Joshua Bilsland

The Odin Project Notes

A collection of notes

Contents

[Foundations – Introduction 2](#_Toc115601754)

[How This Course Will Work 2](#_Toc115601755)

[Introduction to Web Development 3](#_Toc115601756)

[Motivation and Mindset 3](#_Toc115601757)

[Asking For Help 4](#_Toc115601758)

[Join the Odin Community 4](#_Toc115601759)

[Foundations – Prerequisites 5](#_Toc115601760)

[Computer Basics 5](#_Toc115601761)

[How Does the Web Work? 5](#_Toc115601762)

[Installation Overview 6](#_Toc115601763)

[Installations 6](#_Toc115601764)

[Text Editors 7](#_Toc115601765)

[Command Line Basics 7](#_Toc115601766)

[Setting Up Git 7](#_Toc115601767)

[Foundations – Git Basics 8](#_Toc115601768)

[Introduction to Git 8](#_Toc115601769)

[Git Basics 9](#_Toc115601770)

[Foundations – HTML Foundations 10](#_Toc115601771)

[Introduction to HTML and CSS 10](#_Toc115601772)

[Elements and Tags 11](#_Toc115601773)

[HTML Boilerplate 12](#_Toc115601774)

[Working with Text 14](#_Toc115601775)

[Lists 15](#_Toc115601776)

[Links and Images 17](#_Toc115601777)

[Commit Messages 18](#_Toc115601778)

[Project: Recipes 19](#_Toc115601779)

[Foundations - CSS Foundations 19](#_Toc115601780)

[CSS Foundations 19](#_Toc115601781)

[Inspecting HTML and CSS (Chrome) 28](#_Toc115601782)

[The Box Model 30](#_Toc115601783)

[Block an Inline 32](#_Toc115601784)

[Foundations - Flexbox 33](#_Toc115601785)

[Introduction to Flexbox 33](#_Toc115601786)

[Growing and Shrinking 34](#_Toc115601787)

# Foundations – Introduction

## How This Course Will Work

* “The Odin Project is an open-source community dedicated to providing the best information sources to take you from zero to a full-stack developer”
* Will go over the basics of internet, Git, GitHub, HTML, CSS, JS, back-end technologies, etc
* Lessons will contain questions that should be answered before moving on
* After the foundations course, the Full Stack JavaScript and Full Stack Rails paths can be taken
* A collection of the best sources that could be found by the community
* Some things made by the community themselves

## Introduction to Web Development

* Web development is project-focused and involves collaborating with a team that helps to co-ordinate the client’s needs into the end product.
* Front end:
  + What the website visitors see
  + The presentation of content and user interface elements
  + Use of HTML, CSS, and JavaScript
* Back end:
  + “The guts of the application”
  + Stores and serves data to ensure the front end has what it needs
  + Uses languages such as Java, Python, and Ruby
* Full Stack:
  + Developers that are comfortable working with both the front and back ends
  + The Odin Project focuses on teaching full-stack development
* Can work for large tech companies, startups, as a freelancer, or as a consultant
* <https://web.archive.org/web/20160925155912/http://www.happybearsoftware.com/how-to-get-a-programmer-job.html>
* Can help add to the project (Will be good for CV)
* <https://github.com/kamranahmedse/developer-roadmap>

## Motivation and Mindset

* Someone with a ‘fixed mindset’ believes that if they don’t get something on their first attempt, they never will
* Someone with a ‘growth mindset’ believes they can get better at anything with effort and persistence
* Intelligence is not fixed and can be developed
* ‘Focus mode’ is when your mind is focused on learning, reading, watching videos, or working on a project
* ‘Diffuse mode’ is when you are not actively learning and is the state where your mind starts to connect what you have been learning (This is why when you are falling asleep you might have a breakthrough on a problem you were stuck on)
* This is why taking a break when you are really stuck on a problem is good as you can usually come up with a solution
* Teaching things to others is a great way to solidify what you have learned

## Asking For Help

* It is essential to ask well formed questions to make it easier for people to help
* Always provide code and the surrounding context
* Ask for help, not the solution itself
* <https://medium.com/@gordon_zhu/how-to-be-great-at-asking-questions-e37be04d0603>
* Text

  Description automatically generated<https://xyproblem.info/> - When people ask about their attempted solution rather than their actual problem.
* <https://stackoverflow.com/help/how-to-ask>
* https://slash7.com/2006/12/22/vampires/

Knowledge Check Answers:

1. Context or code
2. Where someone asks for help with Y to assist their solution with X when they really should be using an alternative X. Asking about their attempted solution rather than their actual problem.
3. Doesn’t google things/read docs, doesn’t ask specific questions, asks common questions that have already been answered before

## Join the Odin Community

* Working and collaborating with other people is an important part of working as a web developer
* Use the #TheOdinProject
* Community discord
* Rubber duck debugging – Go through the code line by line
* Use backticks in discord to show code different from the rest of the text.
* Use three backticks above and below code for multiple lines of code
* Specify the language after the three backticks to add colour

# Foundations – Prerequisites

## Computer Basics

* <https://edu.gcfglobal.org/en/computerbasics/what-is-a-computer/1/>
* <https://edu.gcfglobal.org/en/computerbasics/understanding-operating-systems/1/>
* <https://edu.gcfglobal.org/en/computerbasics/understanding-applications/1/>
* <https://edu.gcfglobal.org/en/basic-computer-skills/open-source-vs-closed-source-software/1/>
* <https://edu.gcfglobal.org/en/techsavvy/taking-screenshots/1/>
* <https://edu.gcfglobal.org/en/techsavvy/password-tips/1/>

Knowledge Check Answers:

1. Windows is an operating system
2. Open-source software is software where the source code is publicly accessible and usable. Closed source software is the opposite.
3. Showing error messages to show IT support and showing evidence of code for exams.
4. Weak: password123 🡪 Strong: P4$$w0rD231

## How Does the Web Work?

* <https://www.youtube.com/watch?v=eHp1l73ztB8>
* <https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_does_the_Internet_work>
* <https://www.youtube.com/watch?v=7_LPdttKXPc&t=46s>
* <https://www.youtube.com/watch?v=BrXPcaRlBqo>
* <https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/How_the_Web_works#Clients_and_servers>

Knowledge Check Answers:

1. A network is two or more computers that are connected to share data.
2. A global network of networks
3. A unique identifier for a computer using the Internet Protocol
4. A router connects networks together. It allows devices to connect to the internet.
5. A client is a device that is requesting and receiving data from a server and using its services.
6. A server is a device that stores and provides data, services, or programs to client devices.
7. A web page is a document for the WWW, viewed in a web browser.
8. A web server is a server that stores websites and processes HTTP requests
9. A web browser is software that is used for accessing websites/the WWW
10. Software that searches the WWW using search queries
11. A DNS request is a request to a **Domain Name Server** asking it what IP address is associated with a given domain name
12. Google Chrome is the browser I use the most
13. You send a query (a set of keywords) to google who will then go through their database of webpages and find pages that are the most relevant to what you searched. You can then click on one of the pages returned to go through the process of loading the page (DNS request, request to web server, response from web server, website sent to client, web page loaded in client’s browser)

## Installation Overview

* Dual-booting is where you install two operating systems on your computer, which can give you the option to boot either OS when your computer first starts up.
* A virtual machine is an emulation of a computer that runs within your existing OS.

## Installations

* The Odin Project recommends using either a virtual machine or dual-boot to use Linux as it doesn’t support windows.

## Text Editors

* “A good text editor can help you write better code with real-time code checking, syntax highlighting, and automatic formatting.”
* Microsoft Word and Libre-Office Writer cannot be used as they store information about how to display the text on the screen which means interpreters unable to execute the file as code.
* Code editors are tools that can take a text file an provide features such as plugins, syntax highlighting, auto-closing of brackets and braces, and linting. Visual Studio Code (VSCode) is the most popular choice.

## Command Line Basics

* The command line interface (CLI) is where you can enter commands that your computer will run.
* $ is used to show that what follows is a command that should be put into the terminal.

Knowledge Check Answers:

* A command line is an interface that lets the user enter commands which will be carried out by the computer
* You can open it by clicking the icon on your GUI OS or by pressing CTRL + ALT + T (on linux)
* You can navigate to a directory by entering $ cd /file/path
* cd on its own will take you to your home directory
* $ cd .. will take you back up a directory
* $ pwd will display which directory you are in and the path to get there
* $ ls is used to display the contents of the directory you are in
* $ mkdir directoryName is used to create a new directory
* $ touch fileName is used to create a new file
* $ rm fileName will delete the named file 🡪 rmdir directoryName will delete the named directory
* $ mv oldfile.txt newfile.txt will rename a file 🡪 $ mv oldDirectory newDirectory

## Setting Up Git

* Git is a popular version control system.
* GitHub allows you to upload code/files using Git and manage your code using a web interface.

# Foundations – Git Basics

## Introduction to Git

* Git is a version control system – “Git is like a really epic save button for your files and directories”.
* A save in Git records differences in the files and folders and keeps a historical record of each save.
* Enables you to review how your project grows and restore file states from the past.
* GitHub is a remote storage facility on the web for all your projects.
* <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
  + Local Version Control Systems – A local database which stores every file change as a patch. The patches are then put together to re-create a file at a given point in time.
  + Centralised Version Control Systems – All the files and their versions are stored on a central VCS server. Multiple people can then save to and retrieve from the server.
  + Distributed Version Control Systems – A clone of the repository is stored locally on each developer’s computer so that if the server goes down, the repository can be recovered from the mirrored copies stored by all the contributors.
* <https://www.youtube.com/watch?v=8oRjP8yj2Wo>
* <https://www.youtube.com/watch?v=1h9_cB9mPT8&t=13s>
* <https://github.com/TheOdinProject/curriculum>

Knowledge Check Answers:

1. Git is a version control system (VCS)
2. A text editor saves all the words in a single file. There are no other records of the file recorded so you would need to make multiple copies of the file to keep track of how it has changed. Git records differences in the files and folders and keeps a record of these ‘saves’.
3. Git works at a local level.
4. GitHub works at a remote level.
5. An individual developer can use it to show off their projects to employers as well as make it easier to work on files from different computers. It can make bug fixing easier as you can go back to previous versions to see if they contain the same bug. Branches can be used to work on new files/features without affecting the main branch.
6. Branches allow developers to make changes without stepping on each other’s code. Branches can also be tested before merging them to the main branch to find bugs and problems first. Git will alert developers when there is a merge conflict and tell them where it/they occurred.

## Git Basics

* ‘git clone’ is used to clone the contents of a repository to the current directory.
* ‘git remote’ gives you the URL of the repository you are using.
* ‘git status’ displays the state of the working directory and the staging area.
* ‘git add fileName’ will add a file to the staging area. The staging area is part of the two-step process for making a commit in Git.
* ‘git commit’ makes a ‘save point’ in the repository.
* ‘git log’ shows a log of all previous commits, who made them, message given to the commit, etc.
* ‘git push’ updates a remote branch with local commits.

Knowledge Check Answers:

1. A new repository can be made on GitHub by clicking the ‘new’ button
2. The ‘git clone’ command can be used to copy a repository to your local machine
3. ‘Origin’ is the default name of your remote connection
4. origin refers to the remote
5. main refers to the branch you are pushing to
6. You add files to the staging area and the use commit to commit the changes
7. git status
8. ‘git add -A’ or ‘git add fileName’
9. git commit -m “message”
10. git push
11. git log

# Foundations – HTML Foundations

## Introduction to HTML and CSS

* HTML is the raw data that a webpage is built out of (text, links, cards, lists, buttons, etc).
* CSS is what adds style to those plain elements. HTML puts information on a webpage, but CSS positions that information, gives it colour, changes the fonts, etc.
* HTML is a markup language, CSS is a stylesheet language, and JavaScript is a programming language.
* <https://brytdesigns.com/html-css-javascript-whats-the-difference/>

Graphical user interface, website

Description automatically generated

Knowledge Check Answers:

1. HTML stands for HyperText Markup Language. CSS stands for Cascading Style Sheets
2. You would use HTML to add blocks of text to a website.
3. You would use CSS to change the font and background colour of a button
4. HTML is for creating the content of a website. CSS is for styling that content/those elements. JavaScript is used to alter the HTML and CSS of the website.

## Elements and Tags

* Almost all elements on an HTML page are just pieces of content wrapped in opening and closing HTML tags.
* An opening tag tells the browser ‘this is the start of an HTML element’. They are comprised of a keyword enclosed in angle brackets. An example is <p>.
* A closing tag tells the browser where an element ends. They are similar to an opening tag but have a forward slash before the keyword. For example, </p>.
* Some HTML elements do not have a closing tag. They are known as void elements or self-closing elements.
* <https://www.youtube.com/watch?v=LGQuIIv2RVA&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT->

Knowledge Check Answers:

1. A HTML tag is used to mark the start and end of an HTML element.
2. A HTML element is made up of an opening tag, content, and a closing tag.

## HTML Boilerplate

* All HTML documents have the same basic structure or boilerplate.
* index.html is the name for the main page of the website. It is the default page shown on a website if no other page is specified when a visitor requests the site. This name should always be used for this page since it is what web servers will look for by default.
* Every HTML page starts with a doctype declaration. It is used to tell the browser what version of HTML it should use to render a document. <!DOCTYPE html> will use the most recent version of HTML.

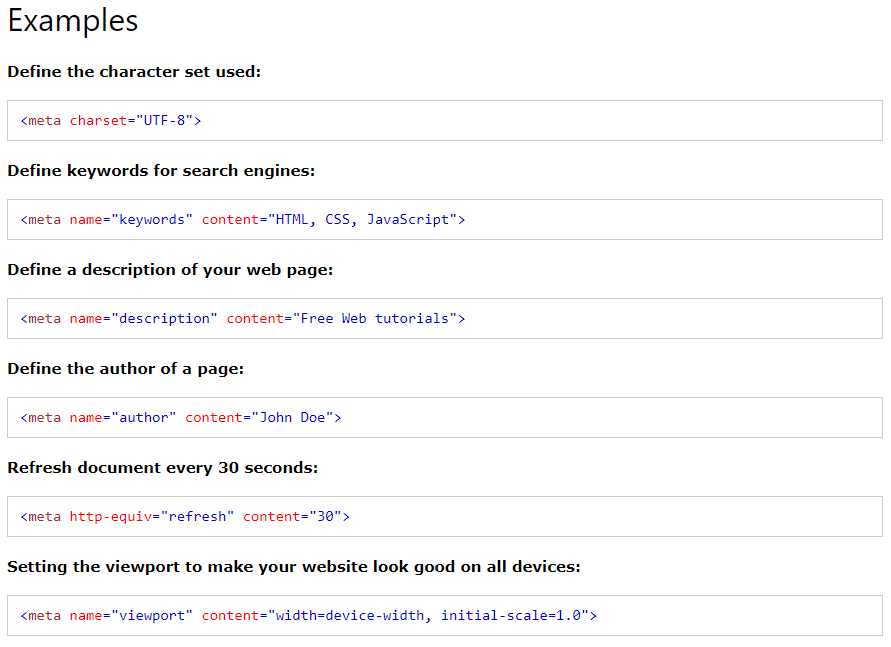
Graphical user interface, application, Word

Description automatically generated

* After declaring the doctype, the next thing to do is to provide an <html> element. This is called the root element of the document because all other elements in the document will be a descendant of it.

Text

Description automatically generated

* The lang attribute specifies the language of the text content in that element. It is mostly used to improve the accessibility of the webpage. Screen readers can adapt based on the lang attribute to change things such as pronunciations.
* The <head> element is used for storing meta data about a webpage. It should not contain any element that displays content on the webpage.
* Setting the encoding is important because it ensures that special symbols and characters from different languages will be displayed correctly.
* The <title> element should be included in the head of an HTML document. It is used to give a webpage a title which is displayed in a browser’s tab.

Text

Description automatically generated

* The <body> element is the final element to complete the HTML boilerplate. This is where all the content that will be displayed to users will go.

Text

Description automatically generated

* VSCode has a built-in shortcut which will generate a boilerplate. This can be used by entering ! on the first line of a blank HTML document.
* <https://www.youtube.com/watch?v=V8UAEoOvqFg&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&t=93s>
* <https://validator.w3.org/>

Knowledge Check Answers:

1. The doctype declaration is used to tell the browser what version of html it should use to render the file in.
2. The HTML element is the root element which every other element will be a descendant of.
3. The head element is where all the meta data about the website is stored.
4. The body element is where all the content of the website is put.

## Working with Text

* When a browser encounters lines of text in an HTML file, it will compress them together on a single line. To create paragraphs in HTML, we use the <p> tag.
* There are six different levels of headings starting from <h1> to <h6>.
* The <strong> tag is used to make text bold. Some screen readers will change how they say the text if it is a <strong> element.
* The <em> tag is used to make text italic. This can also affect how screen readers will say the text.
* <strong> and <em> elements can be used on text that is part of a <p> element.
* <b> can be used instead of <strong> and <i> can be used instead of <em> however it is bad practice. They do change how the text looks, but they do not have any semantic meaning for screen readers.
* Indentation is used to show parents, children, and sibling relationships between different elements. It helps improve readability.
* HTML comments are not visible to the browser but let you leave comments to help explain different parts of the file (similar uses to comments in programming). They start in <!-- and end in -->.
* <https://www.youtube.com/watch?v=yqcd-XkxZNM&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&index=4>
* <https://www.youtube.com/watch?v=gW6cBZLUk6M&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&index=5>

Knowledge Check Answers:

1. A paragraph can be created by using the <p> tag
2. A heading can be created by using one of the heading tags, <h1> to <h6>
3. There are six different heading levels. They have different sizes and levels of importance. <h1> being the most important.
4. <strong> should be used to make text bold and show it is important
5. <em> should be used to make text italicised and to add emphasis.
6. An element is the parent of any elements that are nested within it.
7. If two elements are at the same level, they are siblings
8. Comments are created by placing text between <!-- and --> .

## Lists

* An unordered list can be created by using the <ul> tag. Items can be added by using the <li> tag. The items should be placed between the <ul> tags. An example can be seen below.

Graphical user interface

Description automatically generated

* Graphical user interface, text

  Description automatically generatedAn ordered list can be created by using the <ol> tag. Items can be added by using the <li> tag. The items should be placed between the <ol> tags. An example can be seen below.

Knowledge Check Answers:

1. <ul> is used to create an unordered list
2. <ol> is used to create an ordered list
3. <li> is used to create list items

## Links and Images

* The <a> tag is used to define an anchor/hyperlink element.
* An HTML attribute gives additional information to an HTML element and always goes in the element’s opening tag. An attribute is made up of a name-value pair.
* The href (hyperlink reference) attribute is added to an anchor element to tell it the destination it should go to when clicked.
* Absolute links are used to link to pages on other websites on the internet. An example is “https://www.theodinproject.com/about”.
* Relative links are used to link to other pages without our own website. An example is “about.html”.
* Normally, index.html should be at the root directory. All other HTML files should be in their own directory.
* “./” should be prepended to a relative link to specify that it should start looking for the file/directory relative to the current directory.
* The <img> tag is used to display an image. It doesn’t have a closing tag. It embeds an image into the page using a src attribute which tells the browser where the image file is. It can embed an image using bot absolute and relative paths.
* To go up a directory (for the src attribute) we use ‘../’.
* The alt attribute is used to add text to store with an image. It will be displayed if the image cannot be loaded. Screen readers will use this alt text to describe the image to its users.
* <https://www.youtube.com/watch?v=tsEQgGjSmkM&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&index=6>
* <https://www.youtube.com/watch?v=0xoztJCHpbQ&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&index=7>
* <https://www.youtube.com/watch?v=ta3Oxx7Yqbo&list=PL4-IK0AVhVjM0xE0K2uZRvsM7LkIhsPT-&index=8>
* When naming files, avoid using spaces and keep the names in lowercase.
* Have separate directories for your pages, images, css, js, etc.
* <https://www.internetingishard.com/html-and-css/links-and-images/#image-formats>
* JPG images are designed for images with large colour palettes. They are good for photos and images with lost of gradients in them. Uses lossy compression which means image quality is reduced.
* GIF images are used for simple animations. They have a limited colour palette. The image quality is not as good as JPG images.
* PNGs are good for anything that is not a photo or animated. They are mostly used for icons, diagrams, logos, etc. They use lossless compression which means the quality of the image remains the same.
* SVG images are vector-based instead of pixel-based. This means they can be scaled up or down without a loss of image quality. They are used in a similar way to PNGs. They are good for logos since they can be scaled to different sizes across your website with the logo looking worse.

Knowledge Check Answers:

1. <a> is used to create links.
2. An attribute is additional information that is stored with an element to modify it in some way.
3. The href attribute is used to tell links where to go.
4. An absolute link is a hyperlink which contains the full URL. A relative link is one that only contains a file path (relative to the current directory)
5. The <img> tag lets you create an image element
6. <img> elements need a src attribute and a an alt attribute.
7. To access a parent directory (go up) you use ../
8. The four main image formats are JPG, GIF, PNG, and SVG.

## Commit Messages

* Commit message history will be helpful if you come back to a project after a while as you will be able to read through the changes and processes you were doing.
* It is best practice to commit every time you have a meaningful change in the code. This will create a timeline of progress.
* <https://cbea.ms/git-commit/>

Knowledge Check Answers

1. Good commit messages and a good commit history can be beneficial as it makes going back to old projects easier. It also makes it easier for other people to understand what you are doing, why you are doing it, and how you are doing it.
2. No more than 50 characters for the subject line.

## Project: Recipes

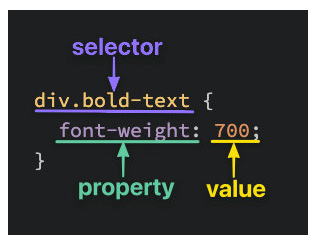
For this project, I had to create a website that has a menu with links to different recipe pages (that I created).

GitHub Link: <https://github.com/JoshuaBilsland/odin-recipes>

# Foundations - CSS Foundations

## CSS Foundations

* CSS is made up of various rules. These rules are made up of a selector and a semi-colon separated list of declarations. Each declaration is made up of a property:value pair.



* The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
* Selectors refer to the HTML elements to which the CSS rules apply (what is being selected for each CSS rule)
  + The universal selector will select elements of any type. The syntax for it is an asterisk.

Shape, rectangle

Description automatically generated

* + The type selector (or element selector) will select all elements of a given element type. The syntax is the name of the element.

Shape, rectangle

Description automatically generated

* + The class selector will select all elements with the given class. The class is an attribute you place on an HTML element.

Text

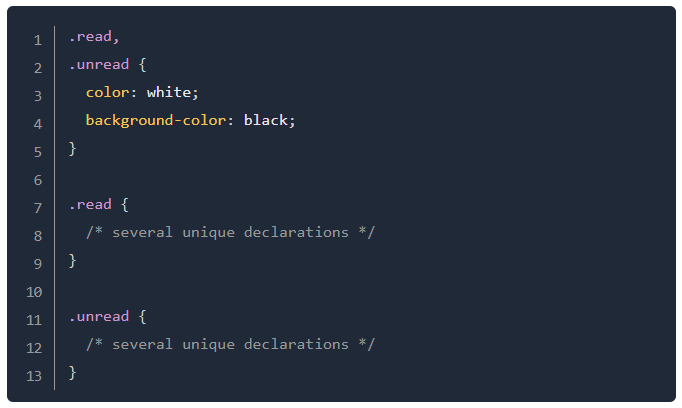
Description automatically generated

* + The ID selector are similar to class selectors. They select an element with the given ID (which is another attribute you place on an HTML element).

Graphical user interface, text

Description automatically generated

* + Grouping selectors can be used to cut down on repetition by applying the same declaration/s to multiple elements at a time.



* + Chaining selectors lets you apply a separate rule to an element by ‘chaining’ selectors so that you don’t affect elements that has a similar class/ID.

Graphical user interface, text, application

Description automatically generated

* + Combinations allow you to combine multiple selectors by showing the relationship between the selectors (child, sibling, etc)

Text

Description automatically generated

* The ‘color’ property sets an element’s text colour. The ‘background-color’ property sets the background colour of an element. Both of these can accept several kinds of values. You can use keywords (such as ‘red’ or ‘transparent’), HEX, RGB, and HSL. https://www.w3schools.com/cssref/css\_colors\_legal.asp

Text

Description automatically generated

* ‘font-family’ is used to select what font an element uses. It can be given the name of a font-family value (such as “times”, “courier”, etc) or it can be given a generic-family value (such as “serif”, “fantasy”, “monospace”, etc). You can give a list of fonts which is good practice since if a browser cannot find or does not support the first font in the list, it will keep going to the next one until it finds one that works.
* ‘font-size’ will let you set the size of the font. An example value would be ’22px’.
* ‘font-weight’ affects the boldness of text (provided the font supports the specified weight). The value given can be a keyword, such as bold, or it can be given as a number, such as 700 (equivalent of bold).
* ‘text-align’ will align text horizontally within an element. The value given to it are the same keywords you will find in word processors. For example, ‘center’.
* By default an <img> element’s height and width will be the same as the actual image file’s height and width. If you want to adjust it, you should give the value of ‘auto’ to height and adjust the width value.

Shape, rectangle

Description automatically generated

* It is good practice to include these properties for <img> elements, even if you don’t plan on adjusting the values from the image file’s original ones. This because it will reserve space on the page which will prevent the page contents moving to ‘fill the gap’ if an image is taking long to load. Instead, a blank space will be created until the image loads.
* If there is some unexpected behaviour (such as a paragraph being a different colour or a button looking a certain way) it’s either because of the default styles used by the browser or it is because you have not used properties correctly.
* Cascade is what determines which rules actually get applied to our HTML.
* Many factors which cascades uses to determine which rules to apply and to what. Some examples are…
  + Specificity - A CSS declaration that is more specific will take precedence over less specific ones. For example, ID selectors are higher (more specific) than Class selectors. Class selectors are more specific than Type selectors.

Text

Description automatically generated

Text

Description automatically generated

* + Inheritance - Inheritance refers to properties that are inherited by an element’s descendants, even if we don’t explicitly write a rule for those descendants. Some examples are ‘Typography based properties’ such as ‘color’, ‘font-size’, font-family, etc.

Text

Description automatically generated

* + Rule Order - If a rule with the same level of specificity, the rule that is declared last in the CSS document will be the one that is applied.

Text

Description automatically generated

* There are three methods to add CSS to HTML
  + External CSS - The most common method. A separate file for the CSS is created. The HTML is linked to the CSS file by using a <link> element within the <head> tags. It keeps HTML and CSS separated which makes the HTML file smaller and easier to read. It also means that any edits to the CSS can be carried out in one place which makes it easier to find what things you need to change.
  + Internal CSS/Embedded CSS - involves adding the CSS into the HTML file itself. All the rules are put into a pair of opening and closing <style> tags (which are put within the <head> tags). This can be useful for adding unique styles to a single page of a website. However, it can lead to very large HTML files which are not clean to look at.
  + Inline CSS - This involves adding styles directly to HTML elements. Selectors are not needed since the styles are added to the elements directly. This method is ok if you want to add a unique style for a single element. This method is not recommended for a few reasons:
    - It can make a HTML file feel ‘bloated’ once you start adding a lot of declarations to many different elements.
    - If you want many elements to have the same style, you would have to copy + paste the same style to each individual element. This also makes updating that style more difficult.
    - Any inline CSS will override the other two methods, which can cause unexpected results.
* <https://github.com/JoshuaBilsland/css-exercises>

Knowledge Check Answers

1. External CSS is where you create a link in your HTML to a separate CSS file. Internal CSS is where you write CSS rules into the head element of a HTML file (rather than writing it into a separate file). Inline CSS is where you write CSS rules directly in an element’s opening tags.
2. A class selector is created by putting a ‘.’ infront of the class name. For example, ‘.alert’. An ID selector is created by doing the same thing but with a ‘#’ instead of a ‘.’ .
3. You can group multiple selectors together into a list so that CSS rules can be applied to all of them together rather than having to rewrite it separately for each class/element.
4. You would write ‘ #title.primary ‘ for the selector
5. A descendant combinator select elements based on if they have the specified ancestors.
6. One class selector will have the higher specificity than three type selectors because class selectors are higher up than any amount of type selectors.

## Inspecting HTML and CSS (Chrome)

* To open the inspector press F12
* In the ‘Elements’ pane, you can see the entire HTML for the page.
* Select an element and read the ‘Styles’ tab to see all the styles that are applied to the element. If any of the styles are overwritten, they will be displayed with a strikethrough it.
* The ‘Styles’ pane allows you to edit styles directly within the browser. You can add new rules or modify existing ones.
* <https://developer.chrome.com/docs/devtools/>
* Overview (<https://developer.chrome.com/docs/devtools/overview/>)
  + DevTools can help edit pages on-the-fly and diagnose problems quickly.
  + <https://www.youtube.com/watch?v=VYyQv0CSZOE>
  + Device mode lets you preview a website on a mobile device (inside of your computer’s browser)
  + Elements panel - View and change the DOM and CSS
  + Console panel - View messages and run JavaScript from the console
  + Sources panel - Debug JS, persist changes made across page reloads, and save changes to disk.
  + Network panel - View and debug network activity
  + Performance panel - Find ways to improve load and runtime performance
  + Memory panel - Provides information about how a page is using memory
  + Application panel - Inspect all resources that are loaded.
  + Security panel - Debug mixed content issues, certificate problems, etc.
* Open Chrome DevTools (<https://developer.chrome.com/docs/devtools/open/>)
  + To inspect an element - Right click on an element, then click inspect element. Or press Control+Shift+C. Or press F12
  + To go straight to the ‘Console’ panel - Press Control+Shift+J
  + To open the last panel - Press Control+Shift+I
* View and Change CSS (<https://developer.chrome.com/docs/devtools/css/>)
  + \* This page is a series of tasks to help learn how to use Chrome’s DevTools to modify CSS \*
  + The .cls option lets you add a new CSS class to an element
  + The .hov option lets you apply a CSS pseudostate to an element.
* Get Started with viewing and changing the DOM (<https://developer.chrome.com/docs/devtools/dom/>)
  + \* This page is a series of tasks to help learn how to user Chrome’s DevTools to view and change the page’s DOM \*
  + Open the command menu - Control+Shift+P
  + Open the search bar - Control+F
  + You can write click an element in the DOM and click the ‘Force State’ option to preview what an element looks like in different states. For example, active, hover, visited, etc.
  + Press H with an element selected to hide it. Press the Delete key to delete it.

Knowledge Check Answers

1. To select a specific element on your page, right click on it and click ‘Inspect’.
2. A strikethrough a CSS rule means that the rule/CSS had been overwritten.
3. You can change CSS in real time by using the ‘Styles’ tab.

## The Box Model

* Every single thing on a webpage is a rectangular box. These boxes can have other boxes in them or next to them.

Graphical user interface, text, application, email

Description automatically generated

* Positioning elements is about deciding how you are going to nest and stack these boxes.
* Padding - Change the size of the space between the edge of a box and the content inside of it.
* Margin - Change the space between a box and any other that sit next to it.
* Border - Add space between the margin and padding.

Text

Description automatically generated

* <https://www.youtube.com/watch?v=rIO5326FgPE>
* Top and bottom margins collapse. The largest Margin between two boxes will merge rather than add together. If two boxes have a margin of 60px, there will be a gap of 60px between them not 120px.
* “box-sizing: border-box;” will make the padding and border included in the width and height.
* <https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/The_box_model>
* There are two types of boxes - block boxes and inline boxes. Boxes have an inner display and an outer display.
* <https://css-tricks.com/almanac/properties/m/margin/>

Knowledge Check Answers

1. The properties of the box model go padding -> border -> margin.
2. box-sizing lets you include the padding and border in the element’s total width and height.
3. The standard box model lets you define a height and width for the box. Any padding and border is then added to those height and width values to give it a final total size. In the alternative box model, the padding and border is taken away from the width and height and so the box stays true to the height and width values.
4. Margins are used to create space between two boxes.
5. Padding is used to create space between the content of a box and its border.
6. You would use a negative margin if you wanted to make two elements overlap.

## Block an Inline

* Block elements will appear on the page stacked on top of each other. (Each will be on a new line)
* Inline elements appear in line with whatever elements they are placed beside.
* Inline-block elements behave like inline elements, but with block-style padding and margin.
* Div is a block-level element. It is used as a container element to group other elements. Allows you to divide the page into different blocks and apply styling to those blocks.
* Span is an inline-level element. It can be used to group text content and inline elements for styling.
* <https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Normal_Flow>
* <https://www.w3schools.com/html/html_blocks.asp>
* <https://www.digitalocean.com/community/tutorials/css-display-inline-vs-inline-block>
* <https://github.com/JoshuaBilsland/css-exercises>

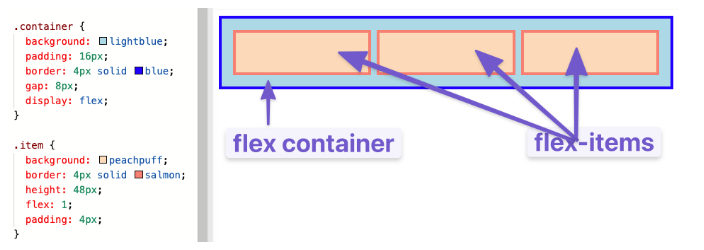
Knowledge Check Answers

1. Block elements will appear on a new line whereas inline elements will appear on the same line.
2. An inline-block element stays on the same line, but the width and height of the block is still respected.
3. A h1 element is a block element.
4. A button element is an inline element.
5. A div element is a block element.
6. A span element is a an inline element.

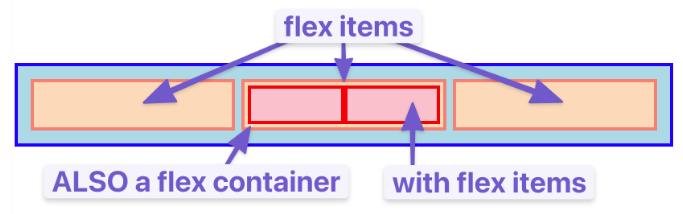
# Foundations - Flexbox

## Introduction to Flexbox

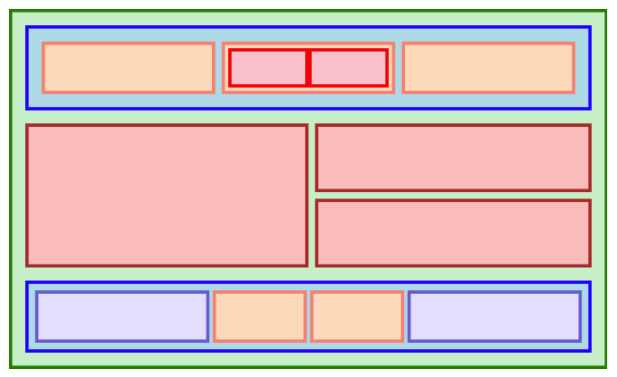
* Relatively new of manipulating elements in CSS. Already a common way of positioning elements for many developers.
* Flexbox is a way to arrange items into rows or columns. The items will then flex (grow or shrink) based on defined rules.
* A flex container is any element that has a “display: flex” attribute on it.
* A flex item is an element that lives directly inside of a flex container.



* Additionally, any element can be both a (flex) container and item.



* Flex containers and items (along with nesting) is the primary way to build up complex layouts of elements. For example…



* <https://www.internetingishard.com/html-and-css/flexbox/>

Knowledge Check Answers

1. A flex container is an element that contains the attribute “display: flex”. An item is an element that goes within a container.
2. A flex item is created by making it part of a flex container class (using the class attribute). In your CSS file, you would give different CSS rules to determine how those elements should look within the container.

## Growing and Shrinking

* The “flex” declaration is a shorthand for three properties that can be sent on a flex item. The properties affect how to item will size themselves within a container.
* “flex” is actually a shorthand for “flex-grow”, “flex-shrink”, and “flex-basis”.

Graphical user interface

Description automatically generated with low confidence

* In this example, “flex: 1” is equivalent to “flex-grow: 1”, “flex-shrink: 1”, “flex-basis: 0”. This would be the shorthand of flex: 1 1 0
* “flex-grow” takes a single number as its value which represents its “growth factor”. If three items all have the flex-grow value of 1, all the objects will grow to the same size. If one of those items has the value of 2 (and the others still have 1), that item will grow twice as large as the others.
* “flex-shrink” is similar to flex grow but references the “shrink factor” of a flex item. It is only applied if the size of all flex items is larger than their parent container. “flex-shrink: 1” is the default and means all items will shrink evenly. “flex-shrink: 0” will tell items not to shrink. Higher numbers will make an item shrink more than others.
* “flex-basis” is used to set the initial size of a flex item. Gives a baseline size for flex-grow and shrink to use. The shorthand defaults to “flex-basis: 0%”. 0 will ignore the items width. “auto” will check for the width and use that.
* “flex: auto” is the equivalent of “flex: 1 1 auto”.
* <https://www.w3.org/TR/css-flexbox-1/#flex-common>
* <https://developer.mozilla.org/en-US/docs/Web/CSS/flex>
* <https://css-tricks.com/understanding-flex-grow-flex-shrink-and-flex-basis/>
* Additional Resource: <https://www.youtube.com/watch?v=u044iM9xsWU&t=1>

Knowledge Check Answers

1. flex-grow: 1, flex-shrink: 1, flex-basis: auto