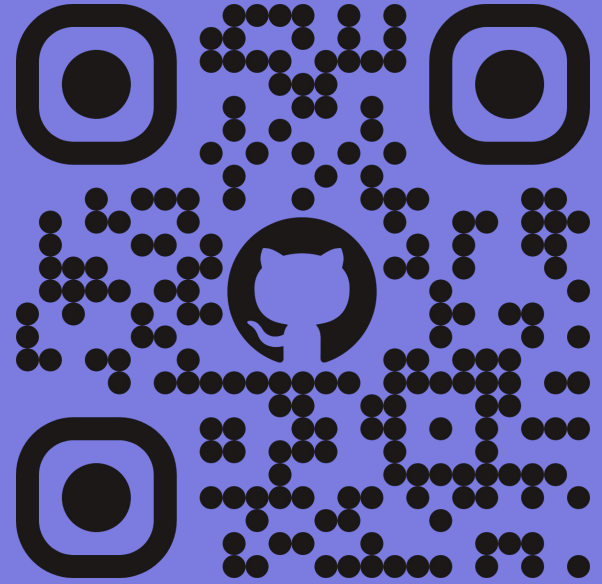


HTML & CSS 101

Second Lecture

By Yordan G. Yordanov 2024

ALL MATERIALS CAN BE
FOUND IN MY GITHUB:
github.com/YorDN



WHAT WE LEARNED SO FAR:

`<p></p>` - **paragraph** element (used for text)

`<h1..6></h1..6>` - **h1** to **h6** element (used for titles)

`<a>` - **anchor** element (used for links)

`` - **image** element (used for images)

`<button></button>` - **button** element

WHAT WE LEARNED SO FAR:

EVERYTHING IS A CONTAINER!

`<div></div>` - block element container used to store data

`` - inline element container used to store data just like the div

WHAT WE LEARNED SO FAR:

Inline vs Block element

- Inline items take as much space in a row as they need

Lorem (this is a inline element)

- Block elements take up the whole row

Lorem

(this is a block element)

WHAT WE LEARNED SO FAR:

CSS structure:

```
h1 {  
  margin-bottom: 20px;  
  font-size: 24px;  
  color: ☐ #333;  
}
```

h1 - the element we want
to access
margin, font.. - properties

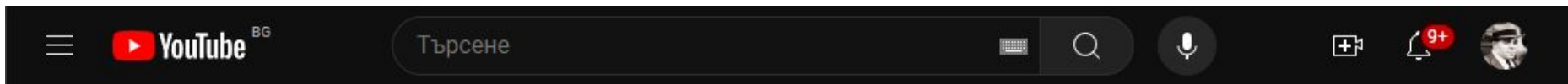
TODAY WE WILL DISCUSS:

- **The structure in a website and how to implement it**
 - Position relative/absolute
 - header/footer/main
- **Buttons**
 - Padding
 - Hover, Active states
 - Pseudo classes
- **Flexbox in detail**
- **Misc features**

THE STRUCTURE OF THE GOOD WEBSITES

THE STRUCTURE OF THE GOOD WEBSITES

Every website should have a header:



The header should always be *stationary* – it should never move out of the screen. How do we do that?

With the **position** property!

THE STRUCTURE OF THE GOOD WEBSITES

position fixed - the element is fixed and no matter what stays in place

position absolute - same as position fixed, but the element is not fixed. Used in combination with position relative.

position relative - the element is always on top. Combined with position absolute can be used as for example a notification icon:



The bell is position absolute
The 9+ is position relative

THE STRUCTURE OF THE GOOD WEBSITES

With any position property we have to specify how many pixels(or any other system) away from the top,bottom,left or right the element should be.

The following element has top:20px, left: 100px, right: 0px:

Example

THE STRUCTURE OF THE GOOD WEBSITES

In HTML we have

`<header></header>`, `//for header`

`<main></main>`, `//for the main content`

`<footer></footer>` `//for the footer`

attributes. Note that they are no different than any other div element and are meant for readability purposes so it's a good practise to have them.

BUTTONS

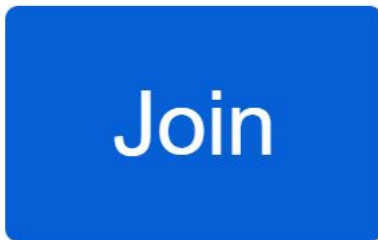
BUTTONS

We are going to create a button based on our knowledge from the past lecture and expand it even further!

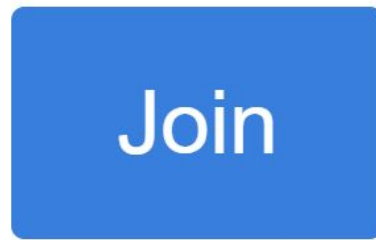
The button will look like this:



default



when hovered



when clicked

BUTTONS

First we create the button as we know and give it a class:

```
<button class = "button">Sign in</button>
```

And that's it for the HTML part now CSS!

First we change the background color and the color of the text:

```
.button{  
  background-color: white;  
  color: rgb(6, 95, 212) ;  
}
```

BUTTONS

But our button looks like this:



We want to change the border's width, color and style. We do that with the border property:

```
.button{  
  background-color: ■ white;  
  color: ■ rgb(6, 95, 212) ;  
  border: solid ■ rgb(6, 95, 212) 1px;  
}
```

solid stands for the style of the border (notice how our border is two-colored), the second item is the color of the border and the third is the thickness of the border

BUTTONS

Our button looks now like that:



But the final design is:



We want more space in the button. We do that with padding. Think of padding like margin, but on the inside – how much space the button has between the border and the text.

BUTTONS

We add the padding and this is the result:

```
.button{  
  /* ... */  
  padding-left: 15px;  
  padding-right: 15px;  
  padding-top: 9px;  
  padding-bottom: 9px;  
}
```



BUTTONS

Now we only have to round the corners. That happens with the border-radius property:

```
.button{  
  /* ... */  
  border-radius: 2px;  
}
```

the pixels is how much the corners should be rounded

If we want the cursor to be pointer when the button is hovered we use the cursor property

```
.button{  
  /* ... */  
  border-radius: 2px;  
  cursor: pointer;  
}
```

BUTTONS - PSEUDO CLASSES

The last thing we need to do is to change the colors of the button when hovered and when activated. This can be easily made with pseudo classes.

Pseudo classes tell the computer to do something when the specific state is achieved. We will discuss 2 pseudo classes today: `:hover` & `:active`

BUTTONS - PSEUDO CLASSES

:hover - does something when the mouse hovers the item

:active - does something when the mouse clicks the item

With that in mind we can add the classes:

```
.button:hover{  
  background-color:  rgb(6, 95, 212);  
  color:  white;  
}  
.button:active{  
  opacity: 0.8;  
}
```

BUTTONS - PSEUDO CLASSES

Finally to smoothen the animation we can use the transition property in the .button class. We tell what property we want to transition and for how many seconds. If we have more than just a one property to transition we separate them by coma:

```
.button{  
    /*...*/  
    transition: background-color 0.15s, color 0.15s;  
}
```

And just like that the button is finished!!!

FLEXBOX IN
MORE DETAIL

WHAT IS FLEXBOX USED FOR?

Last time we looked at flexbox in a very simple matter. Now we are going to explain what it is used for!

Flexbox is used mainly for 2 reasons:

1)To center an element/s

2)To flex an element based on the screen's size

HOW TO CENTER AN ELEMENT USING FLEXBOX

To center an element vertically we use:

```
align-items: center;
```

To center an element horizontally we use:

```
justify-content: center;
```

Now these properties can have many other values other than center, but for now that's enough.

HOW TO FLEX AN ITEM USING FLEXBOX

To flex an item using flexbox we use the *flex property* given to us by the flexbox.

To flex an item we can use `flex:1;`

Flexing an element is useful when we want to optimise our website for mobile devices. For example we want our search bar to stretch based on the user's screen.

USEFUL FEATURES OF CSS

USEFUL FEATURES OF CSS - THE STRONG ELEMENT

One useful feature in css is the `` element.

It gives you the ability to **style a single word or a single part of the text**. Meaning that a single word can be managed apart from the rest of the text.

The strong element in particular gives the text that is inside the element a font-weight of bold (the text is bold), but we can do anything really to it.

USEFUL FEATURES OF CSS - THE STRONG ELEMENT

Let's assume we have a paragraph element and some text in it. We add a strong element in it like this:

```
<body>
  <p style="font-family: Arial, Helvetica, sans-serif;">
    This is an example of a <strong>strong</strong> element
  </p>
</body>
```

And we get that:

This is an example of a **strong** element

USEFUL FEATURES OF CSS - THE BACKGROUND-IMAGE PROPERTY



The background-image property of CSS gives us the ability to as the name implies to make the background of an element to an image:

```
<style>
  .div{
    background-image: url("Экранна\ снимка\ 2024-09-24\ 084847.png_");
  }
</style>
<div class="div">
  This is a div
</div>
```

This is a div

USEFUL FEATURES OF CSS - THE BACKGROUND-IMAGE PROPERTY

Or we can use it to add gradient to the background(set the background color to a gradient). Today we will see only linear gradient as it is the most popular.

```
.div{  
  background-image: linear-gradient(to bottom right,  red,  blue);  
}
```

First we specify the direction of the gradient, then we list the colors - we can have unlimited colors

USEFUL FEATURES OF CSS - THE BACKGROUND-IMAGE PROPERTY

We get the following result:



This is a div

USEFUL FEATURES OF CSS - THE BEFORE AND AFTER PROPERTY

In CSS we can add an element before or after an element. This element is entirely created in CSS.

The `::before` property is a pseudo class and it creates a virtual div before the element. The div acts like it has position relative - we have to specify how many pixels it is from the top, bottom, left and right of the element.

The `::after` element is like the `::before` element, but it puts the element after.

HOMework
TIME 🎉

HOMEWORK:

Create the following button based on the knowledge you acquired today. Use flexbox to **center the button** both horizontally and vertically. The button does not have a border so use **`border:none;`**. For the color use **red**. Make the **cursor pointer** and all transitions should be **0.15s!**



default



hovered



active

THANK YOU FOR LISTENING!

Yordan G. Yordanov