img src="./image10.jpg">
    </body>
  </html>
```

Where the image above, “image10” would be that sunset image.

This part “./image10.jpg” contains two pieces of information. The period (“.”) represents the current directory, which is the directory that the HTML file is located in. The forward slash (“/”) is used to separate directories and subdirectories. Therefore, “./image10.jpg” refers to an image file named “image10.jpg” located in the same directory as the HTML file.

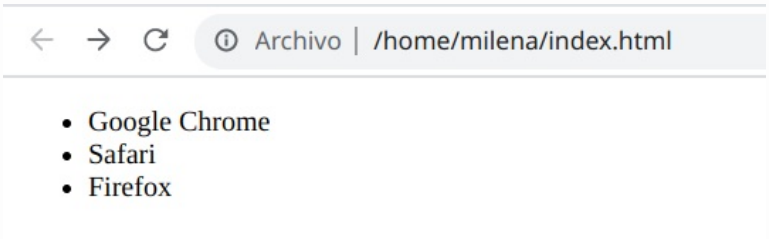
The image extension (“.jpg”) is just as important as everything else. Everyone who has created web pages for a while has wondered why their image wasn’t showing, before realizing they were using the wrong extension, such as .png when the image was a .jpg!

If you want to try this, go to `File/Upload` and upload your own image, and then put the title of your image in the code (no path should be necessary).

Lists

HTML allows us to create lists of items. These lists can be bulleted or numbered. There are three types of tags we consider in creating a list. <**ul**> means “unordered list”, which gives us bullets. <**ol**> means “ordered list” and gives us numbers. The individual items in both lists are marked with <**li**> which stands for “list item”.

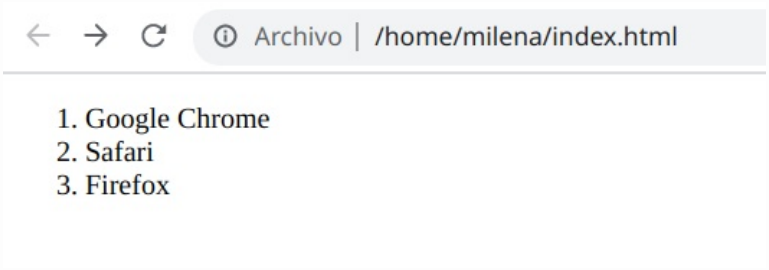
Bulleted lists - tags <ul>



bulleted lists

```
<ul>
  <li>Google Chrome</li>
  <li>Safari</li>
  <li>Firefox</li>
</ul>
```

Numbered (ordered) lists - tags



numbered lists

```
<ol>
  <li>Google Chrome</li>
  <li>Safari</li>
  <li>Firefox</li>
</ol>
```

Exercise 2:

Select the correct HTML code to link an image:

Exercise 3:

Select the right HTML

Attribution

Html icon **https://icon-icons.com/download/78657/PNG/512/**
Xml icon **https://icon-icons.com/icon/xml-filetype/177509**
Markdown icon **https://icon-icons.com/icon/markdown/159971**
Image Landscape by tawatchai07 on Freepik

Build Your Hello World Page

Hello World!

The simple statement “Hello World!” is used by programmers all over the world as a starting place, as displaying the message on the screen is a great introduction to basic syntax of the programming language or framework you are studying. Syntax is another word for grammar; just like a language without grammar would be difficult to use, syntax is the way you structure statements in a program to get the computer to do what you want.

Your first HTML webpage

To create a simple HTML page you will need a (1) text editor and a (2) browser. We are going to use the IDE (integrated development environment) embedded in this lesson.

The HTML code of a basic web page and the result after running the code is shown in the following block:

```
<html>
<body>
    <h1>Hello World!</h1>
    <p>Write your own custom HTML content here...</p>
</body>
</html>
```

Follow the following steps to write your own code:

In the upper left, you should see a text editor. This is a file named “index.html”. In that window, you can type in (or copy and paste) the code you see above.

Continue your message in a second line, using either a `<p>` tag or a heading tag of your choice.

Once you type it in, press the refresh button on the window in the lower left of your screen. You should see these two buttons at the top of that window:

refresh

The one on the right is refresh. When you press it, you can see the results of what you wrote in the window. The button on the left will open it in a new tab on your browser. When you start writing more code, you might want to see a larger output. Press that button now to see what your code produces in your browser!

Congratulations! You're on your way.

More Formatting

Page Background

There are a few different ways to add a background image to a web page. If we were to look at how to add a background color to the entire page, we would use the attribute style.

Example 1

```
<body style="background-color:#0014FF">
```

For this example, we have used hexadecimal notation *(also known as hex code)* for the color blue: `#0014FF`. You can consult the color palette here to know the code for all colors. *(In many cases, you can simply name the color.)* What we're doing here is called "in-line styling" *(putting styling directly in the html code)* and once you learn CSS all styling will be in a separate CSS file.

Exercise 1

Add a title to the webpage using `<title>` between the `<head>` and `</head>` tags, and change the background to your favorite color.

```
<html>
    <head>
        <title>My first page</title>
      </head>
<body style="background-color:#FF6D28">
    <h1>Hello World!</h1>
    <p>Click here to create your custom HTML content...</p>
    <h1>Heading &lt;h1&gt;</h1>
    <h2>Heading &lt;h2&gt;</h2>
    <h3>Heading &lt;h3&gt;</h3>
    <h4>Heading &lt;h4&gt;</h4>
    <h5>Heading &lt;h5&gt;</h5>
    <h6>Heading &lt;h6&gt;</h6>
</body>
</html>
```

You can insert an image as the background instead of a color, by replacing that first body tag with something like this:

```
<body style="background-image:
url('https://upload.wikimedia.org/wikipedia/commons/e/e4/Crimson_sunset.jpg');">
```

You can choose whatever freely available online image you like.

There is starter code in the window in the upper left of your screen, showing you what the headings look like. Delete what you don't need. The above code can help you decide how to make changes. Note that the text between the heading tags like `<h1>` and `</h1>` contains both `<` and `>`. Because some symbols like < and > are reserved for HTML syntax, in order to use them as text we have to write them as "HTML entity symbols". If you ever run into an error in your code that contains a symbol, you can generally look up how you should write it in HTML. Unlike in spoken language, where if someone uses incorrect grammar the listener can generally understand the message, incorrect syntax in a computer program can break the program.

🔍 Click the links to find a list of HTML Entities, HTML Symbols, and HTML Emojis.

Now make the following changes to the HTML code in exercise 1 (all of this should be inserted between `<body>` and `</body>`):
* Choose a lighter background color so we can see the text.
* Create an `<h1>` tag and insert the text "My Home".
* Create a `<p>` tag and write the name of the place you live.
* Create an `<h2>` tag after the close of the `<p>` tag `</p>` and include the text "Things I like about my home".
* Create a separate paragraph using the `<p>` tag, where you list things you like about the area you live in.
* View the page in the browser to see what you created!

Paste your code into the box below and submit.

Tools of the Trade

Tools of the trade: text editors, IDEs, and other development tools

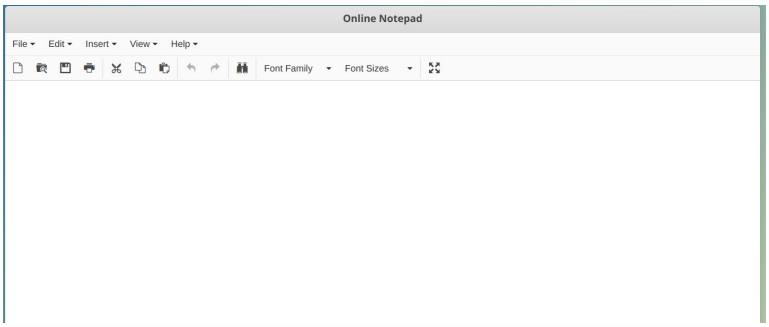
Web development professionals use tools to facilitate the development of their task.

There are programs ranging from the simplest, such as a text editor, to the most sophisticated, such as a project editor.

Text Editors

Text editors are programs to create and edit text files without format (plain text).

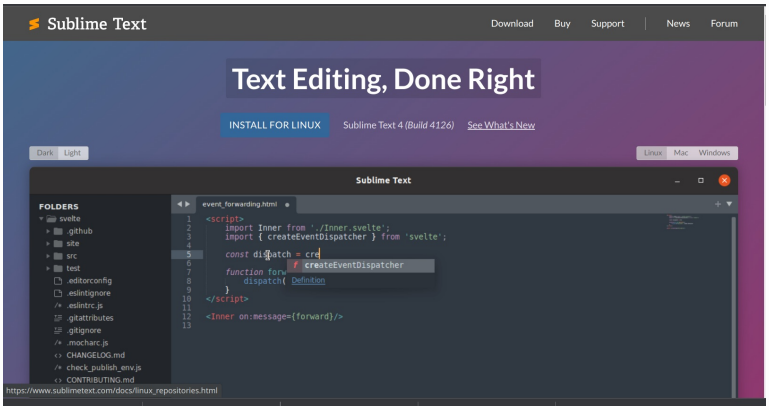
Notepad



Notepad

https://onlinenotepad.org/notepad

Sublime Text



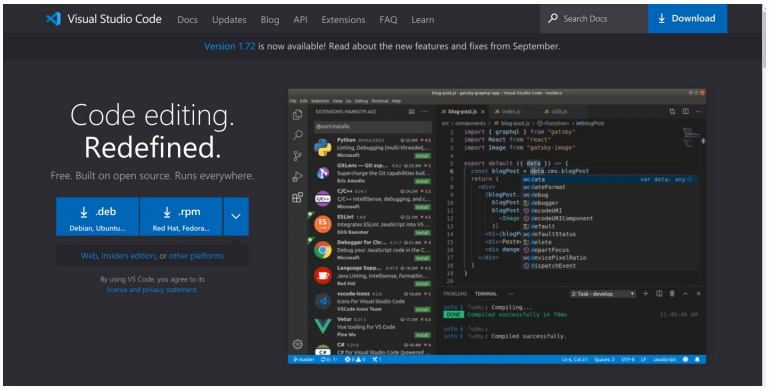
Sublime Text

https://www.sublimetext.com/

IDE - Integrated Development Environment

IDEs are programs developed to facilitate the programmer's task by integrating services and tools that go beyond a simple code editor. IDEs typically allow auto-completion, compilation and debugging of code by offering a more sophisticated graphical interface. This is like what we see in many of our lessons, where we see both the coding environment and the output preview. The example below is vs code, one of the most popular tools used by coders.

Visual Studio code

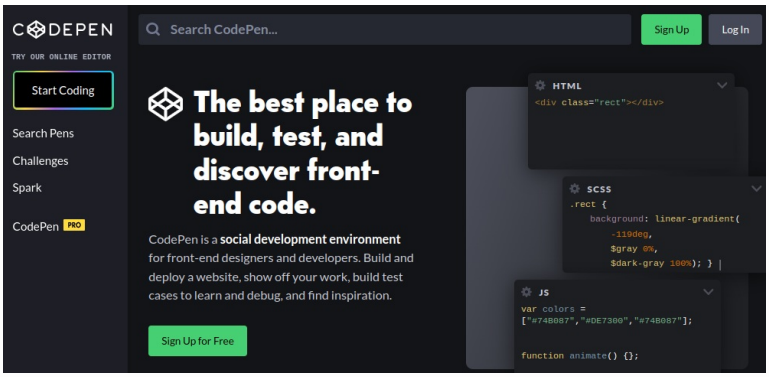


vscode

https://code.visualstudio.com/

Most web developers download an IDE so they can do the work on their own computers. There is often a lot of troubleshooting involved to get things working properly. For instance, if you download vscode, you will have to install plugins in order to see what your code looks like in a web page. This is kind of like when you buy a new phone, and have to figure out the settings and install the apps you want to use. It's annoying, but once you get it set up you're good. Installing vscode is not required for this course, but highly recommended.

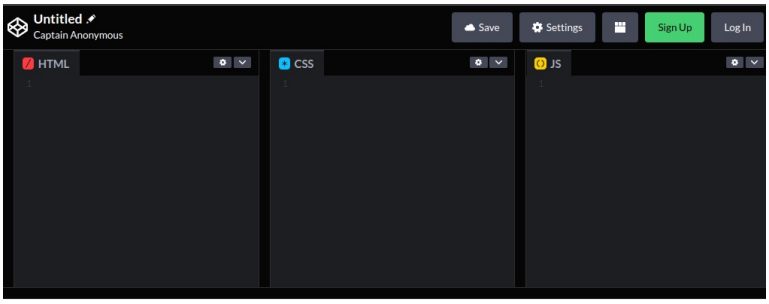
Another coding option is codepen. This is a fully online IDE that allows you to code in html, css, and javascript.



codepen

While you don't have to make an account to use codepen, it is a good idea to do so because (1) it is free, and (2) you can save any code you write there.

Go to codepen.io and create a free account. You should then be able to click on "Pen" on the left and you will see three different coding windows on the page:



codepen

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible"
content="IE=edge">
  <meta name="viewport"
content="width=device-width, initial-
scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

That empty white space above is where you will see what your code looks like when it runs!

The white space below the coding enviornment is where your code will preview, so you can see what you are creating in real time. A lot of coding environments harbor a secret command. Instead of typing in all the basic code that no one ever actually remembers, type an exclamation point `!` into the html window and press tab.

All of that text is self-populated! (The same thing happens in the coding environment we use here.) Now, you'll notice that nothing happened in the white space yet. But as soon as you create text inside the `<body>` (remember to use tags, like `<p>`, and close your tags like `</p>`), you can see it in the space below!

Produce "Hello World" in codepen.io, give it a background color, and embed a link to your favorite website. Save your pen, and share the link to your pen here (just copy the URL for your pen).

Once you learn CSS and javascript, you can create your webpage right in an IDE and check to see how the changes you made in the code affects the output.

Happy coding!

HTML Basics Review

Goals

Review the basic HTML tags

How to write HTML

HTML language is composed of tags. Each tag has a purpose and a syntax.

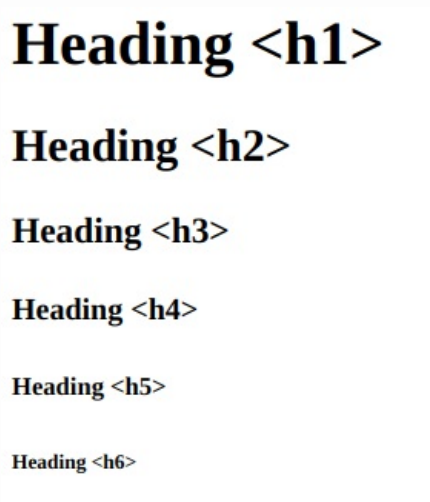
Hello World

Every element has:
1. **Opening part**: <> angular brackets, tag name inside, attributes can be attached.
2. **Content**: any data
3. **Closing part**: </> similar to opening part, forward slash before the tag name. No attributes, Opening and Closing tags must match.

Common HTML elements

Headings from `<h1>` to `<h6>`

These are Headings used for titles and subtitles:



headings

Paragraphs & Images

Paragraph

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

Browsers automatically incorporate a blank space before and after a paragraph.

Image

```
<img src="./image.jpg">
```

Remember that the path to find the image is placed in the src attribute. The name and extension of the file with the image must match, otherwise the browser will not find the image and it will not be displayed. Note that the img tag does not need to be closed.

Exercise 1:

Write HTML code to create two paragraphs about your favorite artist. Use a heading tag to write the artist's name. Then add an image.

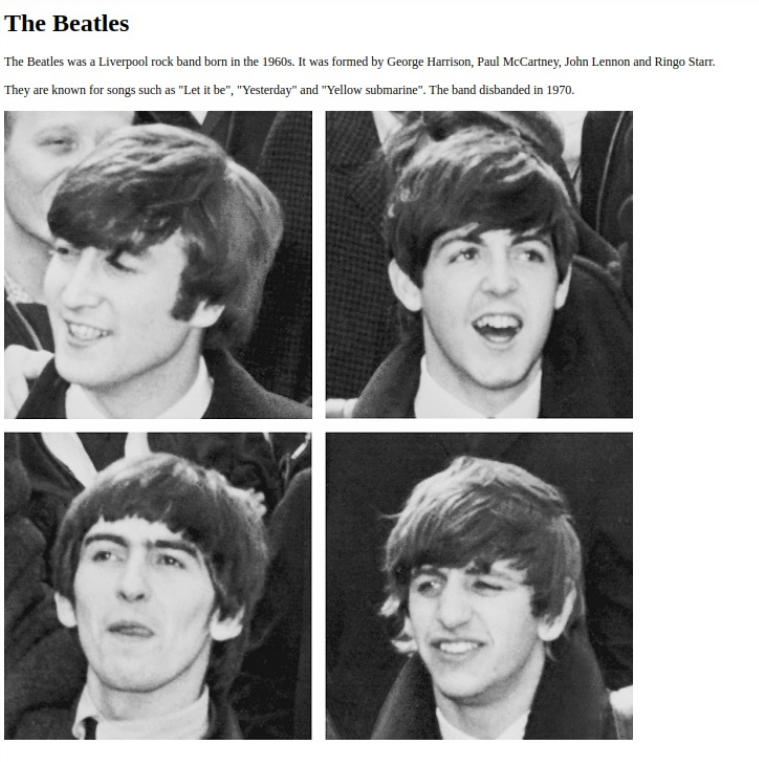
Remember, you can add an image by going to `File/Upload`, and no path is necessary. Format as you see below.

Take a look at the example for how headings were used for titles and subtitles.

Example:

```
<!DOCTYPE html>
<html>
    <head>
        <title>My favorite artist</title>
    </head>
    <body>
        <h1>The Beatles</h1>
        <p>The Beatles was a Liverpool rock band born in the
1960s. It was formed by George Harrison, Paul McCartney, John
Lennon and Ringo Starr. </p>
        <p> They are known for songs such as "Let it be",
"Yesterday" and "Yellow submarine". The band disbanded in 1970.
</p>
        <img src="images/beatles.JPG">
    </body>
</html>
```

Here's what the output would look like:



beatles image

Can you see what happens to paragraph text when the window is resized? Try opening the page in the browser so you can see how it looks.
> The `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.

Once you're finished, submit your code.

Links

Links

Links are an essential part of HTML and are used to link to other webpages or documents. Like adding images, we can add links with a simple syntax. In HTML, links are created using the anchor element `<a>`, which allows you to specify the URL or target of the link. The text between the opening and closing `<a>` tags is the link text, which is what the user sees and clicks on to follow the link.

```
<a href="https://www.google.com/">This is a link to Google.</a>
```

Exercise 2

Write the HTML code to create a web page about your hometown with a link to your town's website and another site that mentions your town. Include at least two types of headings and a list.

Starter code:

```
<!DOCTYPE html>
<html>
    <head>
        <title> </title>
    </head>
    <body>

    </body>
</html>
```

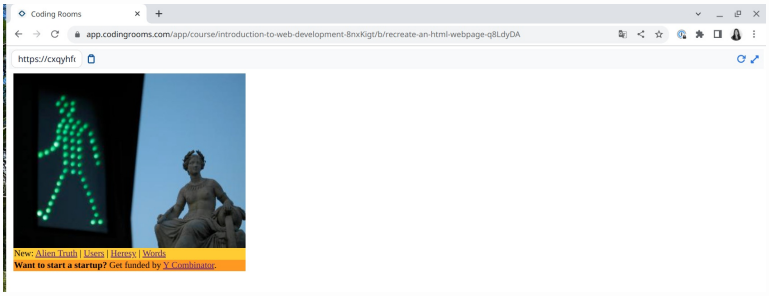
Recreate an HTML webpage

Goals

Through this guided exercise you will be able to:
1. Write HTML code to recreate a basic web page based on a model
2. Practice HTML tags: `<html>`, `<head>`, `<title>`, `<body>`, `<img>`
3. Use the `<table>` tag which will be explained in detail next week
4. Practice pair-coding (time permitting)
>
Introduction to Pair-Coding
This assignment offers a great opportunity to practice a method of learning to code, called pair-coding, if your instructor believes there is time to organize this. In teams of 2, you will share access to a single IDE, and what one person types the other person will see. Generally, one person plays the role of “driver”, and writes the code, while the other plays the role of “navigator”, and tells the person what to write. You may even have exercises like this as a professional developer, when facing difficult coding problems! You will be paired with another learner, and can talk directly through codio or zoom.
>
Introduction to the Challenge
There are millions of pages on the Web, most of which have clean, easy-to-use designs, some of which are quite sophisticated. By the end of this program you will be able to create web sites with many advanced features.

Our task today is to practice what we’ve learned by recreating a simple web page.

Let’s recreate this webpage:



example website

The original webpage is located at http://www.paulgraham.com/index.html (Note: This webpage was chosen for simplicity and for the needs of the lesson. It has since been updated and looks a little different.)

Instructions

We're starting from the beginning, and building out code from there.

Some starting points: you will need that image, which is available with this command: `<img src="Day_3_webpage_image.jpg">`

Write the basic HTML code for a webpage. This is a guide. If you type " ! " and then tab, the editor will create the basic code of an html page automatically. Try to remember the structure and type the code directly in your editor, avoid copy & paste.

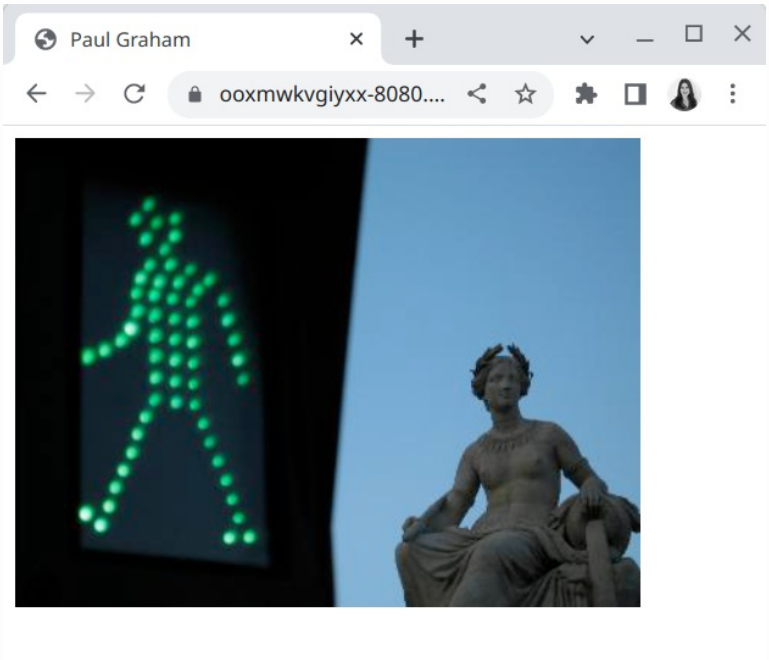
```
<html>
<head>
</head>
<body>

</body>
</html>
```

Add the title. Use tag `<title>`.

You can go to the original website and save the image yourself, but it is already there. Be sure to add the image where you want it using the `<img>` tag and `src` attribute.

At this point, your website should look like this:



website image

Now we want to create the rectangular zones below the image.

New: Alien Truth | Users | Heresy | Words

Want to start a startup? Get funded by Y Combinator.

html tables

Each of these rectangles are HTML tables that are created with the tag `<table>`. Next week you will learn in detail how to create tables, for the moment, you can copy the following code and then just worry about adding the links into the tables.

```
<table width="410" cellspacing="0">
    <tr>
        <td bgcolor="#ffcc33"><!-- Insert links code here -->
</td>
    </tr>
</table>
<table width="410" cellspacing="0">
  <tr>
    <td bgcolor="#ff9922"><!-- Insert links code here --></td>
  </tr>
</table>
```

Add the links `<a>`, browse the original page, and copy the url to the corresponding link. For instance: Alien Truth links to http://www.paulgraham.com/alien.html.

At the end your table should look like this:

New: Alien Truth | Users | Heresy | Words

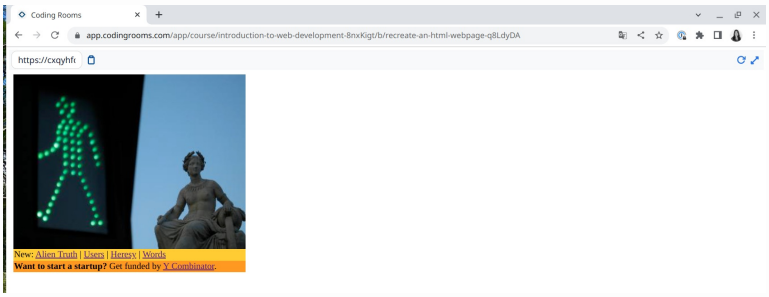
complete table

Do the same for the other table to look like this:

Want to start a startup? Get funded by Y Combinator.

table example

Your recreated page should look like the image below:



final page

If you look at the original webpage, you will notice a navigation sidebar on the left. Can you view the source code for the webpage and figure out how this was done?

When you're finished, submit your code!

Conclusion & Takeaways

With the following tags you can create a simple web page:

- `<html>`: To start an html document
- `<head>`: Place between the tag and the tag. Contains data about the Html document. It is not visible.
- `<title>`: To define a title for the document
- `<body>`: The visible part, it contains all the content of the web page.
- `<img>`: Embed an image in the page
- `<a>`: To define a hyperlink to link other web pages.

Attribution

Paul Graham. (n.d.). Retrieved November 18, 2022, from http://www.paulgraham.com/index.html

A Simple Page

Instructions:

While modern web pages consist of many additional features, we have seen how we can build a simple page in html. Many people have personal pages that allow them to reach consumers, clients, or potential employers. It is a practical necessity that a web developer have their own content online.

For example, this code:

```
<!DOCTYPE html>
<html>
    <head>
        <title> This page is about me! </title>
    </head>
    <body>
        <div>
            <img src="./images/jane.jpeg" alt="Jane"
style="width:30%">
            <div >
                <h2>Jane Wallace</h2>
                <p>Personal Trainer</p>
                <p>I am passionate about exercise and wellness.
</p>
                <p>jane@example.com</p>
                <p><button >Contact</button></p>
            </div>
         </div>
    </body>
</html>
```

(The img path is for example only, and will not work in your built-in ide. Try uploading your own image.)
Will yield this page:

Jane Wallace

Personal Trainer

I am passionate about exercise and wellness.

jane@example.com

Contact

Your “About Me” Page

Project 1

Write the html code for a web page about yourself. This page typically contains the following elements:

1. A short biography. Include sections on:

- interesting experiences
- hobbies
- work goals
- things you enjoy

2. Highlight your work and personal achievements

Be sure to include the following HTML elements learned this week:

1. Headings
2. Images
3. Paragraphs
4. Lists
5. Links

Some of the tags we’ve learned are:

```
<p>
<a>
<li>
<ol>
<ul>
<img>
<h1>, <h2>, <h3>, <h4>
```

Use the built-in ide to write your code and see what your page looks like! Use the `!` trick to get started. (Where, in your empty IDE, you tyle `!` and press `tab` and the basic html starter code prepopulates.)

We will build on this code as we go along in this course! It is recommend that you go back to your codepen account and save your code there when you complete the exercise for easy access in future exercises.

In a few weeks we’ll learn about options for hosting our own web pages, but feel free to do some research on your own. Trial-and-error is a great way to learn.

“So we beat on, boats against the current”, but unlike Fitzgerald’s Gatsby, we are not borne back but push ever forward, making slow and steady progress though continual learning. We hope that this week has been interesting and that you feel a sense of accomplishment in developing your html skills.