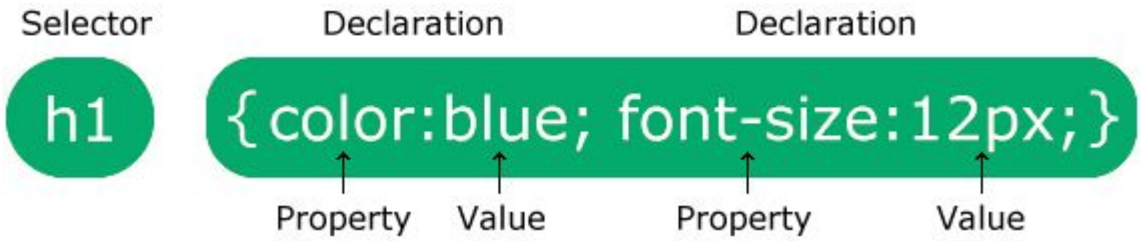
Web development basics

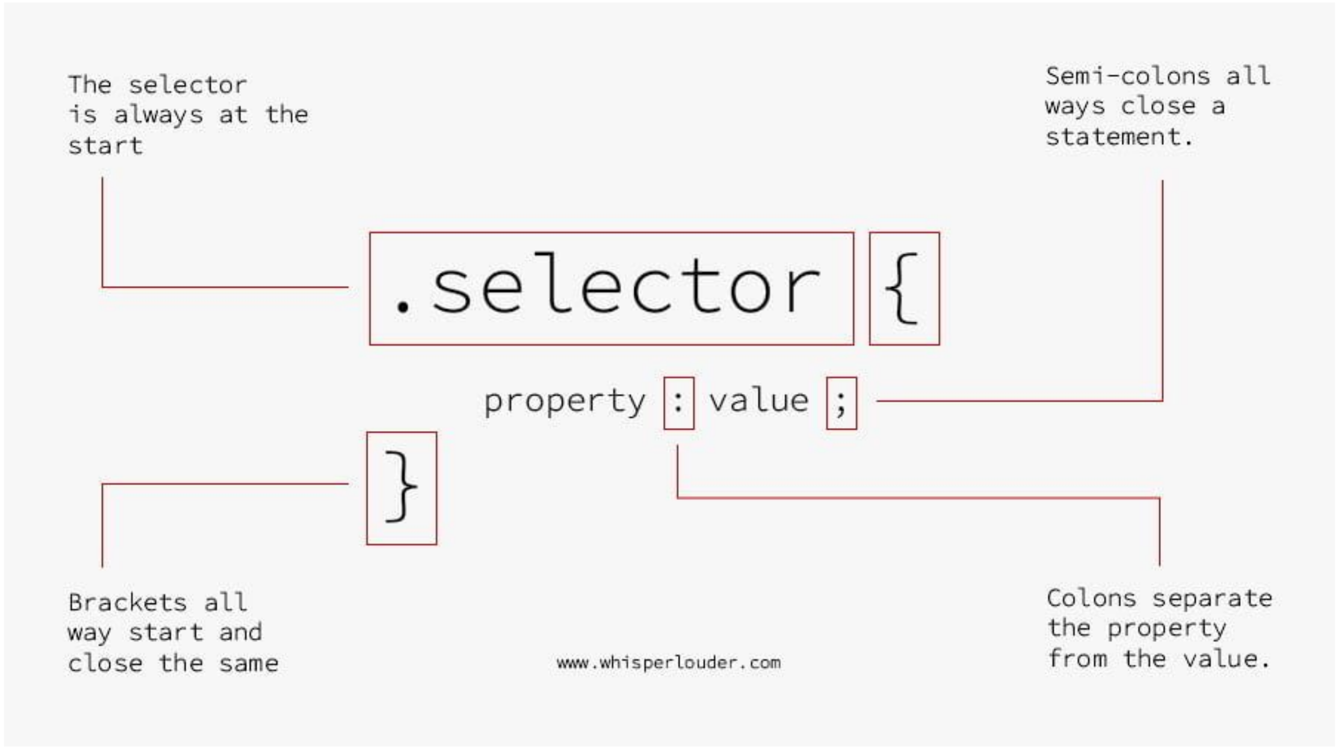
CSS basics

What is CSS

CSS stands for Cascading Style Sheets. It is a styling language used to describe the presentation of a document written in a markup language. With CSS, you can control the layout, colors, fonts, and other visual elements of a webpage. It is separate from the HTML, which is used to structure the content of the page. Using CSS, you can create visually appealing and consistent designs across multiple web pages.

CSS syntax


Selector
Declaration
Declaration
h1
{color:blue; font-size:12px;}
Property
Value
Property
Value

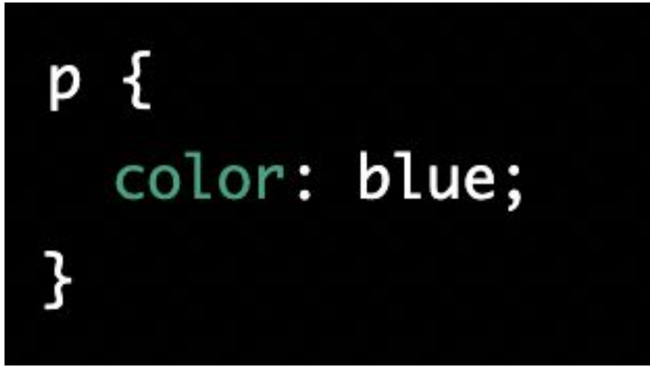

The selector is always at the start
Semi-colons all ways close a statement.
.selector {
property : value ;
}
Brackets all way start and close the same
Colons separate the property from the value.
www.whisperlouder.com

CSS selectors

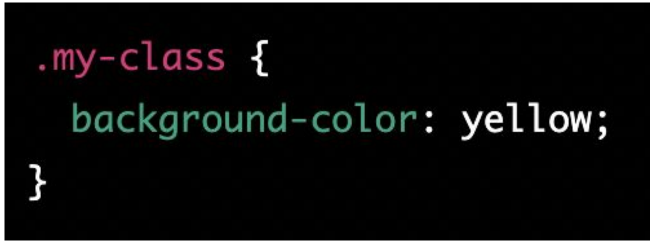
CSS selectors are used to select the HTML elements that you want to style.

There are several types of selectors:

1) Element Selector: Selects all elements with the given tag name, for example "p" will select all the <p> elements on the page.
2) Class Selector: Selects all elements with the given class name, for example ".my-class" will select all elements with class name "my-class".



```
p {
  color: blue;
}
```



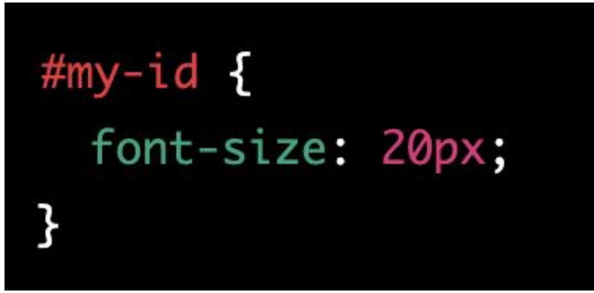
```
.my-class {
  background-color: yellow;
}
```

CSS selectors 2

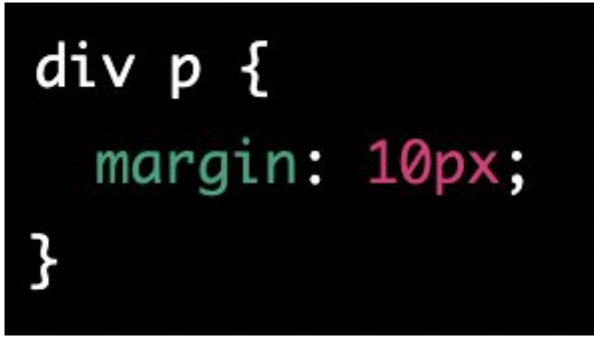
3) ID Selector: Selects the element with the given ID, for example "#my-id" will select the element with id "my-id".

4) Descendant Selector: Selects all elements that are descendants of a given element. for example "div p" will select all <p> elements inside <div>

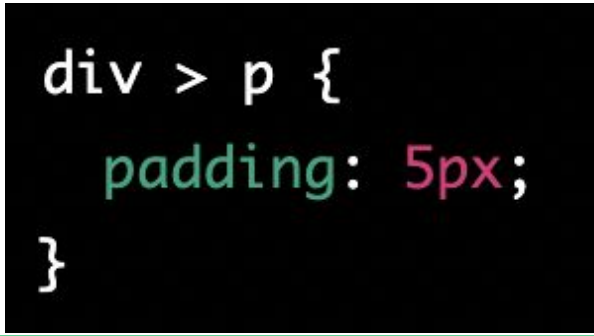
5) Child Selector: Selects all direct children of a given element. for example "div > p" will select all <p> elements that are direct children of <div>



```css
#my-id {
  font-size: 20px;
}
```

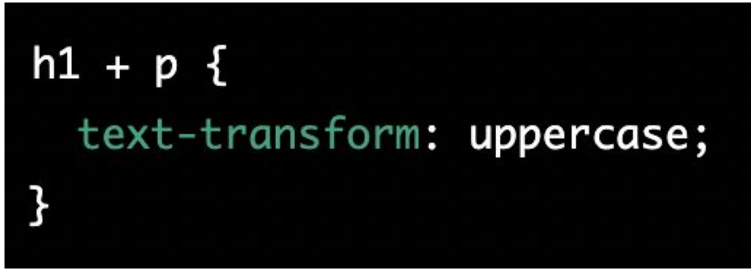


```css
div p {
  margin: 10px;
}
```



```css
div > p {
  padding: 5px;
}
```

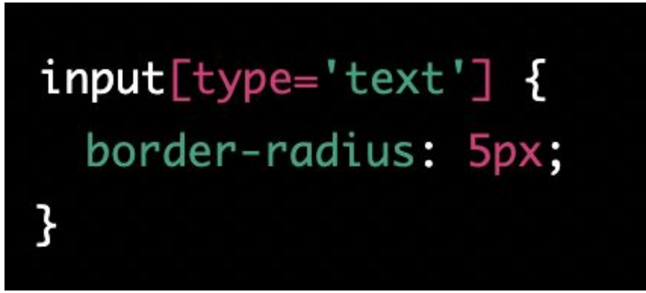
CSS selectors 3



```
h1 + p {
  text-transform: uppercase;
}
```

6) Adjacent Sibling Selector: Selects all elements that are next to a given element. for example "h1 + p" will select all <p> elements that immediately follow an <h1>

7) Attribute Selector: Selects all elements with a given attribute, for example "[type='text']" will select all elements with an attribute "type" and value "text"



```
input[type='text'] {
  border-radius: 5px;
}
```

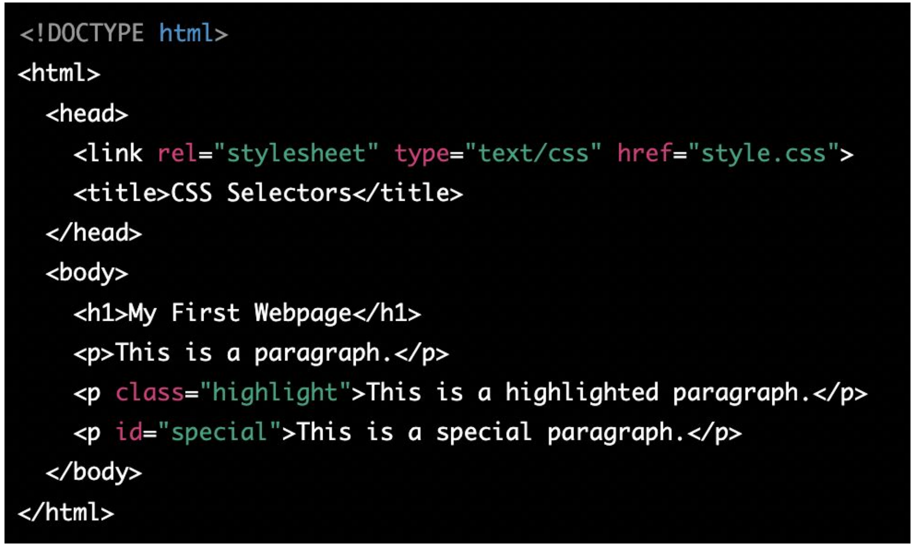
These are just a few examples of the many types of selectors available in CSS. Each selector has its own specific use cases and can be combined with other selectors to create more complex selectors.

Classwork for CSS selectors

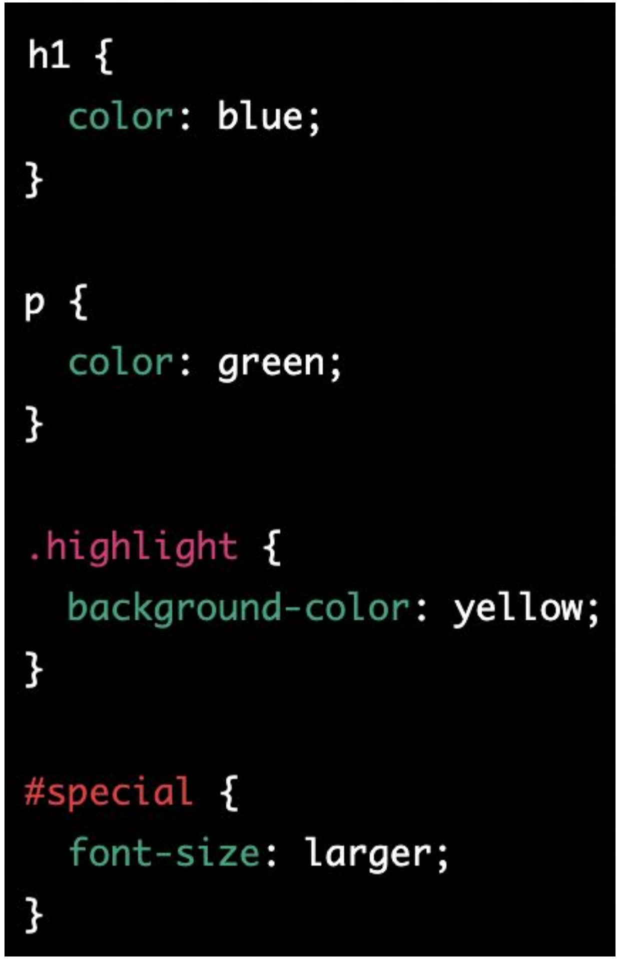
1) Open a text editor and create two new files, one with the HTML extension and one with the CSS extension.
2) In the HTML file, create a basic HTML structure
3) Link the CSS file to the HTML file using the <link> element in the <head> section of the HTML file

Expected outcome: The HTML file should display a blue <h1> heading, green paragraphs, a yellow highlighted paragraph with a class of "highlight", and a special paragraph with larger font size and an id of "special".

Classwork for CSS selectors - solution



```html
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
    <title>CSS Selectors</title>
  </head>
  <body>
    <h1>My First Webpage</h1>
    <p>This is a paragraph.</p>
    <p class="highlight">This is a highlighted paragraph.</p>
    <p id="special">This is a special paragraph.</p>
  </body>
</html>
```

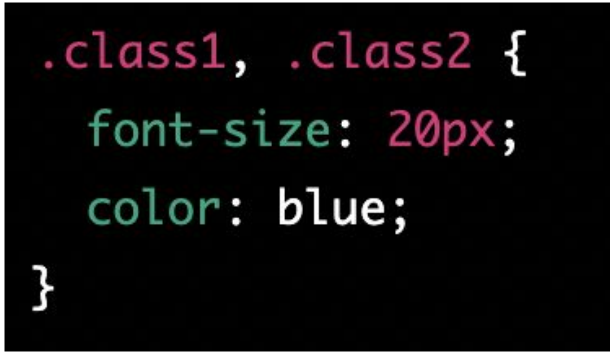


```css
h1 {
  color: blue;
}

p {
  color: green;
}

.highlight {
  background-color: yellow;
}

#special {
  font-size: larger;
}
```

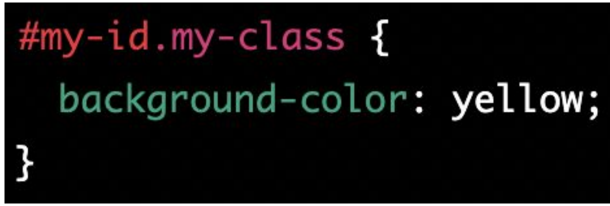
CSS selector combination

CSS allows you to combine selectors to create more specific and powerful styles. There are several ways to combine selectors, including:

1) Combining classes: You can combine multiple class selectors to apply the same styles to multiple elements. For example, you can use two classes to style two different elements in the same way
2) Class and ID: You can combine a class and an ID selector to apply styles to a specific element with a specific class. For example, you can use the following CSS to style a specific element with the ID "my-id" and the class "my-class"



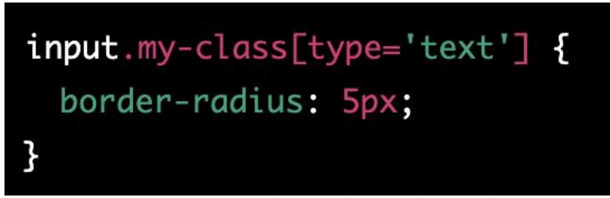
```
.class1, .class2 {
  font-size: 20px;
  color: blue;
}
```



```
#my-id.my-class {
  background-color: yellow;
}
```

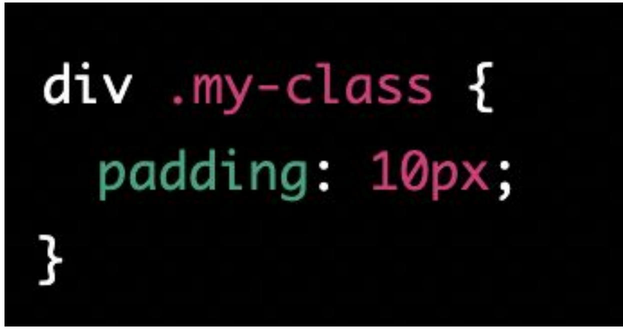
CSS selector combination 2

3) Class and attribute: You can combine a class selector and an attribute selector to apply styles to elements with a specific attribute and class. For example, you can use the following CSS to style all input elements with the class "my-class" and type attribute "text"



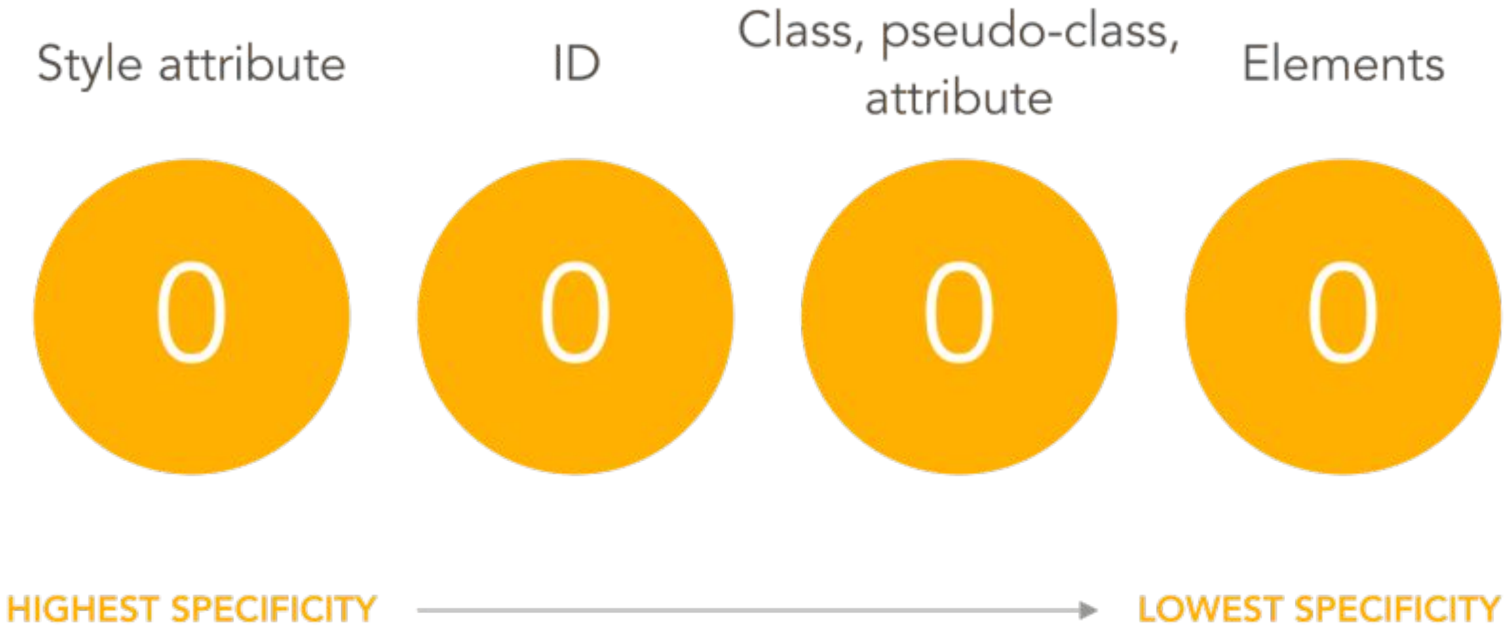
```css
input.my-class[type='text'] {
  border-radius: 5px;
}
```

4) Descendant selector and class: You can combine a descendant selector and a class selector to select all elements with a specific class that are descendants of a specific element. For example, you can use the following CSS to select all <p> elements with the class "my-class" that are inside a <div> element

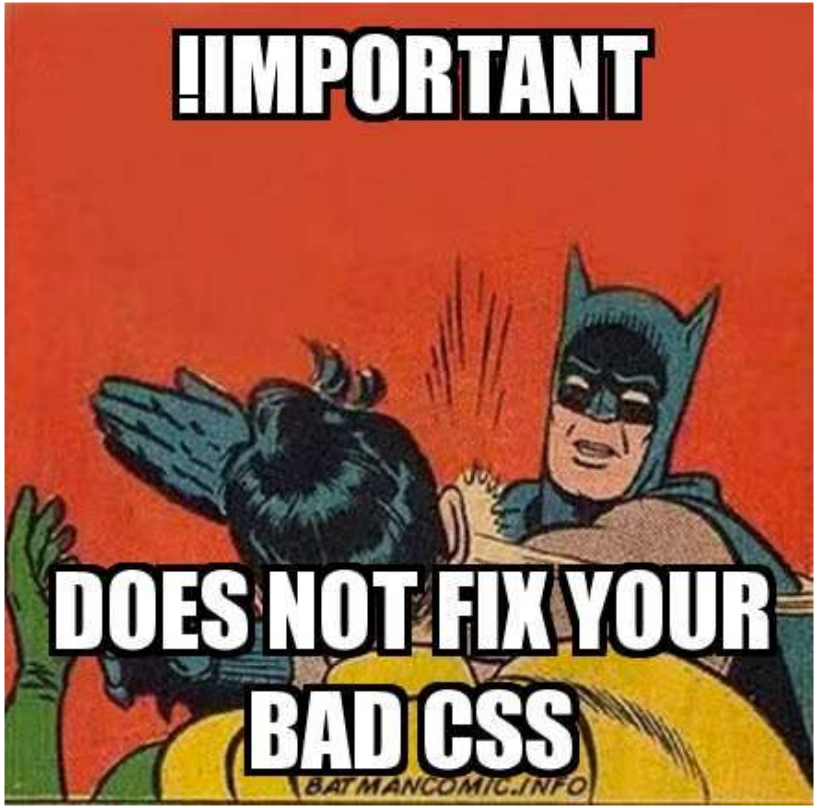


```css
div .my-class {
  padding: 10px;
}
```

CSS specificity

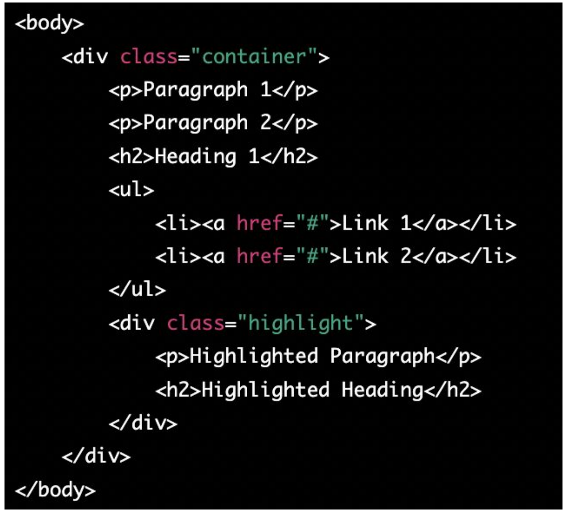


CSS specificity !important



Classwork for CSS selector combination

1) Set the background color of all paragraphs inside a div with a class of "container" to light gray.
2) Set the font size of all unordered lists inside a div with a class of "container" to 16 pixels.
3) Set the font color of all anchor tags inside an unordered list to blue.
4) Set the text color of all h2 headings that come immediately after a paragraph to red.
5) Set the background color of all elements with a class of "highlight" that are inside a div with a class of "container" to yellow.

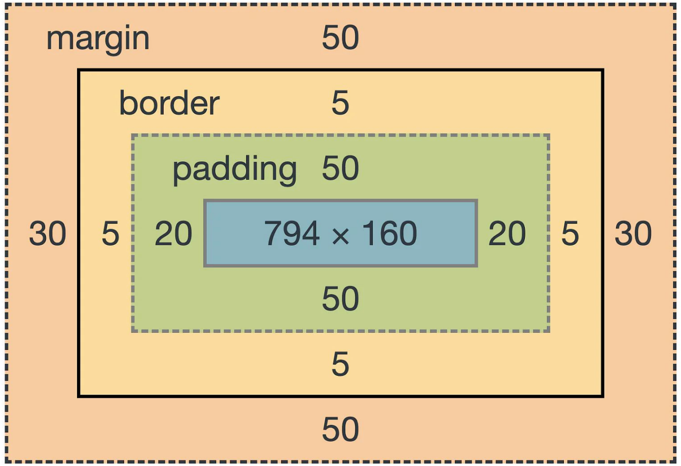


```html
<body>
    <div class="container">
        <p>Paragraph 1</p>
        <p>Paragraph 2</p>
        <h2>Heading 1</h2>
        <ul>
            <li><a href="#">Link 1</a></li>
            <li><a href="#">Link 2</a></li>
        </ul>
        <div class="highlight">
            <p>Highlighted Paragraph</p>
            <h2>Highlighted Heading</h2>
        </div>
    </div>
</body>
```

CSS box model

The CSS box model is a layout model that defines how the dimensions of elements on a web page are calculated. It describes how an element is sized, positioned, and rendered on a web page, including how the element's content, padding, border, and margin are handled. The box model consists of the following parts:

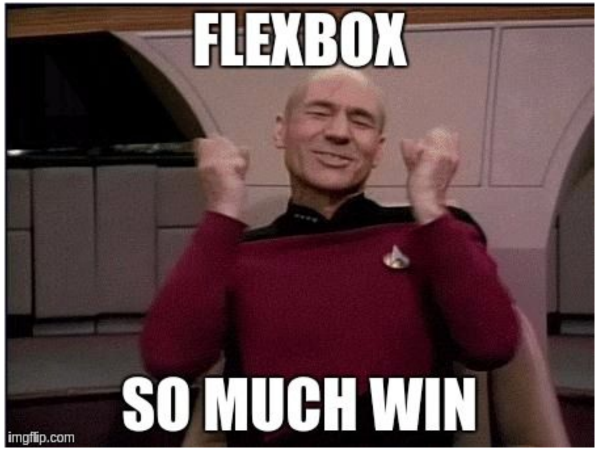
1. Content: This is the actual content of the element, such as text or images.
2. Padding: This is the space between the content and the border. It can be set using the padding property in CSS.
3. Border: This is a line that surrounds the padding and content. It can be set using the border property in CSS.
4. Margin: This is the space between the border and the other elements on the page. It can be set using the margin property in CSS.

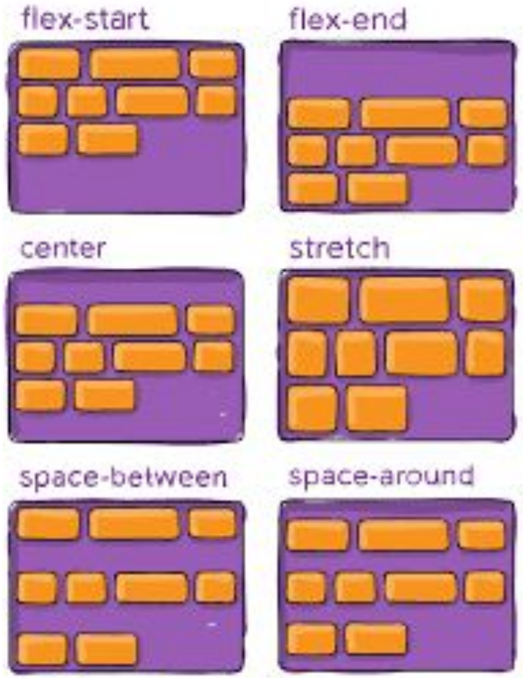


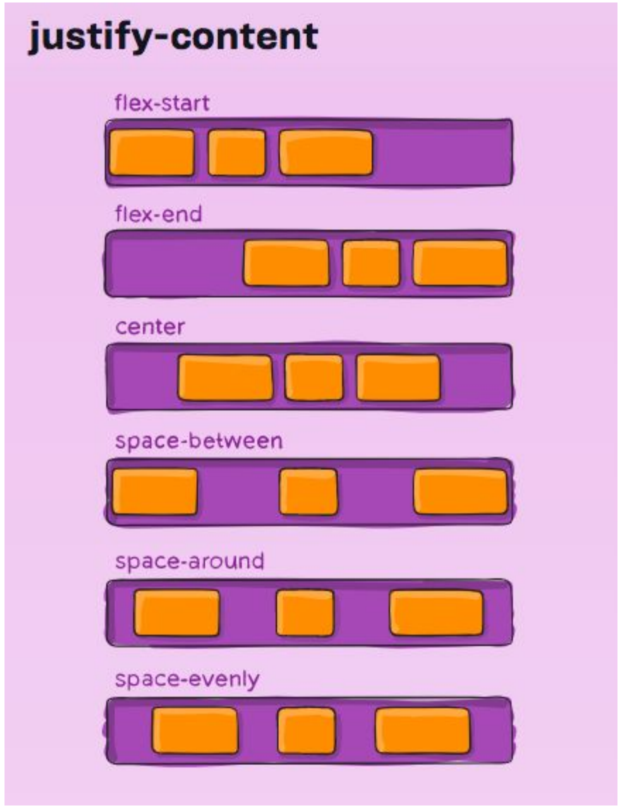
CSS flexbox

CSS Flexbox is a **layout module** in CSS that makes it easy to create flexible, responsive layout structures without using floats or positioning. It allows you to align elements horizontally and vertically, distribute space between elements, and reorder elements within a container. Flexbox is particularly useful when creating **responsive design**, as it allows you to control how elements are displayed on different screen sizes and orientations. To use flexbox, you first create a container element and set its **display** property to "**flex**". Then, you can use various flexbox properties to control the layout of the child elements within that container.

CSS flexbox 2







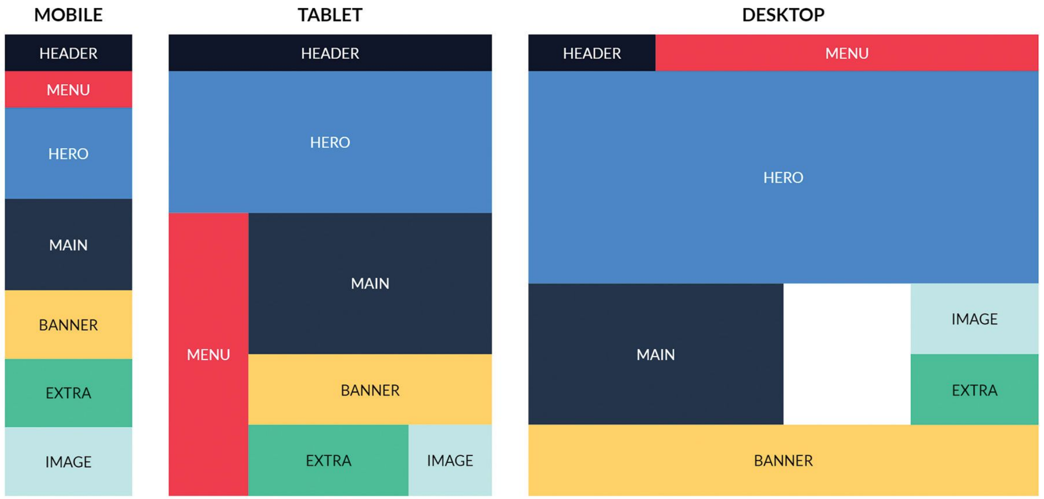
CSS grid

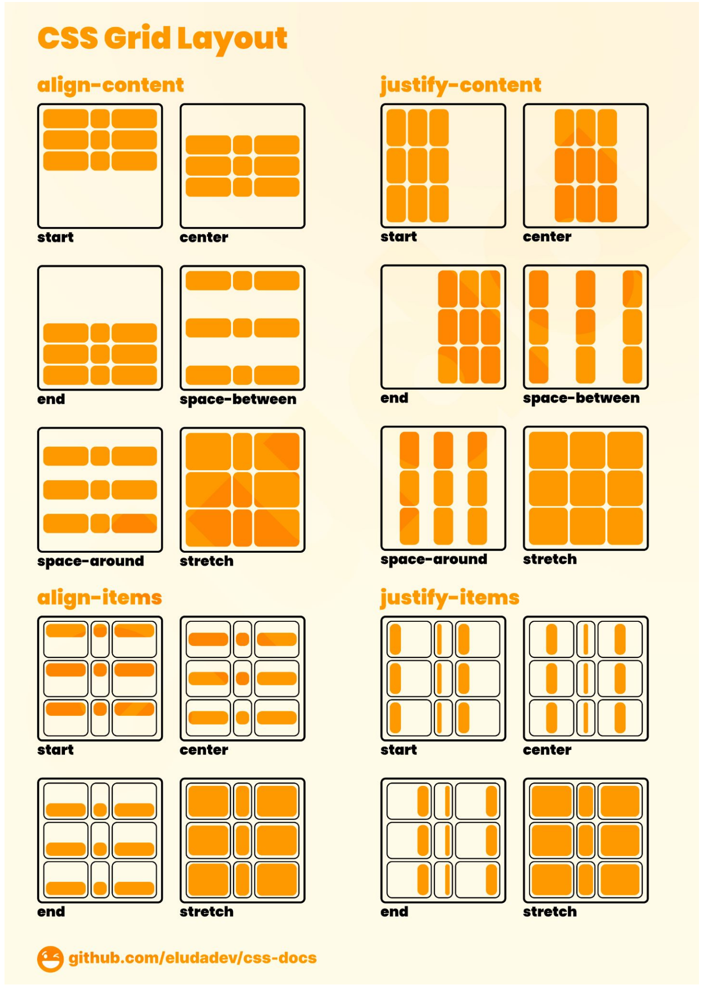
CSS Grid is a two-dimensional layout system for the web that allows you to create **rows** and **columns** to organize elements on a webpage. It is designed to make it easy to create complex, responsive layouts with minimal code. With CSS Grid, you can define a **grid container** and then specify how many **rows** and **columns** the grid should have, and how the elements within the grid should be placed. You can also control the **size** of rows and columns, and specify how elements should span multiple rows or columns. CSS Grid also allows you to control the alignment of elements within the grid and how the grid should respond when the **viewport size** changes.

One of the main advantages of CSS Grid is that it allows you to create layouts that are not possible with traditional CSS techniques such as **floats** and **positioning**. Grid also make it easy to create **responsive design** as it allows you to control how elements are displayed on **different screen sizes and orientations**.

To use CSS Grid, you first create a container element and set its **display** property to "**grid**". Then, you can use various grid properties to define the structure of the grid and the placement of the child elements within it.

CSS grid 2

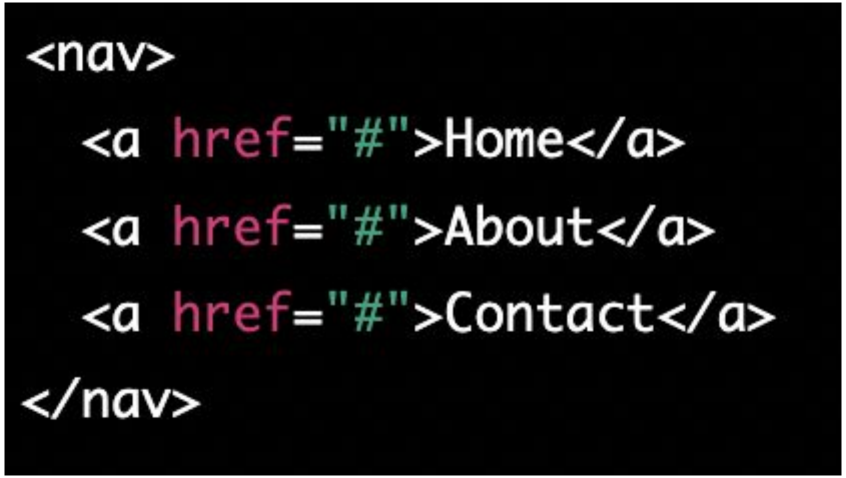

MOBILE
HEADER
MENU
HERO
MAIN
BANNER
EXTRA
IMAGE
TABLET
HEADER
HERO
MENU
MAIN
BANNER
EXTRA
IMAGE
DESKTOP
HEADER
MENU
HERO
MAIN
IMAGE
EXTRA
BANNER


CSS Grid Layout
align-content
start
center
end
space-between
space-around
stretch
justify-content
start
center
end
space-between
space-around
stretch
align-items
start
center
end
stretch
justify-items
start
center
end
stretch
github.com/eludadev/css-docs

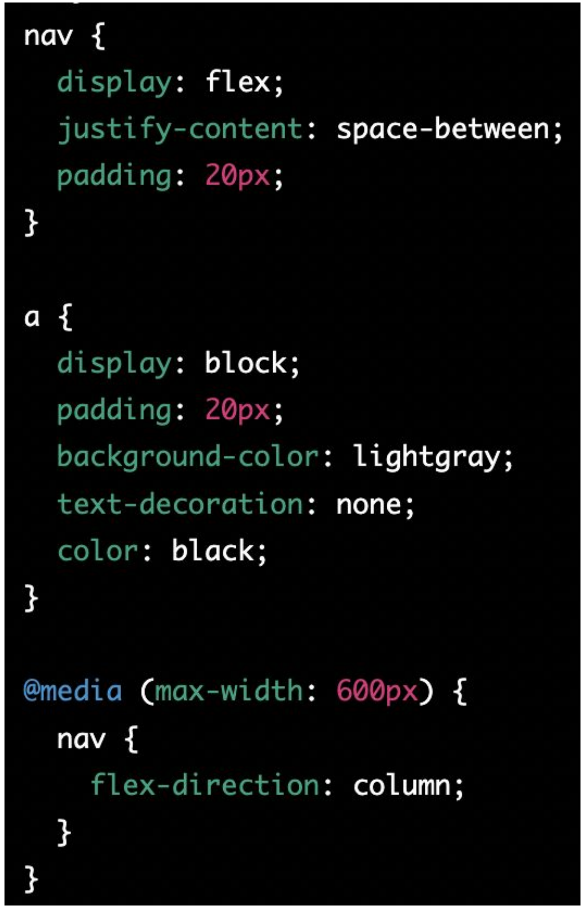
CSS flex classwork

1) Create an HTML file with a nav element to serve as the navigation bar container.
2) Inside the nav element, create a list of a elements to represent each navigation item.
3) Set the nav element's display property to flex and its justify-content property to space-between to evenly distribute the navigation items across the container
4) Add styles to the a elements to make them look like navigation items (e.g. display as block elements and give them padding and a background color).
5) Use media queries to adjust the flex-direction property of the nav element to change the orientation of the navigation items on different screen sizes **(optional)**
6) Preview your changes in the browser and adjust the styles as needed.

CSS flex classwork - solution

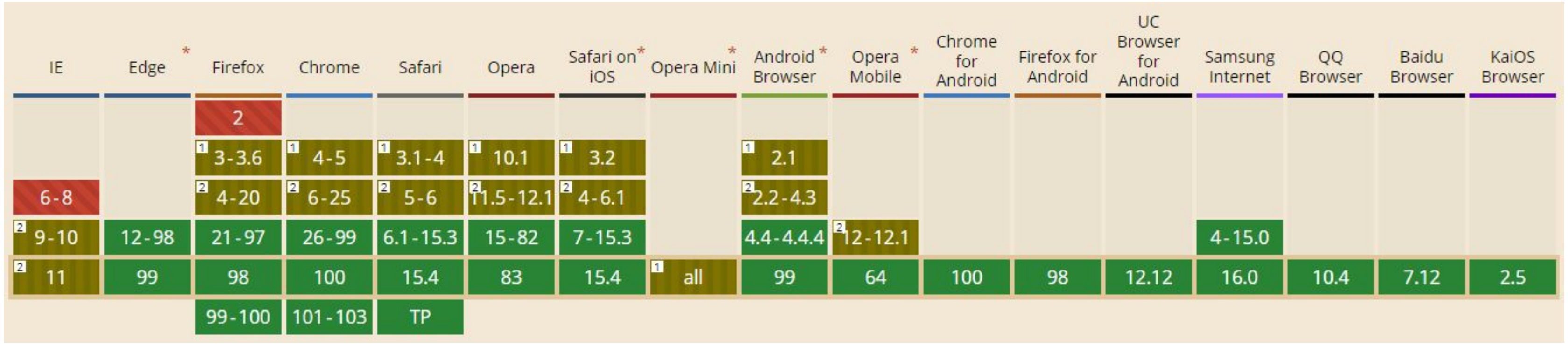


```
<nav>
  <a href="#">Home</a>
  <a href="#">About</a>
  <a href="#">Contact</a>
</nav>
```



```
nav {
  display: flex;
  justify-content: space-between;
  padding: 20px;
}

a {
  display: block;
  padding: 20px;
  background-color: lightgray;
  text-decoration: none;
  color: black;
}

@media (max-width: 600px) {
  nav {
    flex-direction: column;
  }
}
```

Can I use it?



IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
		2														
		[1] 3-3.6	[1] 4-5	[1] 3.1-4	[1] 10.1	[1] 3.2		[1] 2.1								
6-8		[2] 4-20	[2] 6-25	[2] 5-6	[2] 11.5-12.1	[2] 4-6.1		[2] 2.2-4.3								
[2] 9-10	12-98	21-97	26-99	6.1-15.3	15-82	7-15.3		4.4-4.4.4	[2] 12-12.1				4-15.0			
[2] 11	99	98	100	15.4	83	15.4	[1] all	99	64	100	98	12.12	16.0	10.4	7.12	2.5
		99-100	101-103	TP												

Homework

1) Add css styling to created portfolio
   a) Add responsive design to portfolio

Portfolio ideas -
https://bashooka.com/html/free-html-css-portfolio-web-design-templates/