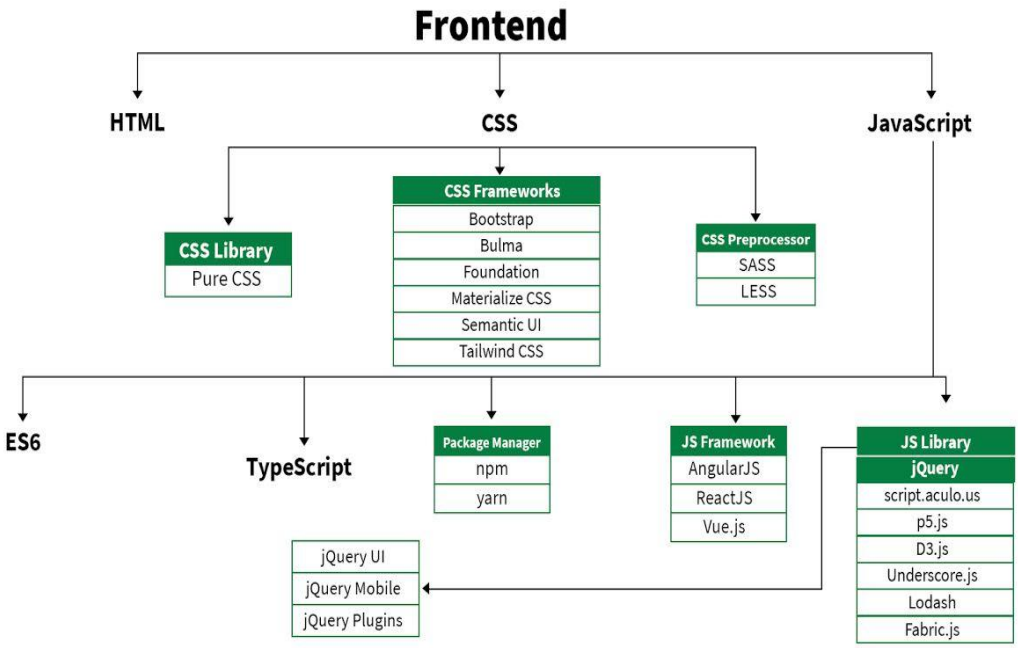


STEP BY STEP EXERCISE WITH - PAUL LEVITES

About the Tutorial

A website can be defined as a collection of several webpages that are all related to each other and can be accessed by visiting a homepage, using a browser like Internet Explorer, Mozilla, Google Chrome or Opera. In this tutorial, we will explain the concept of website development, from the simplest to the most advanced. It will help novice users to learn all about websites and how they are designed and maintained. At the same time, this tutorial has enough material to help even system administrators to broaden their knowledge about websites.

Table of Contents - Unit 1

Table of contents - Unit 2

References

1. https://www.tutorialspoint.com/
2. http://oasis.col.org/
3. https://www.w3schools.com/

Target:

This material is targeted at beginners in web development, and will give them the necessary knowledge needed to get them started to life as a professional Web Developer.

Please note that this E-book was written to introduce aspiring developers to web development, and it does not contain all you need to know to become pro.

To become a professional, constant Research must be made by you and dedication has to be a constant motivation for you to grow.

Main Topics:

1. Hosting
2. CMS - Wordpress

3. Domain Name

Though the focus of this course is on how to create websites using HTML, CSS, JavaScript and other technologies, but mastery of HTML, CSS and other technologies should not be confused with the understanding of the web development process. A web developer should also have basic knowledge about the working of the Internet, TCP/IP protocol suit and concepts related to them. Furthermore, developing large and complex websites require a careful planning and use of a suitable development process. This unit provides an introduction to the basics of Internet and various web development technologies. It provides web development process and various phases involved in website development. Website development process is followed through guidelines and documentation in the systematic way by the developer.

TERMINOLOGIES

IP: **Internet Protocol (IP) is a communication protocol that provides rules for exchanging data among computers.**

TCP: Transmission Control Protocol (TCP) maintains data flow during end to end communication.

ISP: Internet Service Provider is a company that provides Internet services.

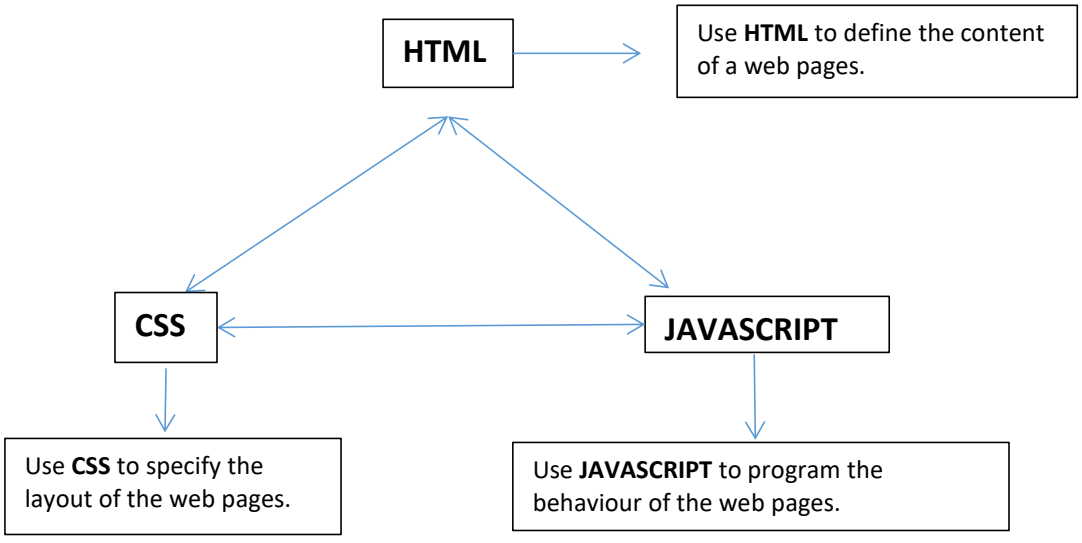
WWW: World Wide Web is a collection of resources over the Internet.

Domain Name: Domain name is used to identify a website on the Internet.

A website can be defined as a collection of several webpages that are all related to each other and can be accessed by visiting a homepage, by using a browser like Internet Explorer, Mozilla, Google Chrome, or Opera. For Example, Google.com.

Every Webpage is Written in **HTML, CSS** and **JAVASCRIPT.** These are the Three basic/fundamentals Languages that every developer must know to build a website.

THE WEB PROGRAMMING TRIANGLE



There are a number of technologies available for developing a website. A web developer should have the knowledge of the following basic technologies to develop a fully functional website.

What Is HTML?

HTML is a Hyper Text Markup Language used to develop a web page. It provides a set of tags that are used to structure the content of web pages and create links with other pages. We will talk more on it later on in this Course. It is used in developing the front-end of the page. I.e The seen part of the page.

What Is CSS?

Cascading style sheets (CSS) is a language that is used to define styles for a web page or web pages. CSS describes how elements of a web page are to be displayed on screen. CSS handles the look and feel of a web page. Using CSS, we can control the color of the text, the style of fonts, the spacing between paragraphs, columns sized and layout, background images or colors, layout designs, and so on.

What Is JAVASCRIPT?

JavaScript Javascript is a light weight programming language which is commonly used in developing dynamic web pages.

Database

A database is a collection of logical related data that can be easily accessed, managed, and updated. A database management system (DBMS) is a software for creating and managing databases.

Self Assessment Questions

1. When a user opens a website, the first page that is displayed is called _____.

A. Backend Page
B. B. Dead End
C. C. Home Page
D. D. None of these

2. Cascading Style Sheets (CSS) is a language that is used to define styles for a web page or web pages.

A. True
B. False

3. A company that provides Internet access to individuals and other companies is called

A. Web server
B. B. Telephone
C. Internet Service Provider
D. None of the above

What Is HTML?

HTML is a Hyper Text Markup Language used to develop a web page. It provides a set of tags that are used to structure the content of web pages and create links with other pages. We will talk more on it later on in this Course. It is used in developing the front-end of the page. I.e The seen part of the page.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Let’s take a house as an example, when you get to your friend’s house for the first time, you notice the outside view before going inside.

A website is just like a house and **HTML** is the first thing(The outside) you notice in a website, The texts, structure, and layouts, are all developed using **HTML**.

A Simple HTML Document

Example

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

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Example Explained

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document
- The `<html>` element is the root element of an HTML page
- The `<head>` element contains meta information about the HTML page
- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The `<h1>` element defines a large heading
- The `<p>` element defines a paragraph

An HTML element is defined by a start tag, some content, and an end tag:

<tagname>Your text goes here...</tagname>

The purpose of a web browser (Chrome, Edge, Firefox, Safari) is to read HTML documents and display them correctly.

A browser does not display the HTML tags, but uses them to determine how to display the document:

We will discuss more on **HTML** in the second unit of this book.

Basci HTML tags

The first HTML tag you'll ever see is called the '**HEADER'** Tag. It is denoted with 'H' or 'h' (Note: The indexing does not matter, but you are often advised to write them in small letters for easier work).

The HEADER tag is used to create headings in websites.

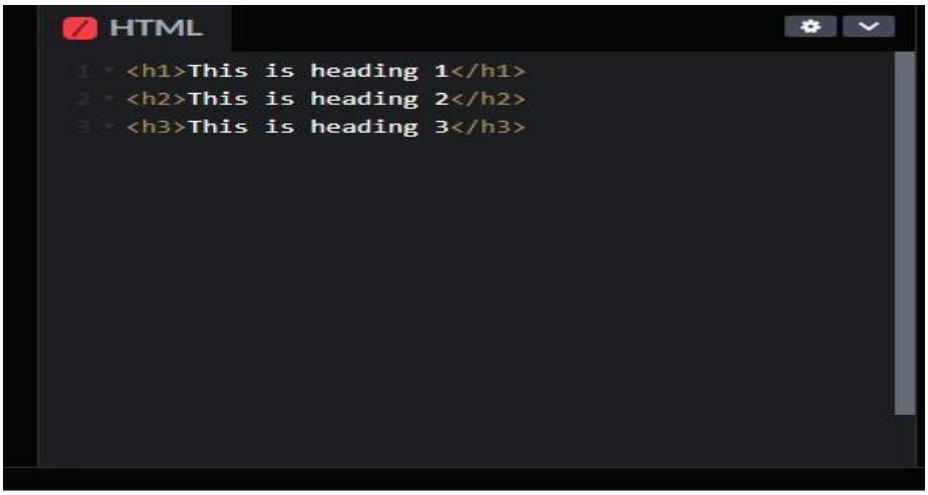
HTML headings are defined with the `<h1>` to `<h6>` tags.

`<h1>` defines the most important heading. `<h6>` defines the least important heading:

For example;

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
```

Output;



THIS IS HEADING 1

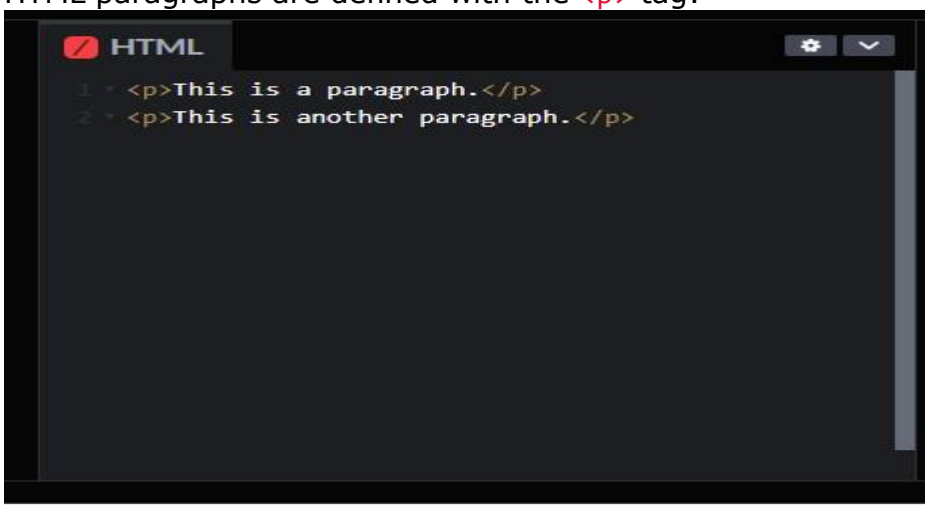
THIS IS HEADING 2

THIS IS HEADING 3

Try out your new knowledge on: codepen.io/pen/

HTML Paragraphs

HTML paragraphs are defined with the <p> tag:



THIS IS A PARAGRAPH.

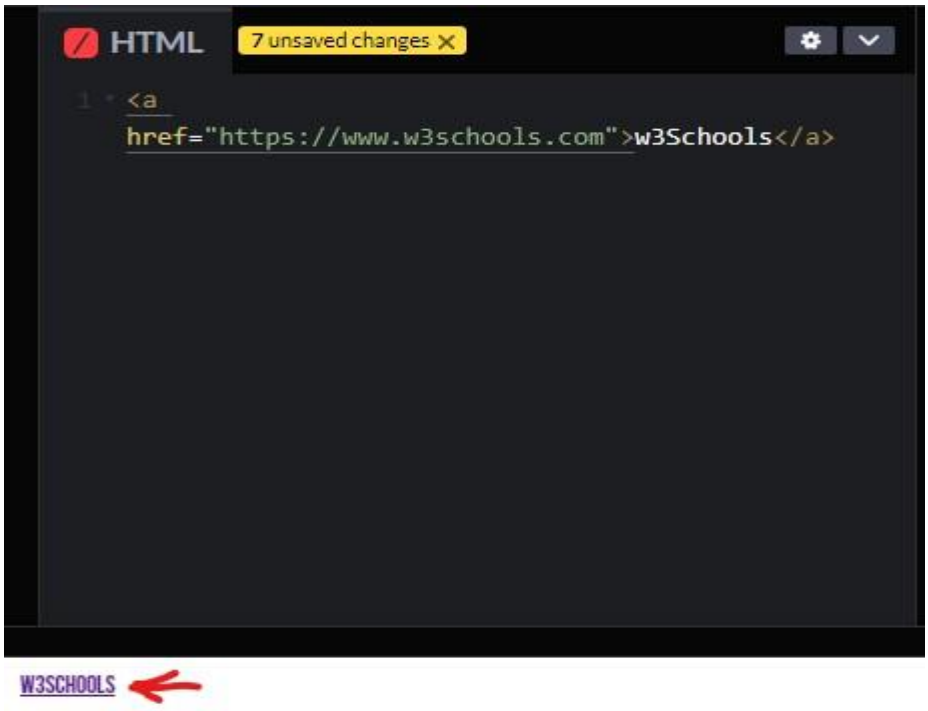
THIS IS ANOTHER PARAGRAPH.

This Tag is Basically used in starting a new page, just like the normal Paragraph in English Language.

HTML LINKS

They are often used to link to another page or website from a particular website. For example, Linking from my website to W3 schools.

HTML links are defined with the <a> tag:



The above shows that my website has a link, linking directly to w3Schools.

How to View HTML Source?

Have you ever seen a Web page and wondered "Hey! How did they do that?"

View HTML Source Code:

Right-click in an HTML page and select "View Page Source" (in Chrome) or "View Source" (in Edge), or similar in other browsers. This will open a window containing the HTML source code of the page.

Inspect an HTML Element:

Right-click on an element (or a blank area), and choose "Inspect" or "Inspect Element" to see what elements are made up of (you will see both the HTML and the CSS). You can also edit the HTML or CSS on-the-fly in the Elements or Styles panel that opens.

We will talk More about **HTML** in the Second Unit of the Book.

CSS - Intro

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

CSS Example

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

CSS Solved a Big Problem

HTML was NEVER intended to contain tags for formatting a web page!

HTML was created to describe the content of a web page, like:

<h1>This is a heading</h1>

This is a paragraph.

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.

To solve this problem, the World Wide Web Consortium (W3C) created CSS.

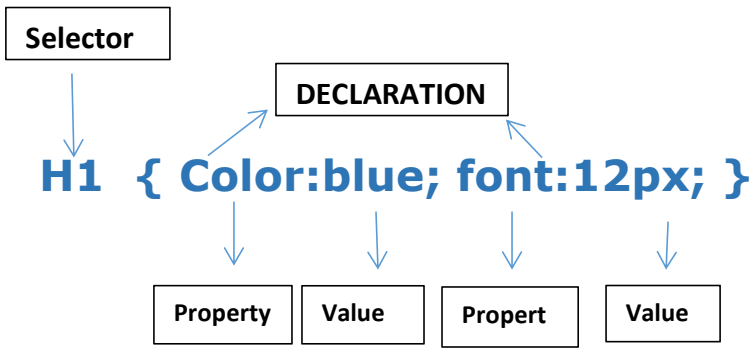
CSS removed the style formatting from the HTML page!

See: https://www.w3schools.com/

CSS files are normally saved in external .css files.

With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS Syntax



The selector points to the HTML element you want to style.

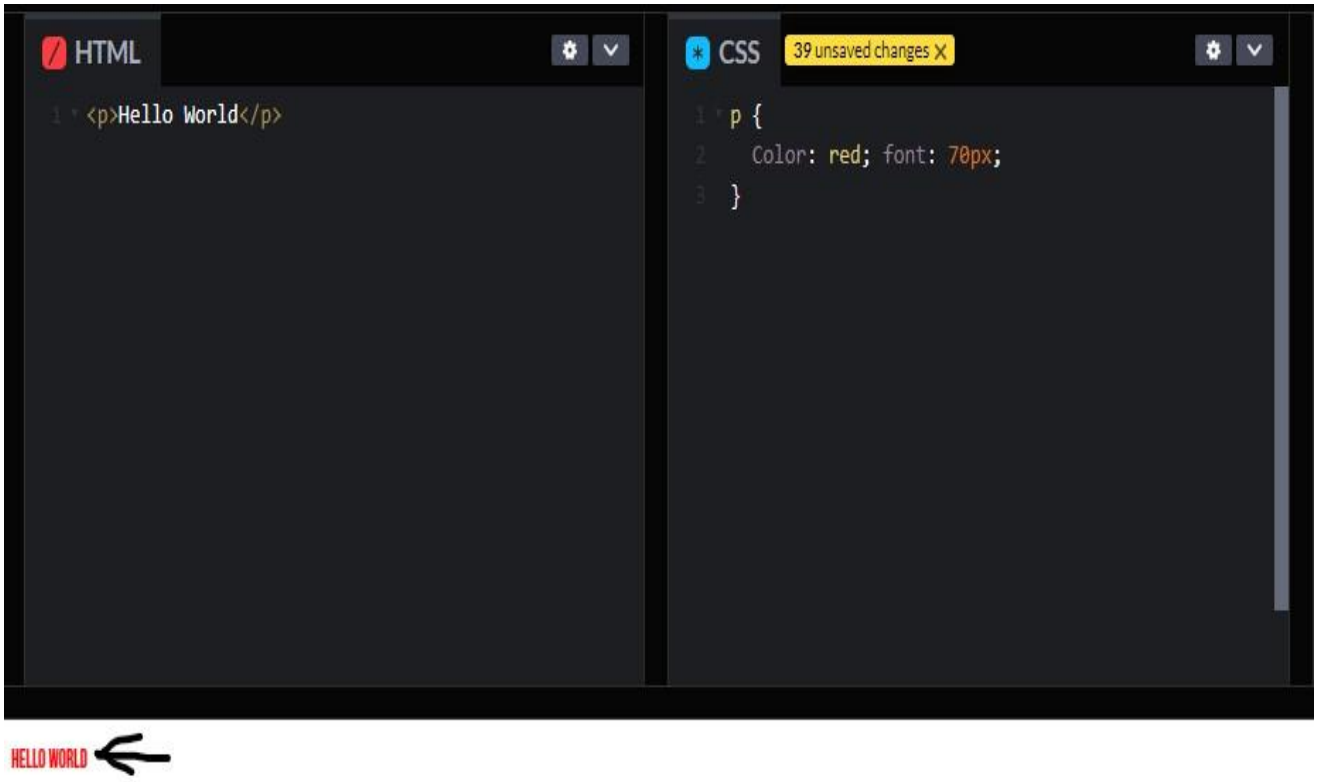
The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

For example, Let's have an Html Paragraph that says **"Hello World",** and we intend on styling it to look beautiful.

We will write out a code in Html on that and style it in CSS in https://codepen.io/pen/ ;



We can see that the text was changed from the initial Black colour to Red.

NB: When changing your colour in programming, make sure it is spelt 'Color" and not 'Colour'.

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

```
p {
  text-align: center;
  color: red;
}
```

Try it out in https://codepen.io/pen/ to get output.

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

Note: An id name cannot start with a number!

```
#para1 {
  text-align: center;
  color: red;
}
```

Note: Most Examples are gotten from w3Schools.

Read More on Selectors at
https://www.w3schools.com/css/css_selectors.asp

CSS Comments

CSS comments are not displayed in the browser, but they can help document your source code.

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment is placed inside the `<style>` element, and starts with `/*` and ends with `*/`:

Example

```
/* This is a single-line comment */
p {
  color: red;
}
```

We will talk about comments in HTML in the second unit of this book.

JavaScript

JavaScript is the world's most popular programming language.

JavaScript is often called the language of the Web.

JavaScript is easy to learn.

JavaScript is a cross-platform, object-oriented scripting language used to make webpages interactive.

In this section, we will cover 2 important areas in javascript;

- Variables
- Functions

Variables

Variables are containers for storing data (storing data values).

Take a Basket as an Example, you've got to store 5 items in it. The basket serve as the collector/containers of those 5 items, we can also compare that to a variable.

4 Ways to Declare a JavaScript Variable:

- Using `var`
- Using `let`
- Using `const`
- Using nothing

Examples

For Var:

```
Var x = 5;
Var y = 7;
Var z = x+y;
```

The output will give us 12

For Let

```
Let x = 5;
Let y = 7;
Let z = x+y;
```

The output will give us 12

For Const

```
Const x = 5;
Const y = 7;
Const z = x+y;
```

The output will give us 12

You'd notice that all of the variables listed above are defined, and the given values are added together to get the answers.

JavaScript Identifiers

All JavaScript **variables** must be **identified** with **unique names**.

These unique names are called **identifiers**.

Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).

Functions

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it). **(Definition from w3Schhols)**

JavaScript Function Syntax

A JavaScript function is defined with the `function` keyword, followed by a **name**, followed by parentheses **()**.

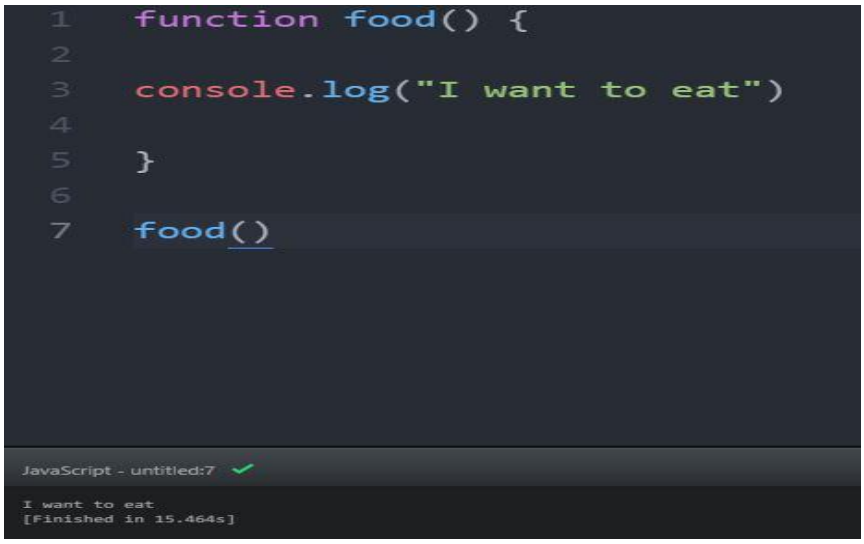
Function names can contain letters, digits, underscores, and dollar signs.

The parentheses may include parameter names separated by commas: **(parameter1, parameter2, ...)**

The code to be executed, by the function, is placed inside curly brackets: **{}**

Note: To include a space between two words in coding, underscores are used. Example, When we call out a function;

"Function my_day ()" we can see how an underscore was used to Separate 'my' and 'day'.



We can see from the above that, we created a function called "Food", we input some parameters and by using 'console.log' and called it out.

There are other interesting and important parts of **JavaScript** like, Statements, for loop, while loop, methods, and data types but we won't be covering them in this course. (Simple definitions will be given on them in the second unit of this book).

Practice

1. Write out the different tags of **HTML** and their works.

2. JavaScript is often called the Language of…?

3. What is **CSS** used for?

4. To start a paragraph in HTML, what tag would be the most suitable to use?

A. The 'p' tag

B. The 'h' tag

C. The 'br' tag

D. The 'a' tag

5. What is the 'a' tag used for?

A. To link to another webpage

B. To Link to an image

C. To start a new webpage

D. To run a JavaScript program

Part 2

In previous chapters, we defined what a website is, talked about HTML, CSS, and JAVASCRIPT.

We said majority of websites contains Those three languages and it is apt that we at least, have basic knowledge on them if we are ever going to build a website.

In this part of the tutorial, we will be going into advanced topics in web development.

We will cover:

HOSTING

DOMAIN

CMS

These three important topics are quite popular and are used make the creation of websites easier.

Hosting:

This simply means to store websites on a server so that it can be accessed on the internet.

Domain:

A domain is the name given to websites stored in a server.

CMS:

A content management system is a computer software used to manage the creation and modification of digital content. A CMS is typically used for enterprise content management and web content management.

Now let’s dive into section fully!

HTML

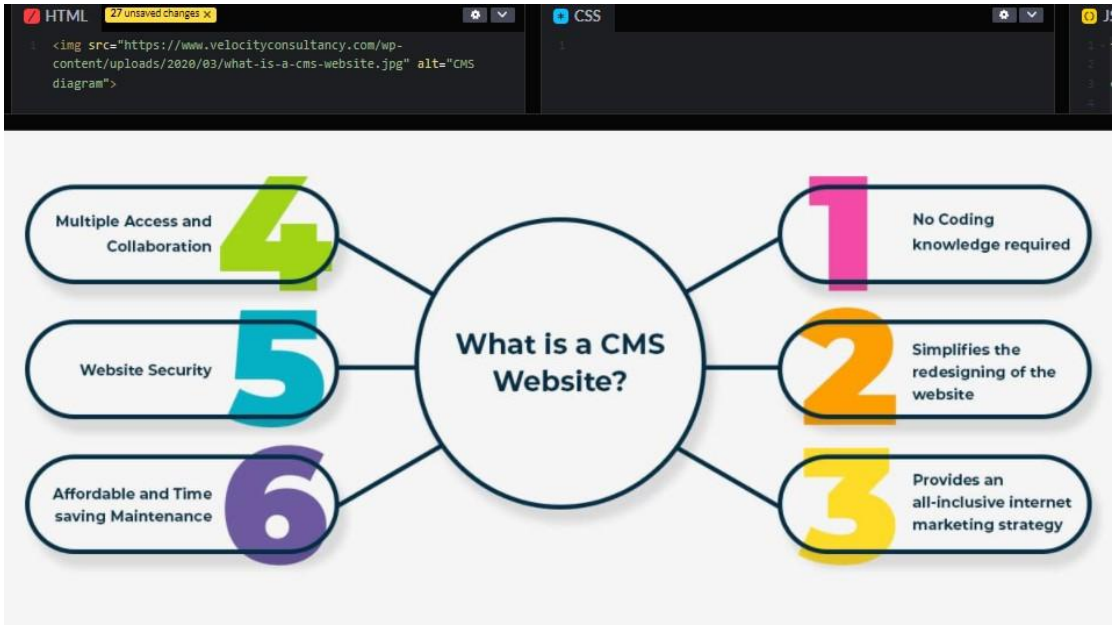
In unit 1 of this course, we talked about HTML and the tags it has. In this part, we will talk another set of HTML tags.

Note: You will learn more tags on HTML when you search for them. Make sure you work more with Google.

The first tag will be the image tag, represented with 'img'. We use it to input images into HTML projects/files.

Example:

<Img src="Link to the image" alt="What the image contains">



The ***SRC*** stands for the link to the particular picture you want to include in your project, the ***ALT*** explains the image in your file.

We can see how they were explained in the above diagram.

We have the 'SRC' showing us the link, and 'ALT' telling us that the diagram is that of a CMS.

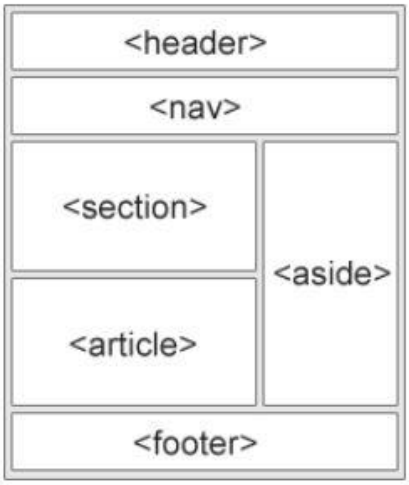
HTML SEMANTICS

What are Semantic Elements?

According to w3Schools, a semantic element clearly describes its meaning to both the browser and the developer.

Examples of **semantic** elements: `<form>`, `<table>`, and `<article>` - Clearly defines its content.

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



One important tag in HTML is the 'video tag' written as `<video></video>`

It is used to input videos in HTML files.

Example:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
</video>
```

The Importance of JavaScrIpt

JavaScript has become integral to the Internet experience as developers build increased interaction and complexity into their applications. Search engines, ecommerce, content management systems, responsive design, social media and phone apps would not be possible without it.

JavaScript is mostly used in web applications and mobile applications.

To be a good JavaScript developer, you've got to learn the languag along with some of it's libraries.

Examples of JavaScript libraries,

- **ReactJs**
- **jQuery**
- **D3.js**
- **Underscore.js**
- **Lodash** etc

What are JavaScript Libraries?

You can use JavaScript libraries for:

- **Dom manipulation**
- **Data Handling**
- **Animation**
- **Image effect**
- **Database**
- **Forms**
- **Data visualization in Maps and Charts etc.**

DOMAIN NAME - Introduction

A domain name refers to your website address. This is what users type in a browser's search bar to directly access your website. A domain name is unique and cannot be shared between different sites.

Just like the definition, domain names are used to identify, locate a website. Examples of domain names include, Google.com, Facebook.com, Netflix.com, etc.

Difference between Web design and web development

What is web design?

Web design is concerned with what the user actually sees on their computer screen or mobile device, and less so about the mechanisms beneath the surface that make it all work. Through the use of color, images, typography and layout, they bring a digital experience to life. Some common skills and tools that distinguish the web designer from the web developer are:

- Adobe Creative Suite (e.g., Photoshop, Illustrator) or other design software
- Graphic design
- UI design
- UX design
- Logo design
- Layout/format
- Placing call-to-action buttons
- Branding
- Wireframes, mock-ups, and storyboards
- Color palettes
- Typography

Note: Don’t worry about UI/UX, we won’t be covering that in this course.

What is web development?

Web development governs all the code that makes a website tick. It can be split into two categories—front-end and back-end. The front-end or client-side of an application is the code responsible for determining how the website will actually display the designs mocked up by a designer. The back-end or server-side of an application is responsible for managing data within the database and serving that data to the front-end to be displayed. As you may have guessed, it's the front-end developer's job that tends to share the most overlap with the web designer. Some common skills and tools traditionally viewed as unique to the front-end developer are listed below:

- HTML/CSS/JavaScript
- Frameworks (i.e., AngularJS, ReactJS, Ember)
- Web template design
- Libraries (i.e., jQuery)
- Git and GitHub
- On-site search engine optimization (SEO)
-

Back-end developers handle the business logic and data management on the back-end of an application. They write the APIs and routing that allow data to flow between the front and back end of an application. Programming languages and tools unique to back-end developers are listed below:

- Server-side programming languages (e.g., PHP, Python, Java, C#)
- Server-side web development frameworks (e.g., Ruby on Rails, Symfony, .NET)
- Database management systems (e.g., MySQL, MongoDB, PostgreSQL)
- RESTful APIs
- Authentication and security (e.g., OAuth, PassportJS)
- Servers (e.g., Linux, Apache, Express)

Web developers who possess a working knowledge across the frontend and backend of a technology stack are called full-stack developers.

Most web designers do not code

Web designers are responsible for the general look and feel of a website. They might use a visual editor like Photoshop to create images or an app prototyping and animation tool such as InVision Studio to design layouts and generate high-fidelity mockups. But none of these main responsibilities require coding.

Web developers do not create assets

Web developers are the programmers with the coding skills needed to add functionality to a website. They translate the designer's wireframes and mockups into code using HTML, CSS, and JavaScript. Web developers don't usually have to create the visual assets themselves such as the images behind buttons, color schemes, and fonts. They just need to use code to implement them into the page.

Both Skills actually work hand in hand, a web designer creates a a design for the website and the web developer brings out the imagination of the designer with codes.

CMS

CMS stands for content management system. CMS is computer software or an application that uses a database to manage all content, and it can be used when developing a website.

The most popular CMS is called **Wordpress** and we will be talking about that extensively in this Unit.

CMS sometimes makes the work of a web developer easier, you tend to do less of coding or no coding at all when using a CMS to build a website.

WORDPRESS

"WordPress is a factory that makes webpages"; it stores content and enables a user to create and publish webpages, requiring nothing beyond a domain and a hosting service. WordPress has a web template system using a template processor.

It is the world most popular CMS and almost every web developer use it one way or the other.

Popular Hosting Services that uses Wordpress includes:

- **Godaddy**
- **Bluehost**
- **Hostinger**
- **Kinsta**
- **SiteGround**
- **Flywheel**
- **DreamHost**
- **WIX**

We will talk about Go daddy in this part and explain how it can be used to create simple websites.

HOSTING - GoDaddy

GoDaddy is a beginner-friendly website builder that uses Artificial Design Intelligence (ADI) to make creating your website easier and faster than ever before. It uses information you enter to create a customized site in a matter of minutes, making it perfect for beginners who want to publish a simple site online.

At some point in 2017, it was rated the best website builder and hosting service.

It uses simple drag and drop techniques to give developers easier ways to build simple, fast and unique websites.

Although, experts in web developments can use it to create harder tasks by simply joining knowledge of codes and tools n the website together.

We will run through Godaddy in the next few pages and finally show you how to build easy websites with them.

BUILDING A WEBSITE WITH GODADDY

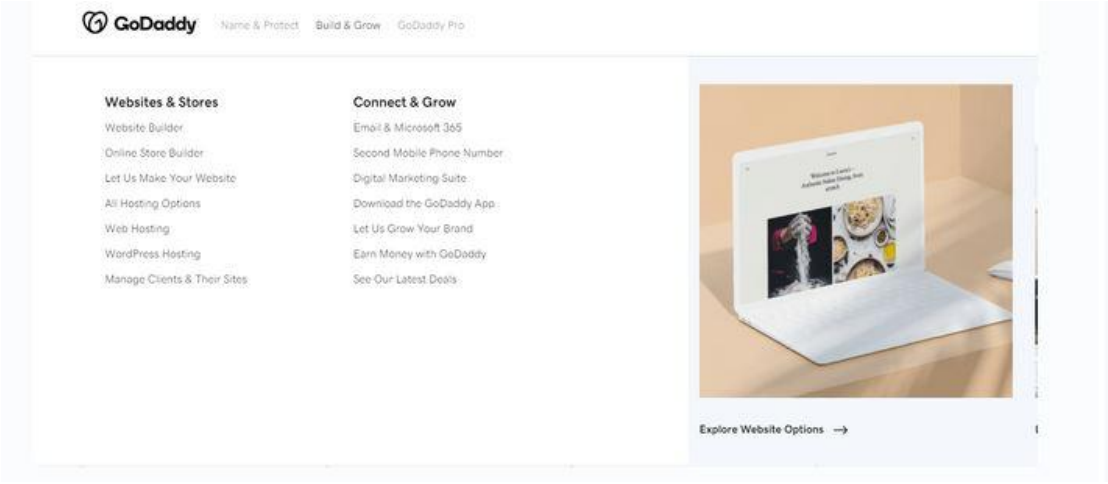
Steps:

1. **Choose your site type.**
2. **Start your free trial.**
3. **Select your industry and website title.**
4. **Start editing your site: pick your theme.**
5. **Customize your design: add your own images.**
6. **Add your own text.**
7. **Add new sections.**
8. **Add pages.**
9. **Add social links**
10. **Connect a Custom Domain**
11. **Preview and Publish Your Site**

1. Choose Your Site Type

Go to the top

The very first thing you need to do is **decide which type of website you want to build**. If you want to sell online, you'll need to select the online store option. If you won't be accepting payments through your website, then you should select the website builder option.

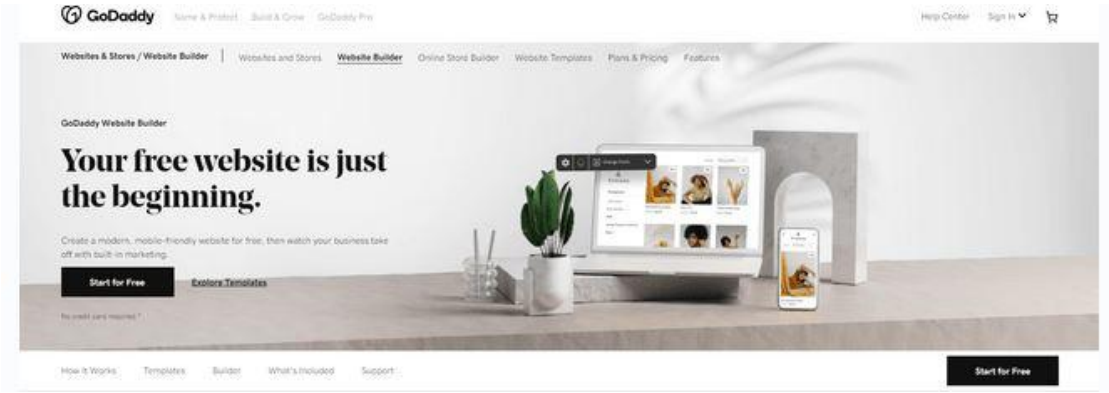


In this step-by-step guide, we'll be showing you **how to build a website** rather than an online store, although a lot of the steps are similar.

2. Start Your Free Trial

GoDaddy has a **one-month free trial,** which is definitely worth making the most of. You can practise these steps and get your site ready on the free trial, then choose to upgrade when your trial draws to a close.

That means you can learn how to use GoDaddy without spending a dime!



It's easy to start your free trial – just click any of the **Start for Free** buttons on GoDaddy's website, then create an account. It's totally free, and **you don't need to enter any credit card details to sign up.**

All you need is an email and password, and you're good to go!

GoDaddy's free trial is extremely generous, and gives you plenty of time to make sure it's the perfect builder for you. **There's no risk and no pressure to sign up** once the trial finishes – you only pay if you loved using GoDaddy!

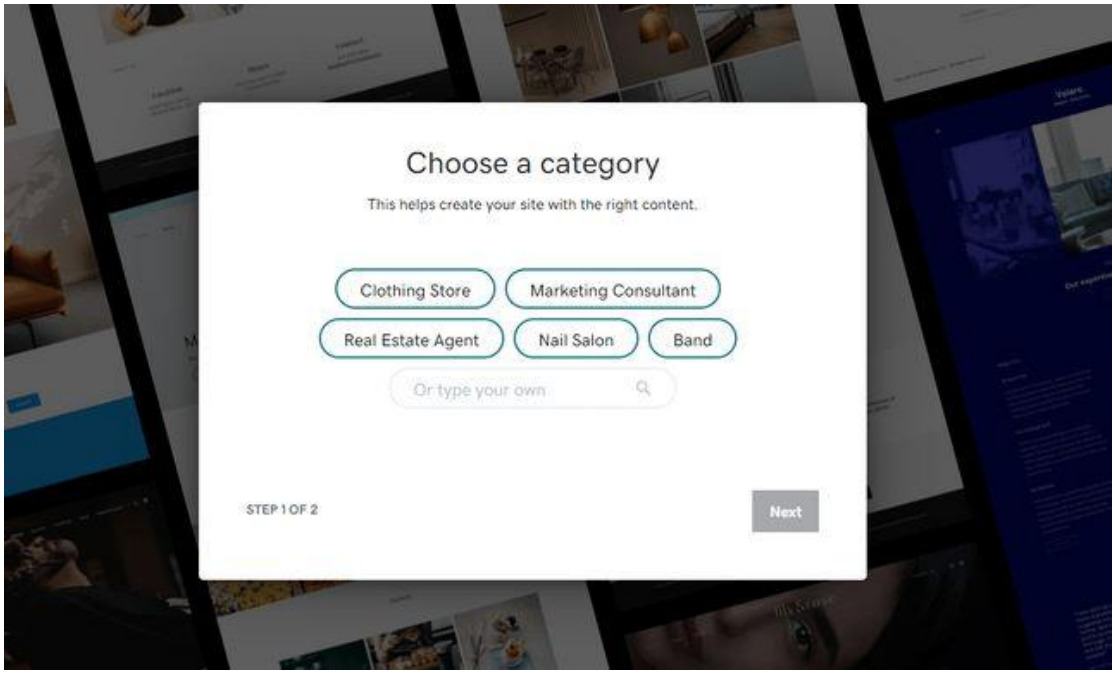
For now, though, simply start building for free and come back to pick your price plan once your trial is over – simple!

Before you move on, make sure you've:

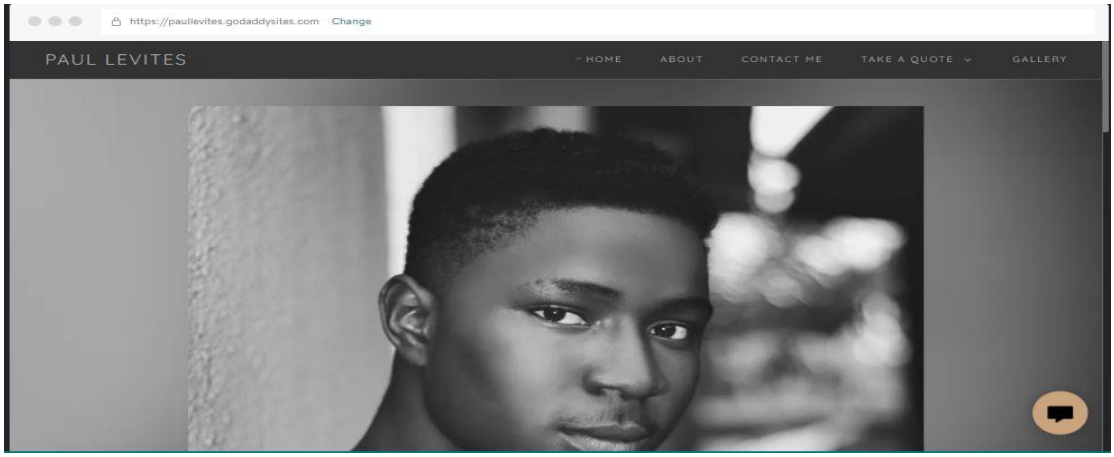
- Clicked Start for Free
- Created your GoDaddy account

3. Select Your Industry and Website Title

Once you've signed up, you'll see this page in front of you, encouraging you to get started. This is the first step to actually building your site – and it couldn't be easier.



All you need to do is enter the industry your site comes under, and then the name of your website. As you do so, GoDaddy automatically creates an example page based on the basic information you've entered.

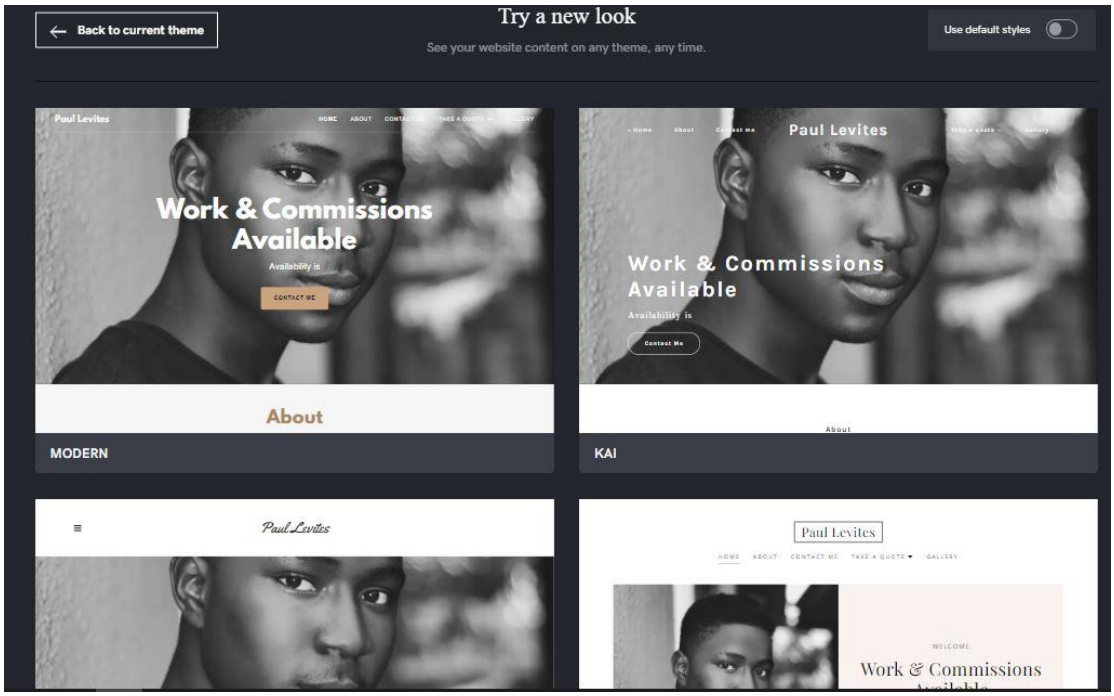


We wanted a website under the industry "Portfolio Website", titled "Paul Levites". GoDaddy **instantly gave us this simple, stylish theme,** centered around what we'd told it so far. You can play around with different industries and site titles to find one you're happy with – once you're satisfied, just click the Continue button.

4. Start Editing Your Site: Pick Your Theme

Now you've got your base template, you can begin to edit it so that it's more unique and in-line with your own brand. **The first thing to do is to pick your theme.** Your theme changes the design and layout of your site, so it's important to choose one you like!

It's easy to change your theme – just click that Theme button, and you'll find tons of different themes for you to preview.



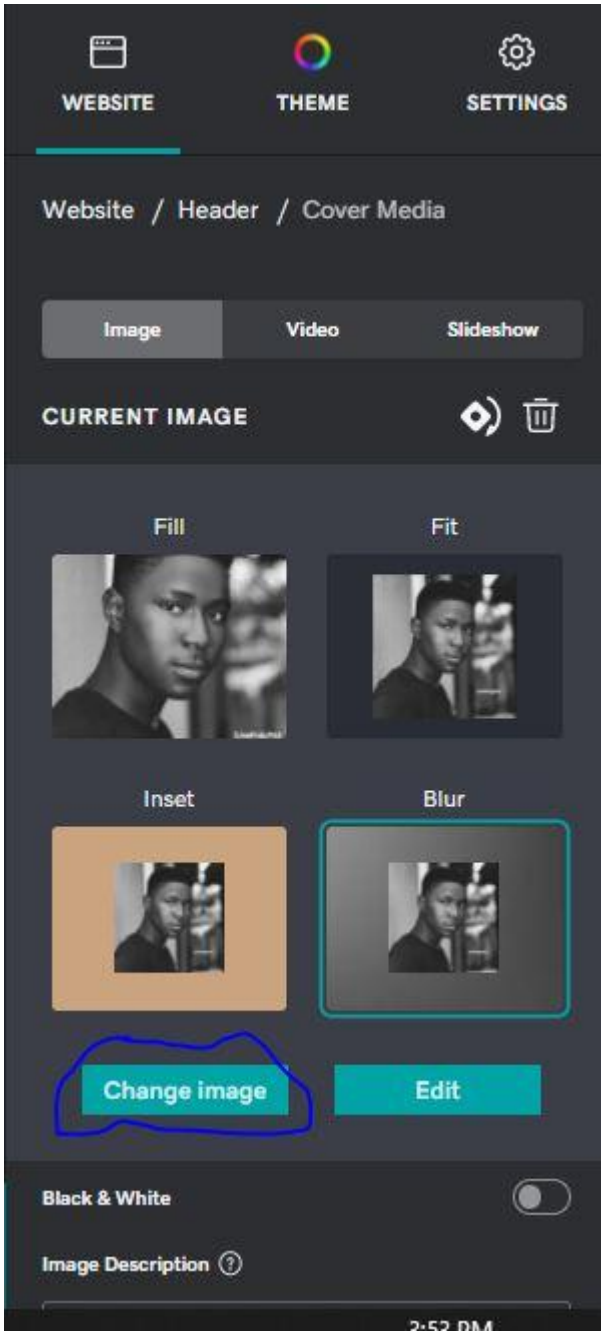
5. Customize Your Design: Add Your Own Images

Go to the top

Images are super important for your website's success. Your theme comes with stock images already in place – but you probably want to add your own!

However, you might want to replace this stock image with a different one – maybe even photos you've taken yourself! To do this, **click the Update button on the right-hand side of the image.**

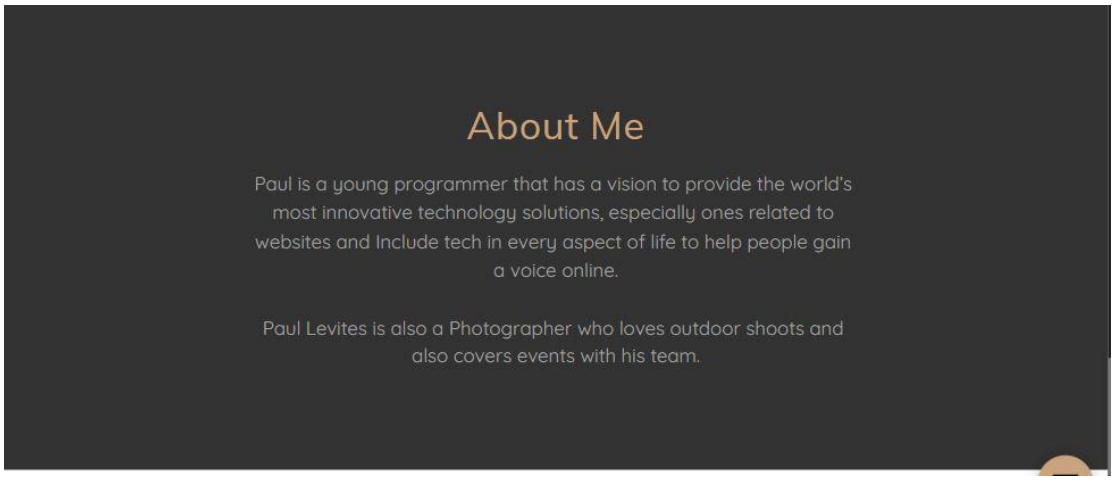
Here, you can upload your own images, browse free stock photos, or even connect to social media to add images at the click of a button.



Change the image by clicking on that ‘Change image’ Icon.

6. Add Your Own Text

This is a simple but crucial step to customizing your website. Content is king in the website world, so you need to replace that stock text and put in your own killer content!



It's easy to edit the stock text already on your page, but you're more limited when it comes to moving or even adding new text boxes. You can't just click and drag text boxes around, and you can't add a single text box to your page.

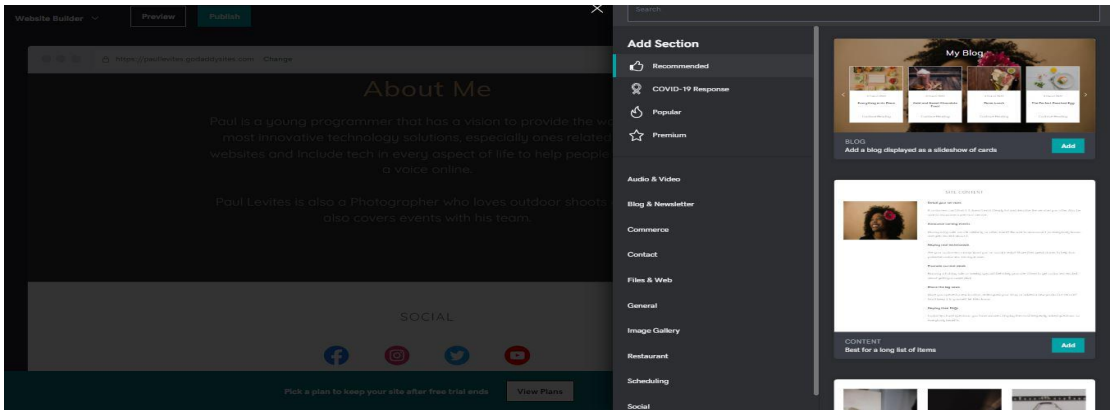
Instead, you need to add a new section (we'll get to that next) and change the stock text that comes with it.

7. Add New Sections

You'll notice your website already has four sections included – Header, About Us, Contact Us, and Footer. One of the best ways to make your site unique is by **adding new sections** – with this handy tool, you can add photo galleries, blogs, videos, menus, and more.

When you're editing your page, you'll notice blue buttons appearing with plus signs on them – hover over one of these, and the words Add Section will appear.

Click this button, then choose what type of section you'd like to add from the long list that pops up:



You can add up to 20 sections on each page, which should be plenty. The only sections you can't add are more Headers or Footers, but other than that, you're free to choose whatever you like from that impressive menu.

If you want to add lots of new content, it's best to add a Content section, as this means you can write as much as you like!

If you add a section by accident or decide you no longer need it, **you can easily delete sections** by clicking on them, then hitting the red Delete button at the bottom of the side bar that appears on the right.

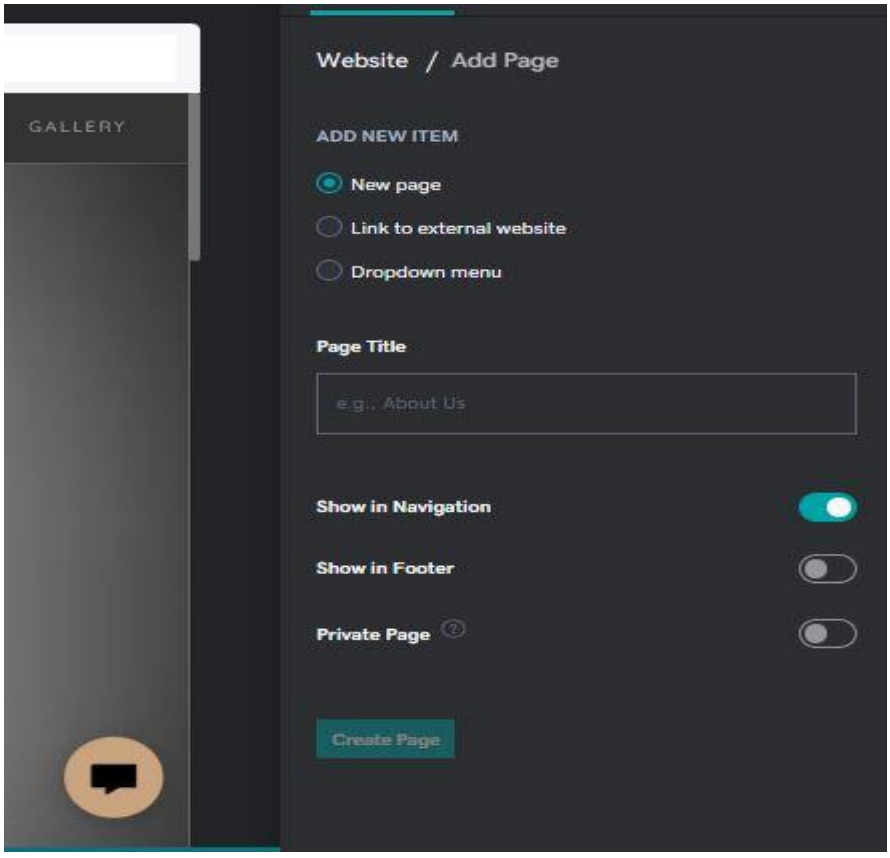
Before you move on, make sure you've:

- Added new sections to your site
- Deleted any sections you don't need

8. Add Pages

Some website builders only let you build one-page websites, while others let you create complex sites with hundreds of pages. So how does GoDaddy work? Well, GoDaddy sits modestly in between these extremes – **you can add up to 50 pages** on your site, or keep it simple with a single page.

Adding pages is easy – simply go to your main website menu, then click the '+' button next to Site Navigation. You'll see a list of your existing pages underneath.



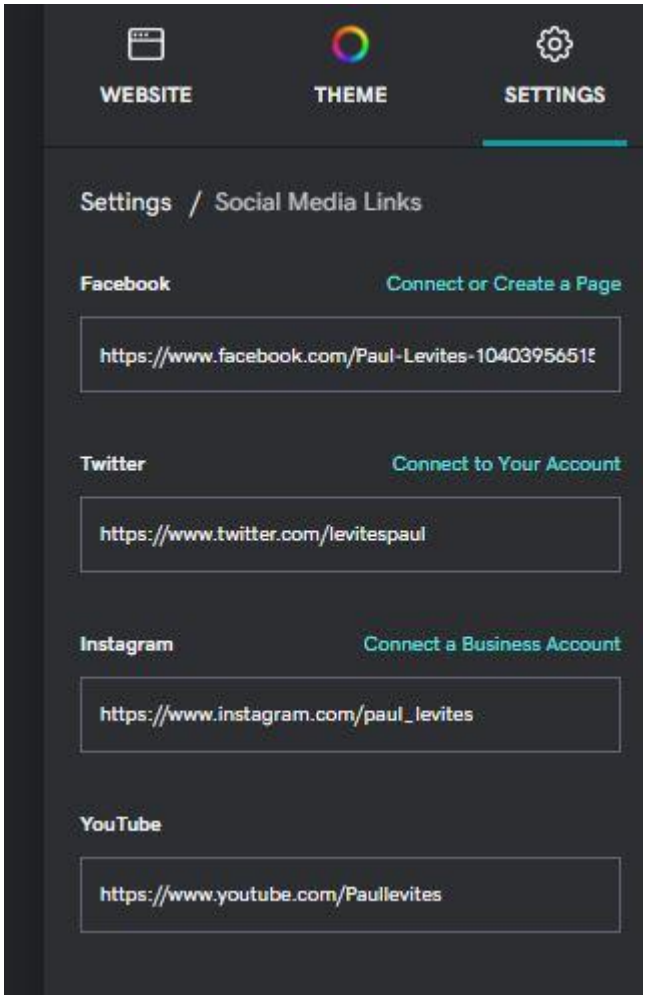
All you need to do now is give your new page a title, select the settings you want – for example, if it should appear in the navigation menu – and **click Create Page.**

Top Tip! Keep your page title short, as it will show up in your navigation menu. Long, complicated page names will make your menu cluttered and confusing.

Once you've created your shiny new page, you can start breathing life into it by adding sections to fill it up. Whether you want videos, contact forms, or even to embed custom code on the page, it's a blank canvas for you to play with.

9. Add Social Media to Your Site

Stay connected by integrating social media with your website. You can add buttons linking to your various social media accounts, making it easy for visitors to find and follow you. Simply **add a new section** and select "Social". You can then link out to your social accounts by putting in the appropriate links.



10. Connect a Custom Domain

You're nearly there! The last thing to do before you can publish your site is to connect your very own domain name.

You can either connect an existing domain name if you already own one, or buy a new domain from GoDaddy.

If you need to buy a new domain, GoDaddy gives you a few suggestions based on the name of your site, or you can search for one yourself. You will have to pay for your new domain, but it shouldn't be too expensive.

11. Preview and Publish Your Site

It's time! You're ready to publish and share your site with the online world. In order to avoid any surprises, you should **always preview your site as you build** – but now it's time for the final check. Get someone else to look around your site as a test run before hitting the big green button.

Look out for broken links, images that won't load, or sections you meant to delete and forgot about. Once you're totally happy with your website, go ahead and hit that ultra inviting Publish button.

Congratulations – your site is online!

Before you move on, make sure you've:

- Previewed your site
- Tested your website to make sure there aren't any mistakes, flaws, or broken links
- Hit publish!

As you can see, building a website with CMS isn't hard , it is easy and interesting. Now you can build a simple website.

Note: There are other effective ways of building websites and they all have to do with codes.

Here are the list of things you need to learn to improve your web development skills;

1. Learn HTML
2. CSS
3. JavaScript
4. Reactjs
5. PHP
6. Python

Each of these languages have their specific function and we will just take little notes on them.

HTML: Deals with the structure of the page

CMS: Styles the page

JavaScript: Gives web pages interactive elements that engage a user.

ReactJs: React is a declarative, efficient, and flexible JavaScript library for building user interfaces.

PHP: With PHP, you have the freedom of choosing an operating system and a web server. Furthermore, you also have the choice of using procedural programming or object-oriented programming (OOP), or a mixture of them both.

Python: This a high programming language that deals with the back-end and also the front-end of a website.

Note: One way to improve your skill, is by constant practice, constant thinking, and thorough research.

Practice:

Build a portfolio website using the Godaddy web hosting service

Important topics covered:

HTML, CSS, JavaScript

Hosting

CMS

Godaddy web hosting

Outro

More books on programming will be released soon stay alert. Follow me on all social media platforms to get more tips on web development and programming. (Link on my website)

Thank you for your time, and I wish you a successful programming Journey!