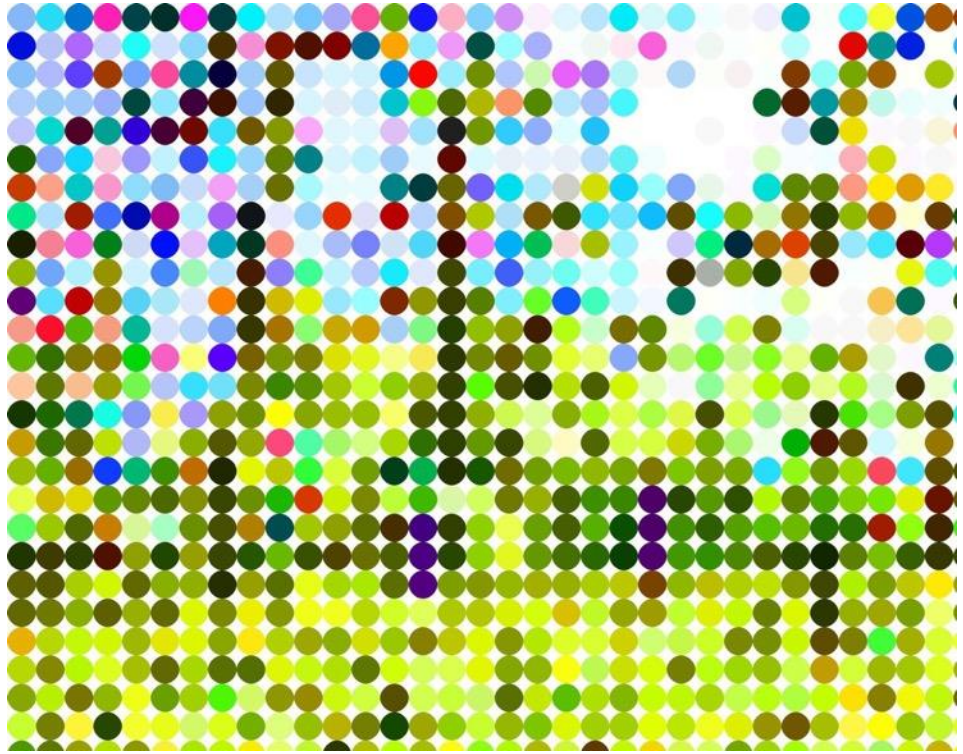
SHARE NETWORK

Inspiration Day 2

Share Academy

Dorien & Bilal



WIFI

SSID:

Lokaal Lokaal Gast

Password:

allpalacesaretemporarypalaces

Download slides:

https://code.share-network.org/inspiration-days

What’s up?

10:30	Introduction: Overview of today’s activities
10:45	Codecademy course + mentoring + project - How is it going? - Need help? – Any questions before we start? - Website Project; Share progress / show your website / your idea’s
11:00	Introduction to CSS – theory & instructions
11:30	Challenge – Dive into CSS
12:30	Lunch break
13:30	Challenge: Layout with Flexbox
14:00	Work on your project
15:00	End of this Inspiration Day

Recap html

HTML = Hyper Text Markup Language

Used in a semantic way, elements have meaning

Used for structure, not for styling

Most element have a opening and a closing tag

Browsers already have some default styling for elements

Html files have the .html extension

CSS

CSS = Cascading Style Sheet

A CSS file contains style rules

A CSS rule defines which element(s) to select and how to style them

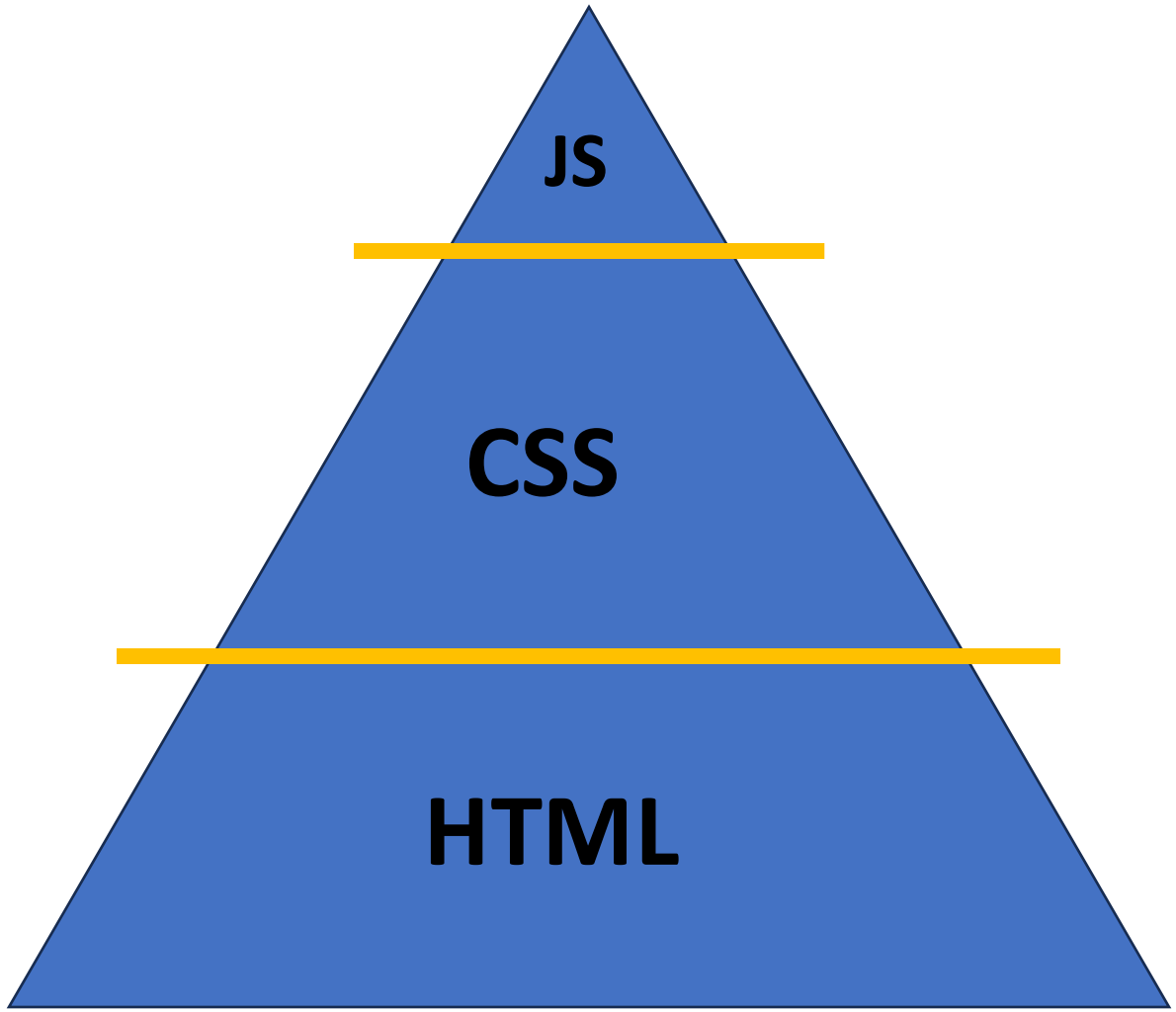
“Link” a CSS file to your HTML page with the <link> element

The webbrowser applies the CSS rules to the selected HTML elements

HTML = Structure and semantic content

HTML + CSS = structure + style = Beautiful webpages

CSS files have the .css extenstion


JS
CSS
HTML

HTML without CSS

```
<body>
    <section>
        <h1>Hello Kittens!</h1>
        <p>This is a page about cute little kittens.</p>
        <img src="https://placekitten.com/200/300" alt="Cute" />
    </section>
</body>
```

HTML page with:

Section element that contains:

- Heading (h1) element
- Paragraph (p)
- Image (img) element

Very basic and boring right now.
(default browser styles will be applied)

Create a CSS file

```
# style.css > p
1  body {
2      background-color: black;
3  }
4
5  section {
6      background-color: olivedrab;
7      padding: 20px;
8      border: 5px solid yellow;
9      width: 300px;
10 }
11
12 h1 {
13     font-size: 36px;
14     color: yellow;
15     font-family: "Roboto", sans-serif;
16     margin: 0;
17 }
18
19 img {
20     border-radius: 20px;
21     box-shadow: 1px 1px 5px #333333;
22 }
23
24 p {
25     color: white;
26     font-family: "Roboto", sans-serif;
27     font-size: 18px;
28 }
```

The **style.css** file contains all the style rules.

The style rules will be applied to the HTML page:

```
<body>
    <section>
        <h1>Hello Kittens!</h1>
        <p>This is a page about cute little kittens.</p>
        <img src="https://placekitten.com/200/300" alt="Cute" />
    </section>
</body>
```

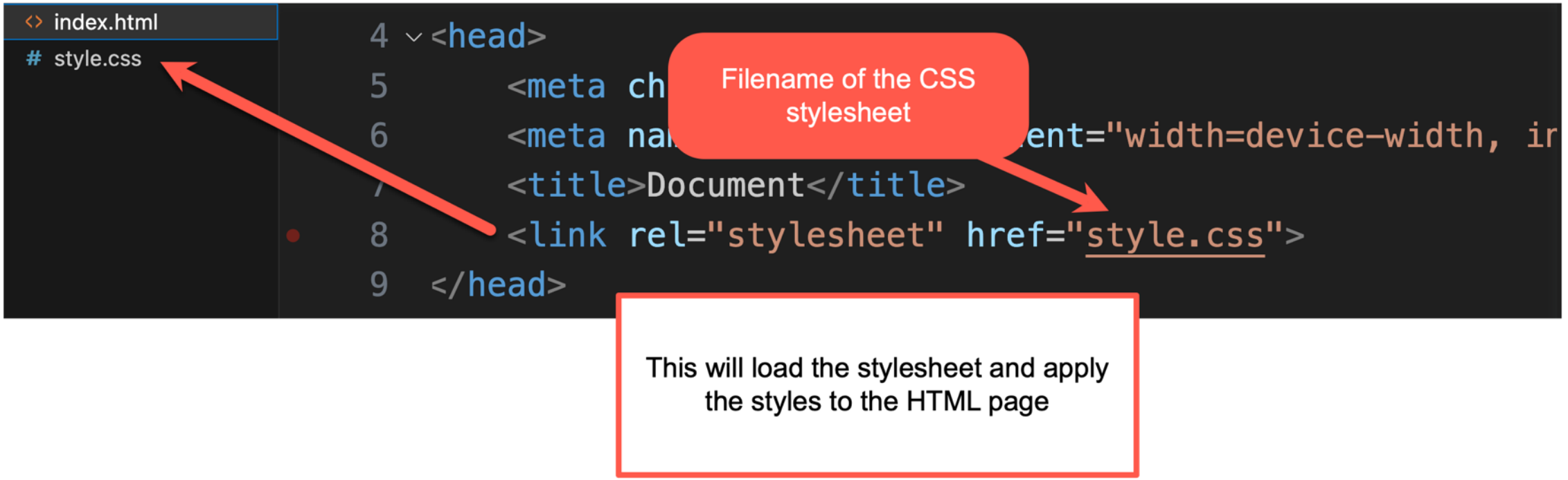
In this file there are style rules for these html elements:

<body>, <section>, <h1>, <img>, <p>

Link the CSS stylesheet to the HTML page

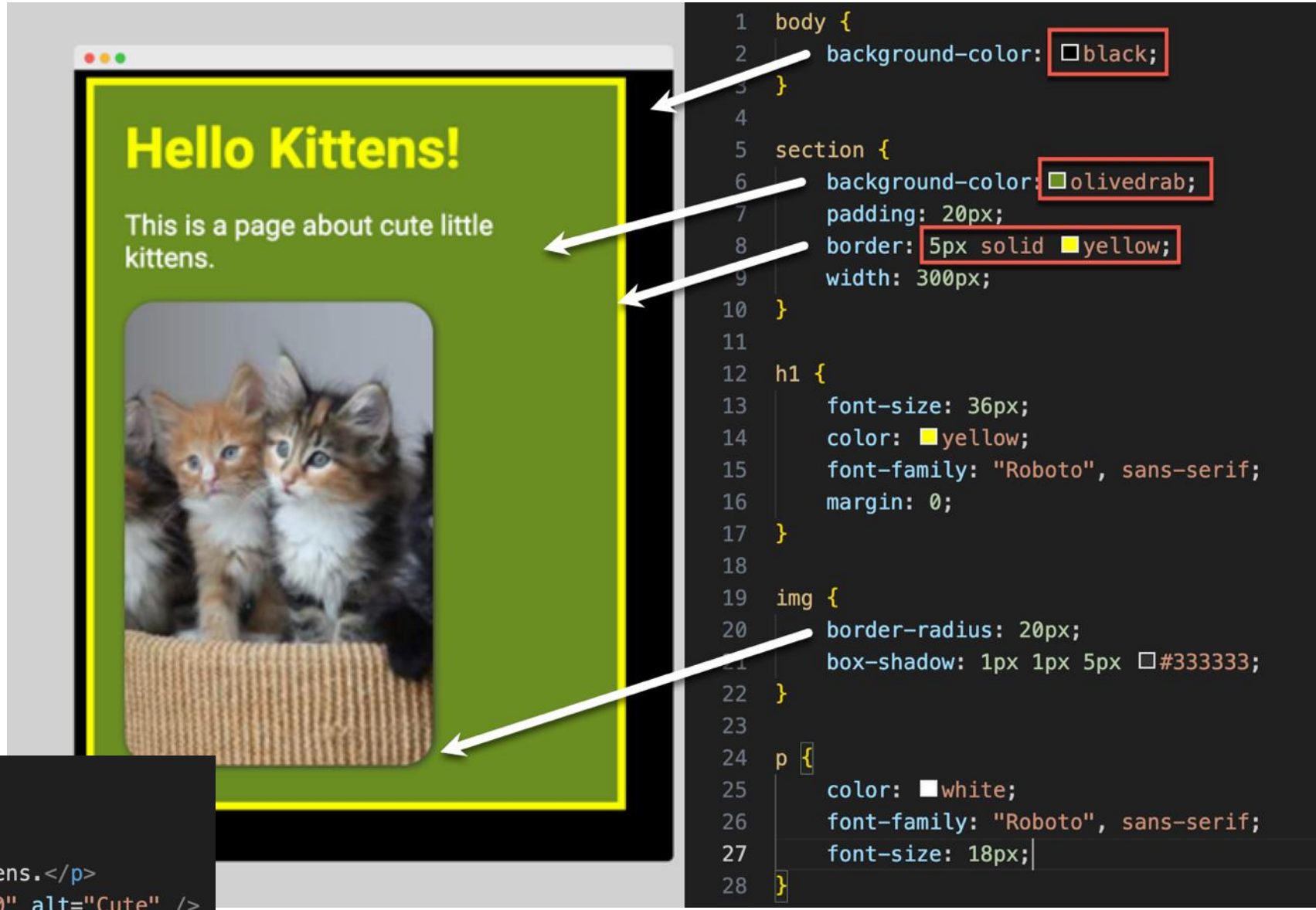
You have to **link the CSS file to the HTML page** to apply the style rules

Add a <link> with a reference (href) to your CSS file.
The <link> has to be inside the <head> element



Now the styles are applied!

This looks much better!

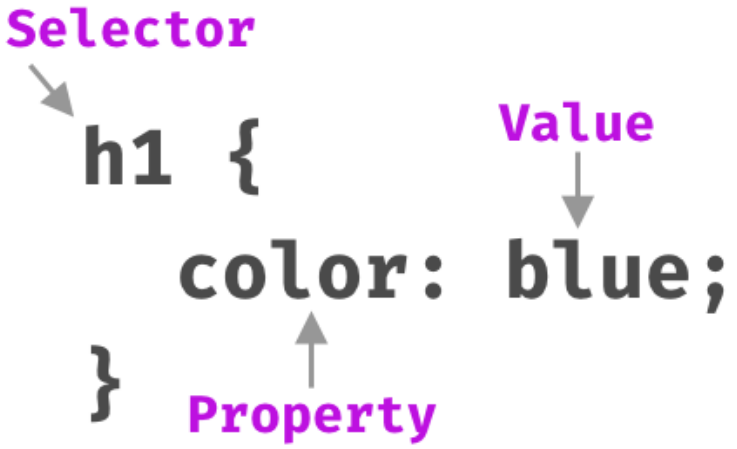


The HTML is still the same!

```
<body>
    <section>
        <h1>Hello Kittens!</h1>
        <p>This is a page about cute little kittens.</p>
        <img src="https://placekitten.com/200/300" alt="Cute" />
    </section>
</body>
```

How to write CSS?

1. First you write a **selector** to **select** the elements you want to target
2. Then you write the style declarations between **{** and **}**
3. Every style rule has a **property** and a **value,** separated by a **:** (colon)
4. Always end with a ; (semicolon)



This CSS declaration will set the **color** property of all **<h1>** elements to the value **blue**

Writing CSS Rules

Every CSS ruleset has this structure:

```
selector-of-element {
    style-property: value;
}
```

To style a **<h1>** tag with the **background-color** blue, the **color** white and a **font-size** of 18px you write:

```
h1 {
    background-color: blue;
    color: white;
    font-size: 18px;
}
```

Then, if you have this in your HTML file:

```
<h1>Have a nice day</h1>
```

Without the CSS it looks like:

Have a nice day

With the CSS applied to the HTML:

Have a nice day

There are a lot of CSS properties!

They all have different purposes.

You can look them up here:

https://developer.mozilla.org/en-US/docs/Web/CSS/Reference#index

Some are for visual styling elements, some for layouting elements, some for very specific situations.

A

- abs()
- accent-color
- acos()
- :active
- additive-symbols (@counter-style)
- ::after (:after)
- align-content
- align-items
- align-self
- align-tracks
- all
- <an-plus-b>
- <angle>
- <angle-percentage>
- animation
- animation-composition
- animation-delay
- animation-direction
- animation-duration
- animation-fill-mode
- animation-iteration-count
- animation-name
- animation-play-state
- animation-timeline
- animation-timing-function
- @annotation
- annotation()
- :any-link
- appearance
- ascent-override (@font-face)
- asin()
- aspect-ratio
- atan()
- atan2()
- attr()

B

- ::backdrop
- backdrop-filter
- backface-visibility
- background
- background-attachment
- background-blend-mode
- background-clip
- background-color
- background-image
- border-block-start
- border-block-start-color
- border-block-start-style
- border-block-start-width
- border-block-style
- border-block-width
- border-bottom
- border-bottom-color
- border-bottom-left-radius
- border-inline-style
- border-inline-width
- border-left
- border-left-color
- border-left-style
- border-left-width
- border-radius
- border-right
- border-right-color

How?

We explain 5 to 6 topics about CSS

- First
  - Short introduction to the subject
- Then
  - Example code, to show how it is done
- Finally
  - Try to complete the assignment on your own

01 - CSS selectors

To indicate which HTML elements you want to style you have to **make a selection**. You do this by writing a **CSS selector**.

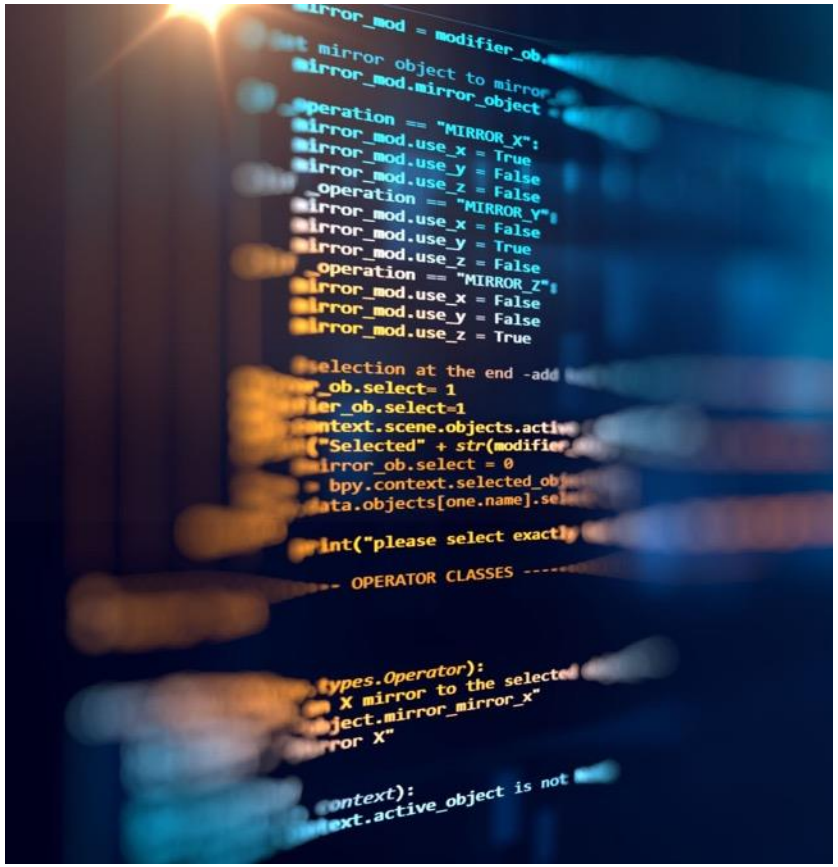
There are a number of commonly used selectors, some of which you are going to learn:

Element (or type) selector

- Selects all HTML elements of a given type, for example (all <h1>, or <p> or other elements)

Class selector

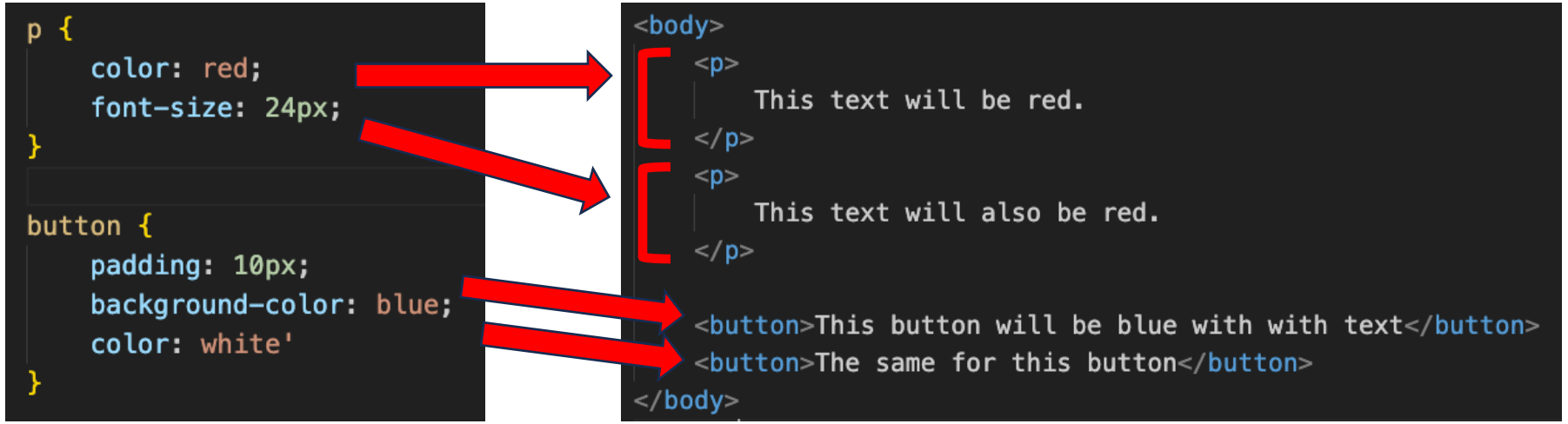
- Select all HTML elements that have a specific class attribute



The element (or type) selector

Selects all elements of the given type within a document.

Example:



CSS HTML

Exercise: 01a - CSS Type Selectors

The class selector

The **class** selector selects a HTML elements with the **class** (attribute) value.

The class selector starts with a **dot** followed by the name of the class

```css
.big-red {
    color: red;
    font-size: 80px;
}

.small-underlined {
    font-size: 10px;
    text-decoration: underline;
}
```

```html
<body>
    <p class="big-red">Will be big and red</p>
    <button class="big-red">Als big and red</button>

    <p class="small-underlined">Small and underlined</p>
    <h2 class="small-underlined">Also small and underlined</h2>
    <h5 class="small-underlined">Also small and underlined</h5>
    <button class="small-underlined">Also small and underlined</button>
</body>
```

These elements all have the same **class**. We can style them all at once with the **class selector in CSS**

Exercise: 01c - CSS class Selectors

The descendant selector

This will select elements only if they are a **child of** another element

Make al **<p>** element red but only if they are a **child** of a **<section>** element

Use a **space** between the parent and the child element

```
section p {
    color: red;
}
```

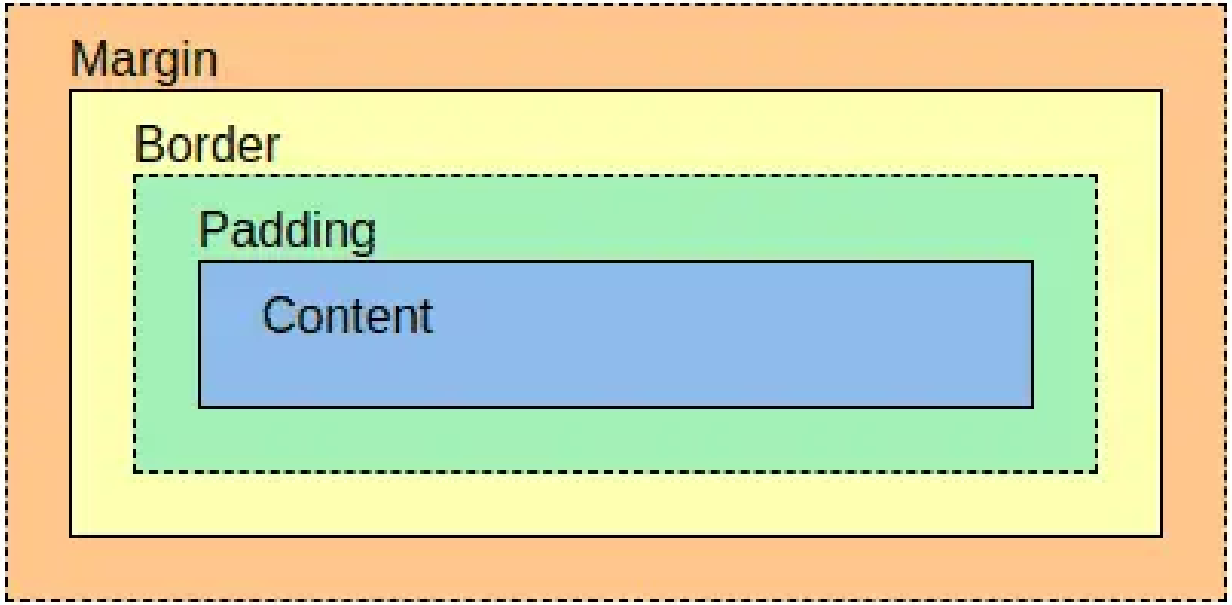
This <p> is a child of the <section> element

```
<body>

    <section>
        <p>
            This text will be red, because it
            is a child of a section element
        </p>
    </section>

</body>
```

02 – Box Model

Every HTML element is a box that is consists of:

- Margin.
- Border
- Padding
- Content



This is important to know.
It will provide you with the ability to position and layout elements on a page

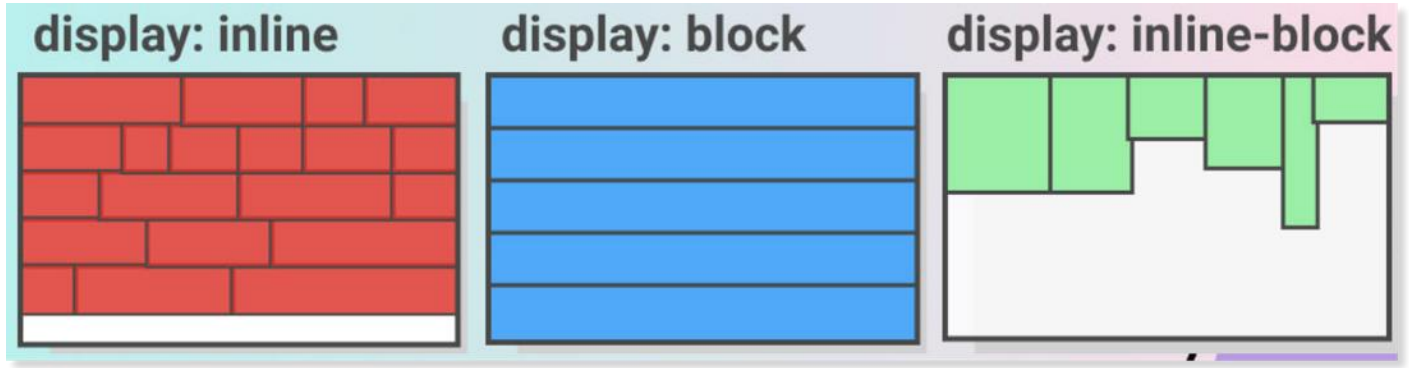
03 – Element flow and display

Words in a sentence flow until they reach the end of a line and then continue on the next line.

HTML elements flow differently depending on the element that is being used

- **Block** elements take up all the width that is available
- **Inline** elements flow like words and do not have a specific width/height to style
- **Inline block** elements flow like words but **do have** a width/height and margin

With the CSS **display** property you can manipulate the flow and layout of HTML elements.



https://learntheweb.courses/topics/css-layout-cheat-sheet/

04 – Sizes & units

You can specify the **size** of elements or **sizes** of fonts in different units.

For **absolute** units you use **pixels** (px)

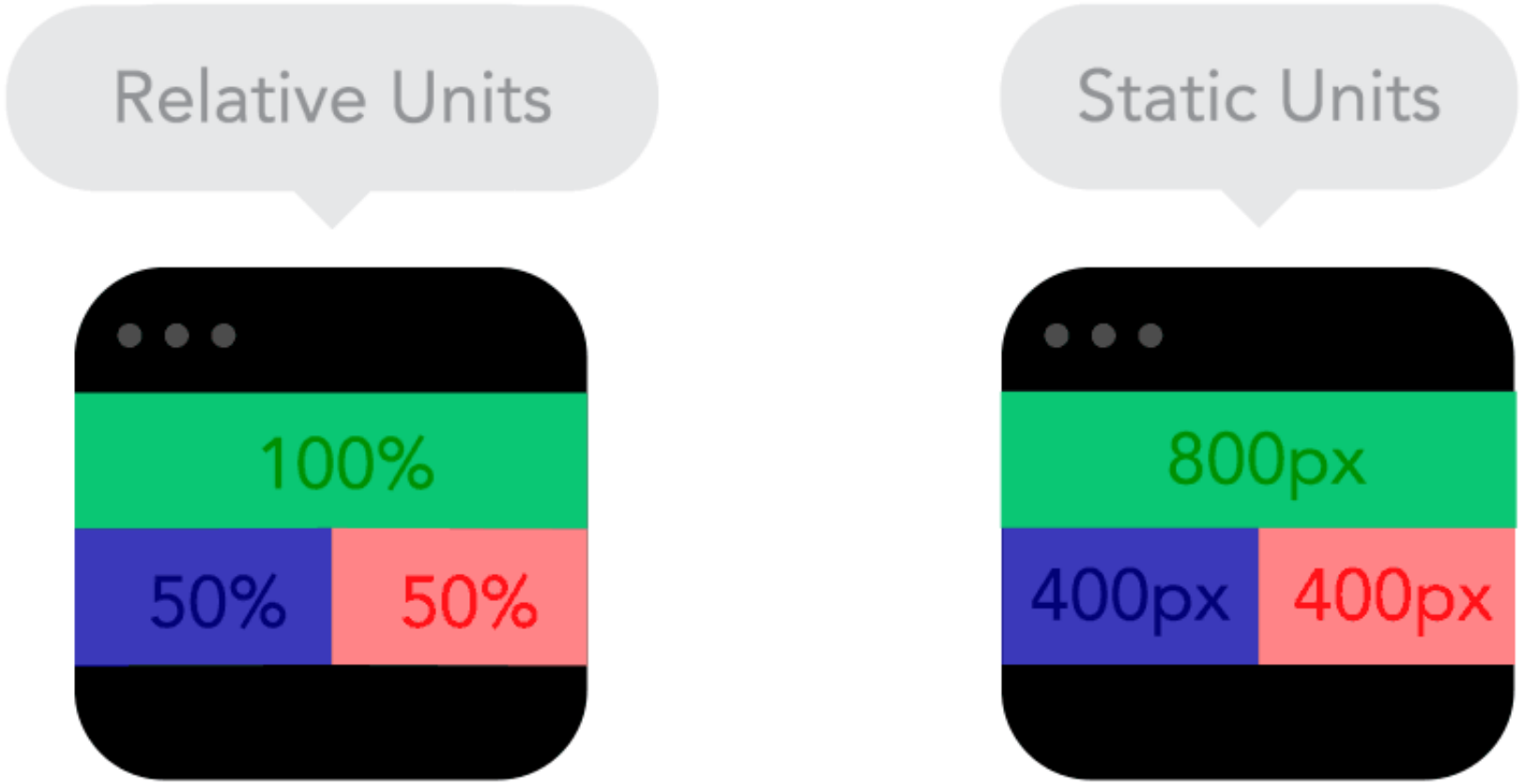
```
div {
  width: 300px;
  height: 100px;
}
```

For **relative** units use a **percentage** (%)

```
div {
  width: 50%;
}
```

Relative units are important if you want to make your website responsive.

04 – Sizes & units



05 – Colors in CSS

Colors can be specified in different ways in CSS

By specifying a color name:
This is a fixed list of color names

```
div {
  color: white;
  background-color: darkblue;
}
```

White and dark blue

By the HEX color code:
Basically the Red, Green and Blue values in hexadecimal notation.
(Tip: search for "hex color" in Google)

```
h1 {
  color: #b71540;
}
```

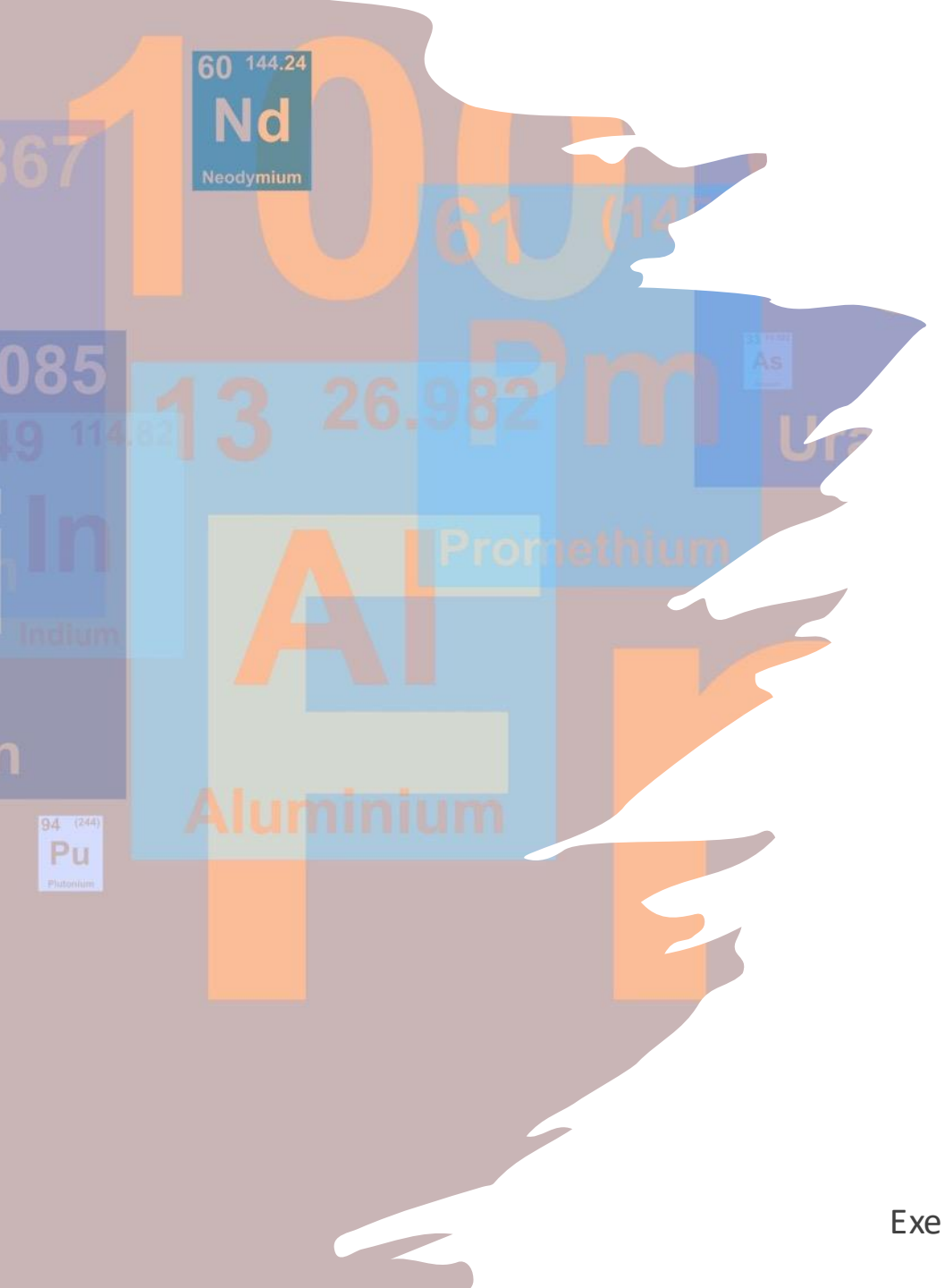
Color #b71540

By specifying the red, green, blue values (RGB)

```
h1 {
  color: rgb(13,221,154);
}
```

Color rgb(13,221,154)

Exercise: 05 - Colors



06 – Fonts & typography

With the right font and typography everything will look better.

Think about font-size, line-height, and margins to achieve consistent typography

Be aware:

- Only a few standard fonts are available on everyone's computer
- Arial, Helvetica, Times New Roman, Courier etc.

If you need a special font, use web fonts

- Google Fonts - free webfonts to use
- Own font (need license)

06 – Fonts & typography

```css
/** Include the "satisfy" webfont from Google */
@import url('https://fonts.googleapis.com/css2?family=Satisfy&display=swap');

h1 {
    font-family: Helvetica, Arial, sans-serif;
    font-size: 24px;
    font-weight: bold;
    text-decoration: underline;
    text-shadow: 1px 1px black;
    text-transform: uppercase;
    color: lightcoral;
    line-height: 36px;
    letter-spacing: 2px;
}

.special-font {
    font-family: 'Satisfy', cursive;
    text-transform: none;
    text-decoration: none;
    font-size: 36px;
    color: turquoise;
}
```

```html
<h1>What can you do with fonts?</h1>
<h1 class="special-font">This will be in the Staisfy webfont from Google</h1>
```

Exercise: 06 – Fonts / typography

09 – Images

Images can be added to your HTML with the <img> element and a src attribute pointing to your image file.

With CSS you can use an image as the background for an HTML element

You can (for example) add content on top of it and make beautiful designs

```
<section>
    <h1>I'm on top of the image</h1>
    <p>The image is in the background</p>
</section>
```

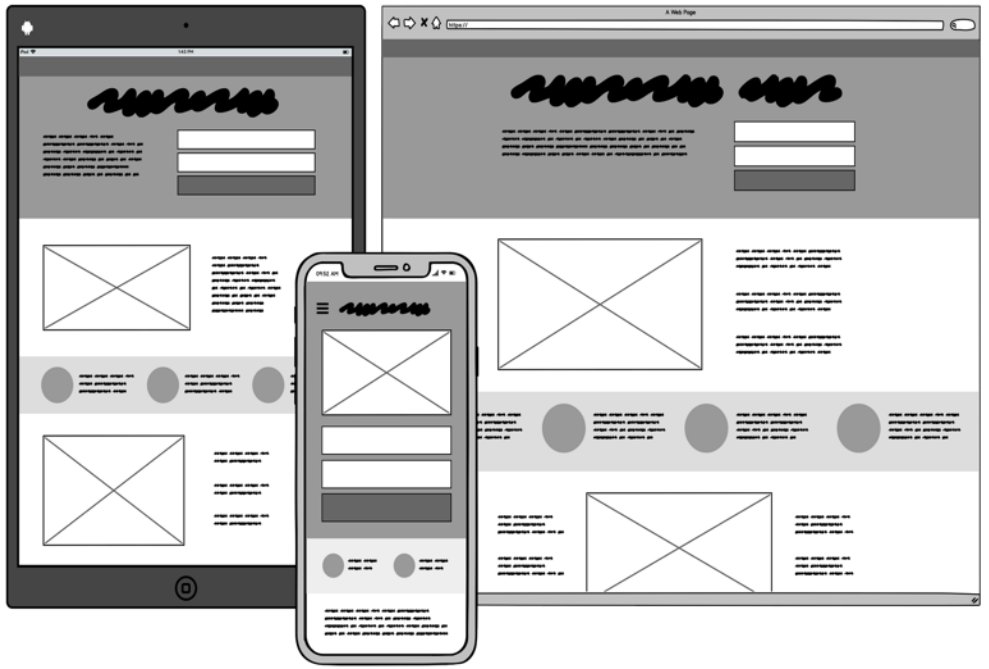
```
section {
    width: 400px;
    height: 300px;
    background-image: url(photo.jpg);
    background-size: cover;
}
```

Lunch break

Wireframes

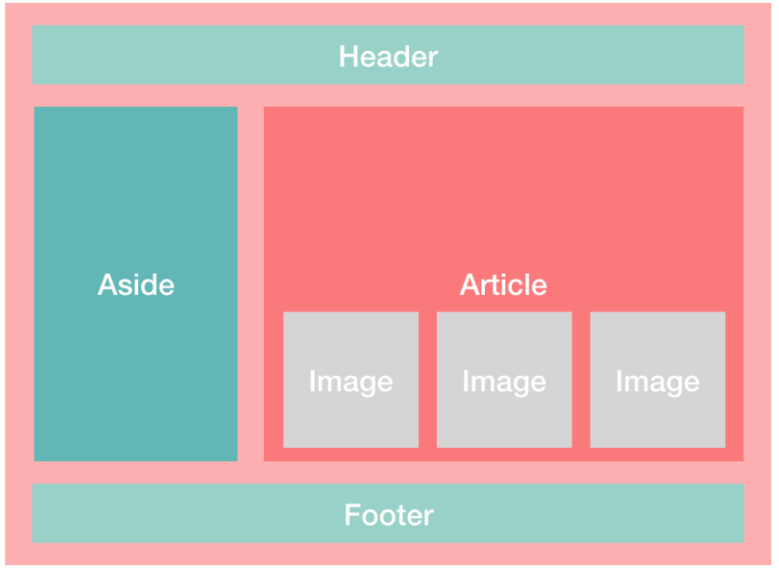
A wireframe is a schematic, a blueprint, useful to help you to think and communicate about the **structure** of the software or website you're building

https://balsamiq.com/learn/articles/what-are-wireframes/



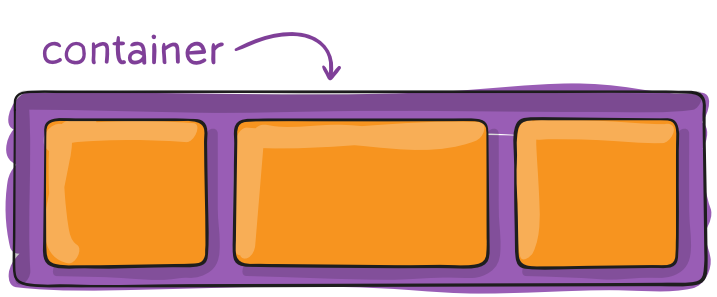
What is Flexbox?

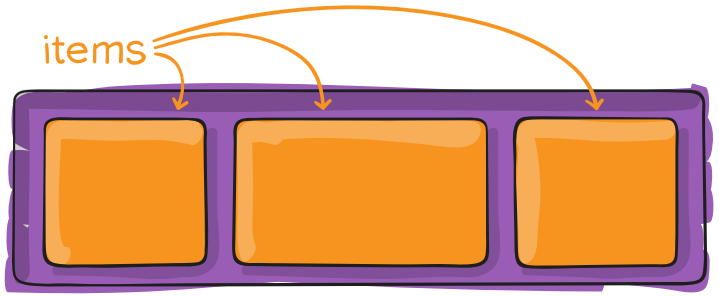
- Flexbox is a CSS layout module that provides an efficient way to arrange and align elements within a container.
- Think of Flexbox as a powerful tool that helps you organize and control the positioning of elements on a web page.



Flex Container and Flex Items

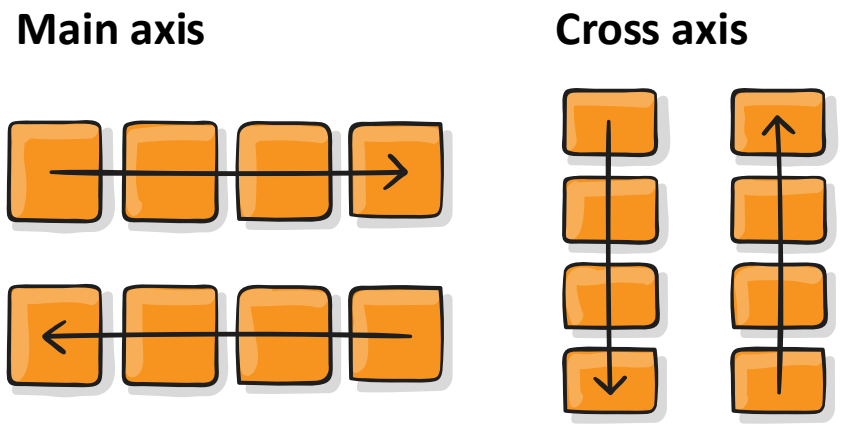
- In Flexbox, we have two main components: the **flex container** and **flex items**.
- The flex container is the **parent element** that holds the **flex items**.
- Flex items are the **child elements** within the **flex container**.
- In your CSS file set **display: flex** on an element and it will become a **flex container**
- All **direct child elements** of the element will automatically become **flex-items**





Aligning Flex Items

Flexbox allows us to easily **align** flex items **horizontally** and **vertically**.



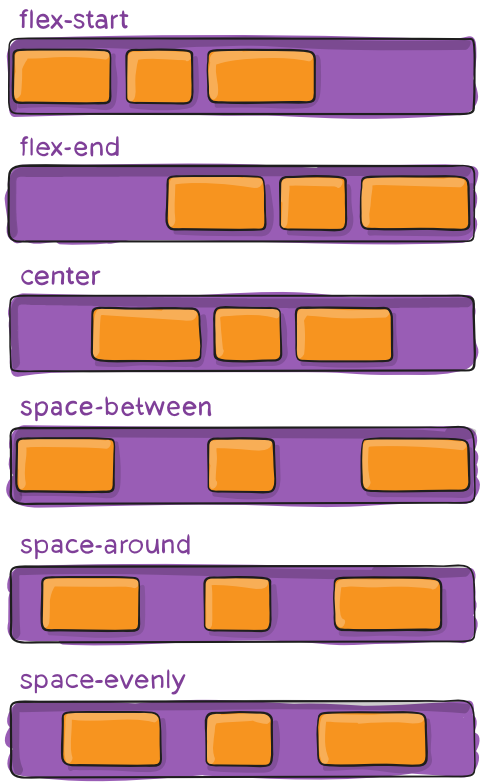
- For **horizontal alignment (main axis)**, use the property **justify-content**
- For **vertical alignment (cross axis)**, use the property **align-items**

Justify content (horizontal alignment)

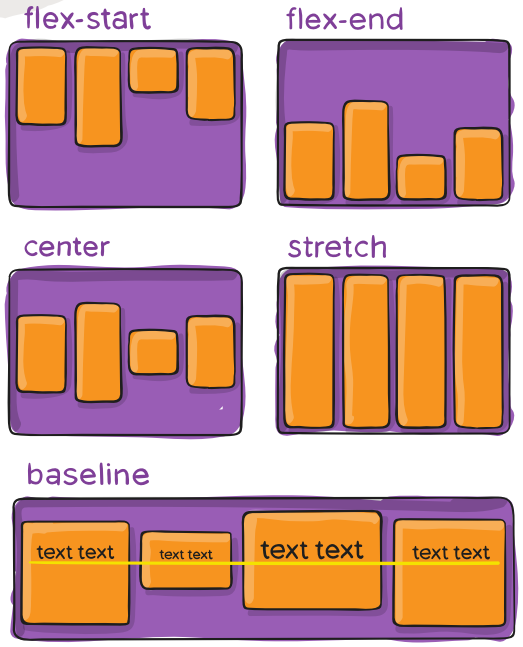
```css
section {
    display: flex;
    justify-content: flex-start;
    border: 1px solid black;
    gap: 10px;
}

aside {
    width: 100px;
    padding: 10px;
    background-color: green;
    color: white;
}
```

```html
<section>
    <aside>Some element 1</aside>
    <aside>Some element 2</aside>
    <aside>Some element 3</aside>
</section>
```

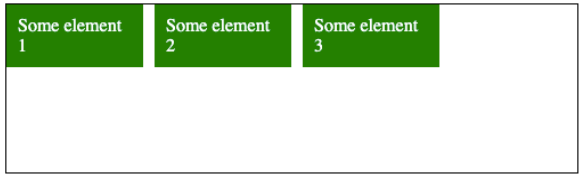
justify-content:



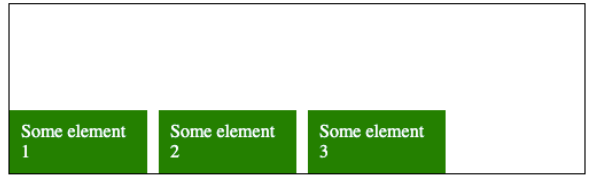
Align items (vertical alignment)



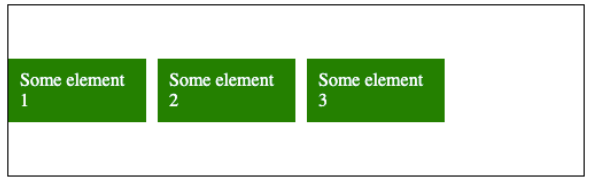
align-items: **flex-start;**



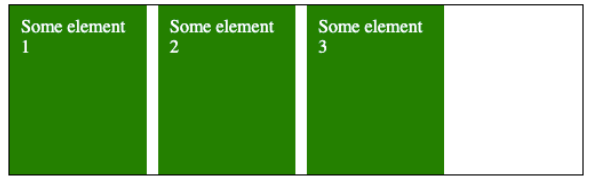
align-items : **flex-end;**



align-items : **center;**

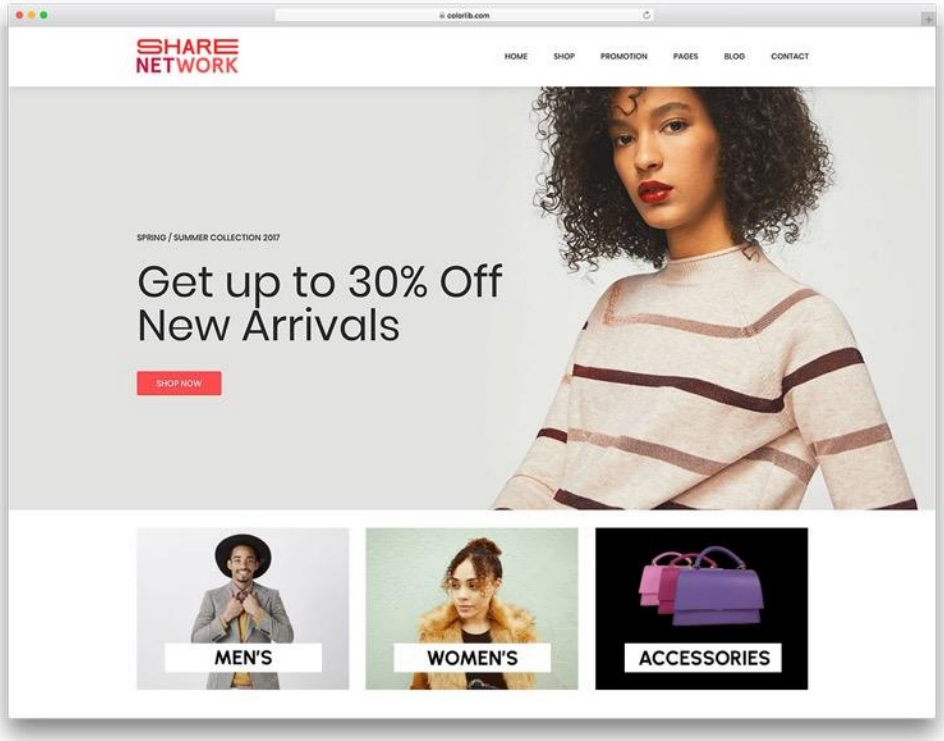


align-items : **stretch;**



Create a layout with Flexbox

1. Code along / Do the assignment
2. Or .. use the Flexbox layout technique in your website
   - Make a sketch / wireframe of your website layout
   - Think about how to use flexbox to layout the website
   - What elements belong together?
   - What elements are **flex-containers**
   - What elements are **flex-items?**



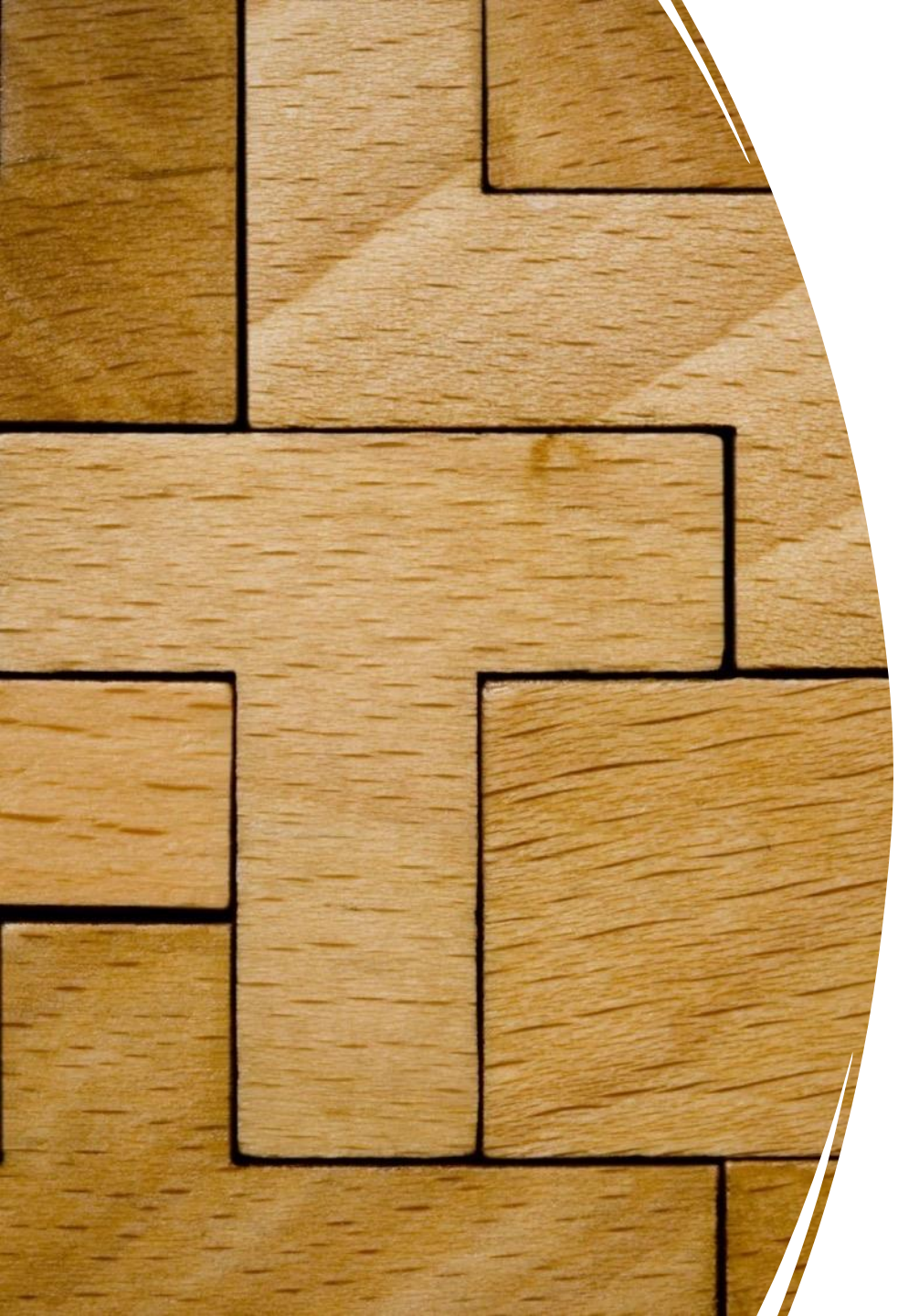
https://css-tricks.com/snippets/css/a-guide-to-flexbox/

SHARE NETWORK

HOME
SHOP
PROMOTION
PAGES
BLOG
CONTACT

SHARE NETWORK

HOME SHOP PROMOTION PAGES BLOG CONTACT



Good tutorials on Flexbox

- https://css-tricks.com/snippets/css/a-guide-to-flexbox
- https://flexboxfroggy.com
- https://flexbox.io
- https://flexboxzombies.com
- https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

Google: search for “Flexbox CSS Tutorials”

Inspiration Day 3

Last Inspiration Day

Introduction to responsive CSS

Introduction to interactivity

Get help with your website

Get support for HTML/CSS questions

Work on (and finish) your website / project

Questions/ Feedback?