* **no closing tag**

<!DOCTYPE html>

<img src=”someimage.png”>

<link type=”text/css” rel=”spreadsheet” href=”stuff.css”>

<!-- some comment here –>

**has closing tag**

<html>

<ul> - unordered list

<ol> - ordered list

<li> - list item (the content of ul/ol)

<table> - initiates table

<tr> - table rows

<th> - table header

<td> - table data

<blockquote> - quote for a large amount of text

<abbr title=””> - thing inside title is unabbreviated

<a href=””> - thing inside href is a link

<title> - title of website that display on tab

<div> - section block level

<span> - section inline

<address> writes in italic

**keywords**

tags - these things <> </>

element - main things in <> ex. <**body**>

attributes - small things that helps element ex <img **src**=“”>

selector - the target of style in css ex. **h1** {color: black;}

property - what is being changed ex. h1 {**color**:black;}

value - value of the property ex. h1 {color: **black;**}

**attribute**

id=”name” - unique to an element

class=”name” - can be used for multiple elements

href=”/file.ext” - used to reference other files/url/sections

rel=”stylesheet” - used to relate current document to outside resource

src=”/image.png” - for img file

alt=”some text” - an alternative text in place of image

title=”some text” - a tooltip when moused over the text

**css**

<link rel=”stylesheet” href=”/something.css”> can be used as well

inline style ex. <h1 **style=“color:black”**>text</h1>

external css ex. **tag/#id/.class** {display: flex;}

margin - create space (outside border)

padding - create space (inside border)

outline - margin except it only creates line instead of space

text-align - horizontal alignment of text

vertical-align - vertical alignment of text

font-family - sets fonts with subsequent fonts being backups

pseudo class ex. a**:link/visited/hover/active**

**selector**

**\*** {}

**a** {}

**.className** {}

**#idName** {}

**a.className** {} - targets <a> with class “className”

**a[class=className] {}** - targets <a> with ATTRIBUTE class=”className”

**div > a** {} - a is directly in div

**div, a** {} - div or a

**div a**{} - a somewhere in div

**specificity**

id > class > element > \*

a#idName > a[id=idName]

a[class=className] > a.className

inline css > css stylesheet

**web components**

client - sending request

server - receives the request

protocol - http or https

database - store and retrieve data

rendering - html, css, and javascript for display and interaction

security - https for secure data transmission

**protocol**

http(80) - used to transfer web pages and resources across the web

https(443) - a secure version of http with encryption via SSL or TLS

ftp(20 and 21) - protocol to transfer files between computers

tcp - ensures data packets are sent accurately and in the correct order

ip - used to route data between devices

dns - translates domain to ip

**box model** - padding, border, margin

**flexbox**

justify-content - aligns items on the main axis

flex-start - left side

flex-end - right side

center

space-between - equal spacing between

space-around - equal space around

space-evenly - equal everywhere

align-items - aligns items on the cross axis

flex-start - top

flex-end - bottom

center

baseline - aligns by bottom of text?

stretch - fit the container

flex-direction

row - default (left to right)

row-reverse in reverse order (right to left)

column - default (top to bottom)

column-reverse - in reverse order (bottom to top)

order - changes the order of items

.targetClass {order: +-int} - where int is to adjust order

align-self - align-items except it is specific to the target

flex-wrap - spread out items

nowrap - default everything on a single line

wrap - wraps around additional lines

wrap-reverse - wraps in reverse (mirror’d on x axis)

flex-flow - combines flex direction and flex wrap

align-content - multiple lines/wrap’d alignment

**inline format**

<b> bold

<strong> bold, defines importance

<i> italicized

<em> italicized, defines importance

<small> smaller

<mark> highlights

<del> strikethrough ~~like this~~

<ins> underline like this

<sub> subscriptlike this

<sup> superscriptlike this

<q> - short quote

<bdo> writes everything right to left: olleh (hello)

<cite> writes in italic

**semantics**

<article> to distribute independently from the rest of the website

<aside> content placed on the side like a sidebar

<details> toggleable content that shows and hides when clicked

<figcaption> used with figure tag, adds a text

<figure> self contained content for diagrams, photos, etc

<footer> defines the footer section of the document

<header> defines the header section of the document

<main> defines the main section of the document

<nav> defines navigation links, used with <a>, separated with |

<section> defines sections, commonly used with id=”idName”

<summary> defines a visible heading for <details>

<time> defines specific time or datetime doesn’t render anything special

**grid layout**

.container{

display: grid;

grid-template-columns: 1fr 2fr 100px;

grid-template-rows: auto

}

with above code, being columns of 3 item

rest of the page = full page - 100px

first item: 1/3 of rest of the page

second item: 2/3 of rest of the page

third item: 100px

each with height depending on text wrap, like using sheets/excel

another example grid layout

.container{

display: grid;

grid-template-areas:

“header header”

“sidebar main”

“footer footer”;

grid-template-columns: 200px 1fr;

grid-template-rows: 50px 1fr 50px;

}

.header{

grid-area: header;

}

.sidebar{

grid-area: sidebar;

}

.content{

grid-area: main;

}

.footer{

grid-area: footer;

}

note, semicolon after grid temp areas is after the last row’s quotation marks

grid temp areas’ section names are specified by grid-area: name;

the code above will have

header 1 page wide, 50px tall

sidebar 200px wide, (page height - 50px - 50px) tall

content (page width - 200px) wide, same height as sidebar

footer 1 page wide, 50px tall

outputting something similar to the standard page setup but without nav bar

@media only screen and (max-width: 600px) {

body {

background-color: lightblue;

}

}

above is an example of media usage, the body’s bg colour is set to lightblue if the webpage’s width is 600px or less

div {

width: 100px;

height: 100px;

animation: mymove 5s infinite;

}

@keyframes mymove {

0% {top: 0px; left: 0px; background: red;}

25% {top: 0px; left: 100px; background: blue;}

50% {top: 100px; left: 100px; background: yellow;}

75% {top: 100px; left: 0px; background: green;}

100% {top: 0px; left: 0px; background: red;}

}

mymove can be any name, the percents can be any number between 0-100

above is an example of keyframes usage, a box of 100x100px, starts top left,

start is 0%, 100% is 5 seconds, and loops infinitely.

moves right 100px changing from red to blue

next moves down 100px changing from blue to yellow

next moves left 100px changing from yellow to green

moves back up 100px changing from green to red

**css part 2**

div{

width 100px;

transition: width 2s;

}

div:hover{width: 200px;}

on hover over the div, increases the width from 100 to 200 in 2 seconds. can be used with any pseudo classes and the width part can be any property

:root{

--some-color: red;

--more-color: blue;

}

div {

background-color: var(--some-color);

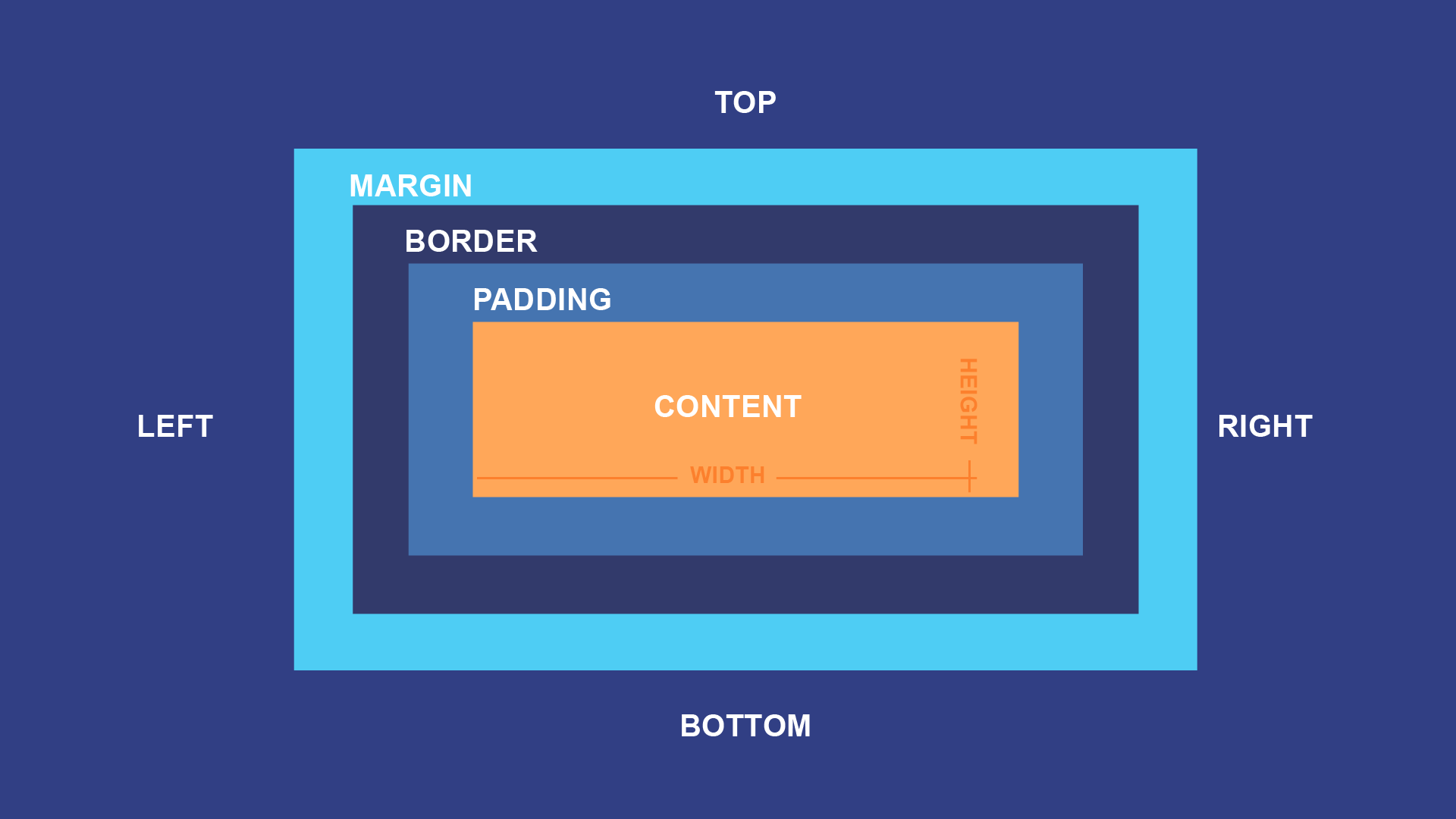
color: var(--more-color);

}

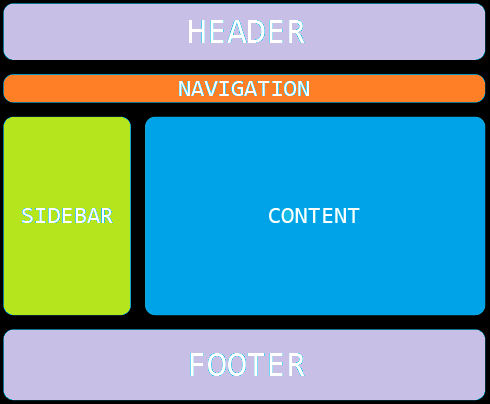
setting a custom value in css used with root

sets the background color to --some-color and font color to --more-color

where --some-color is red and --more-color is blue



box model



standard page setup