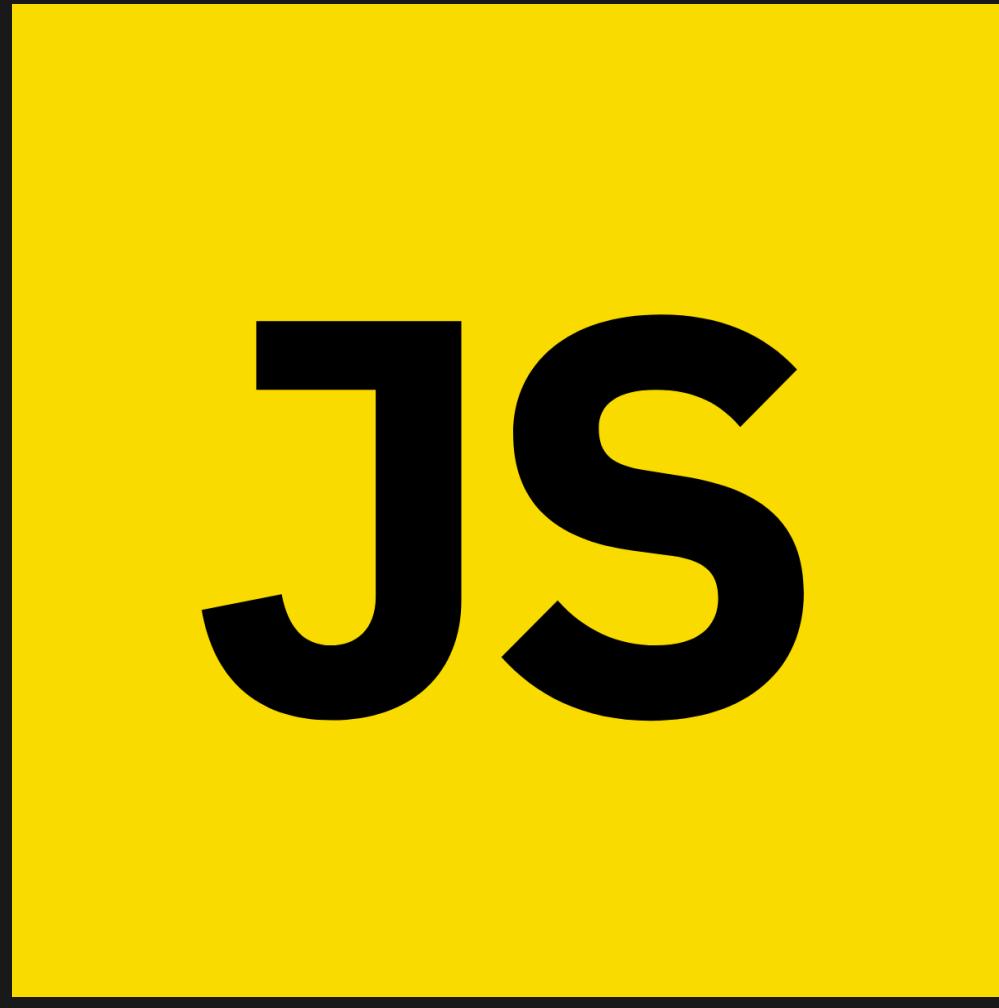




Intro to JS

→ **Lecture 0**



Gulnaz Serikbay

in!

JS





→ VISION

- if you have a problem, Google it first, then come to a mentor
- do not reinvent the wheel, use boilerplate components
- release early, release often
- copy-pasting is fine as long as you understand the code
- you need to have an idea by the end of week 1
- you need to have a fully working project by 1st of August



Gist

- Style: Tailwind CSS
- HTML DOM
- JS events + HOF



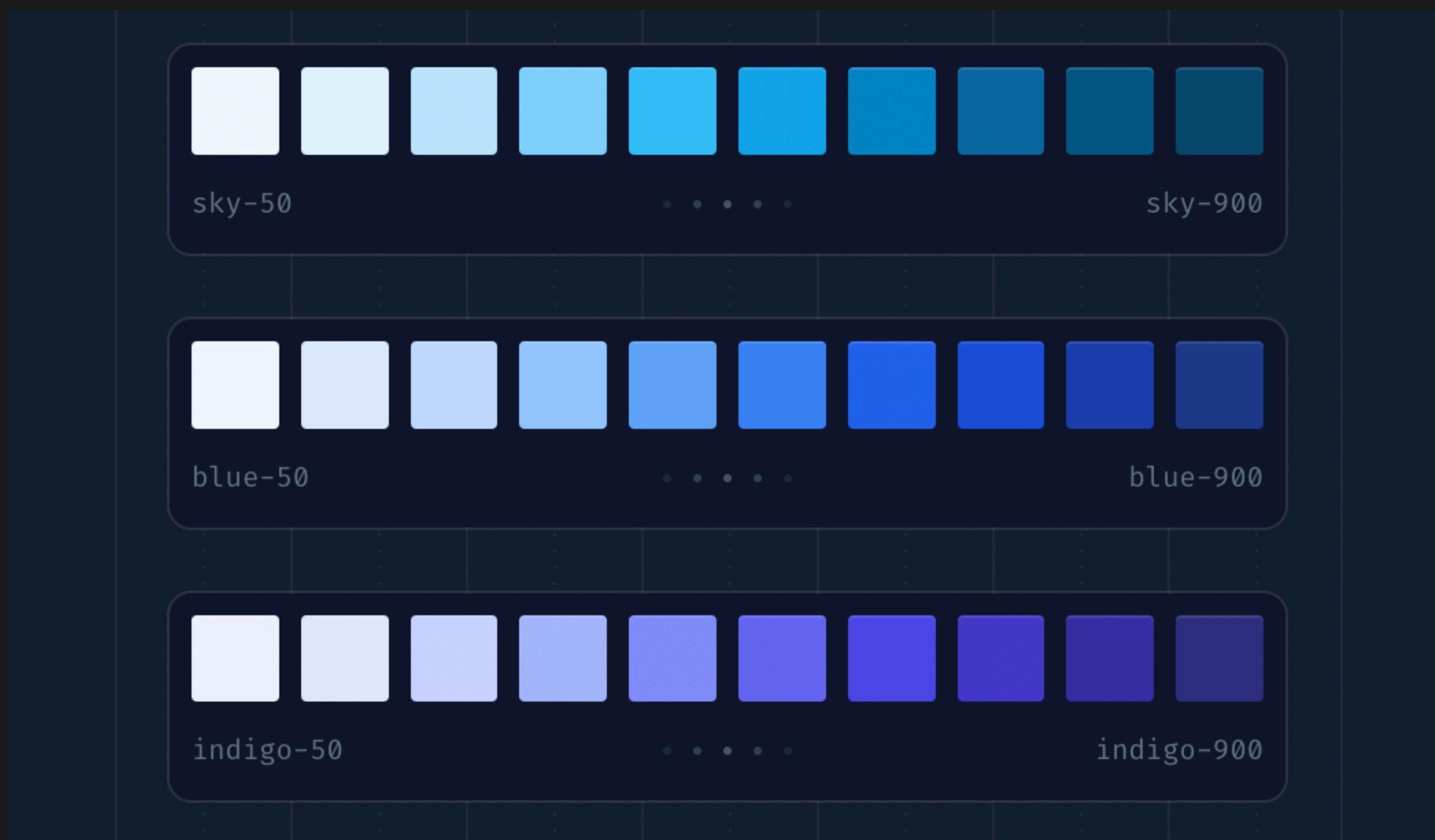
Все же знакомы с
HTML & CSS??

→ Tailwind CSS: **compact** CSS

- Just use classes to style
- Create your own classes with certain style
- Responsive design
- Tailwind automatically removes all unused CSS when building for production.
<10 kB of CSS to the client
- Compatible with React, Angular, Vue, etc etc

in!

Tailwind CSS example



```
<div class="grid grid-cols-10 gap-2">
  <div class="bg-sky-50 aspect-square"></div>
  <div class="bg-sky-100 aspect-square"></div>
  <div class="bg-sky-200 aspect-square"></div>
  <div class="bg-sky-300 aspect-square"></div>
  <div class="bg-sky-400 aspect-square"></div>
  <div class="bg-sky-500 aspect-square"></div>
  <div class="bg-sky-600 aspect-square"></div>
  <div class="bg-sky-700 aspect-square"></div>
  <div class="bg-sky-800 aspect-square"></div>
  <div class="bg-sky-900 aspect-square"></div>
</div>
```

in!

A screenshot of the Visual Studio Code interface demonstrating Tailwind CSS IntelliSense. The code editor shows a snippet of Tailwind CSS utility classes applied to an element. A tooltip is open over the class `bg-t`, listing various background color utilities: `bg-transparent`, `bg-teal-50`, `bg-teal-100`, `bg-teal-200`, `bg-teal-300`, `bg-teal-400`, `bg-teal-500` (which is highlighted), `bg-teal-600`, `bg-teal-700`, `bg-teal-800`, `bg-teal-900`, and `bg-top`. The background color of the tooltip matches the selected `bg-teal-500` utility.

```
1 <div class="w-full flex items-center justify-between block p-6 space-x-6">
2   <div class="flex-1 truncate">
3     <div class="flex items-center space-x-3">
4       <h3 class="text-gray-900 text-sm font-medium truncate">Jane Cooper</h3>
5       <span class="text-teal-600 bg-t">Admin</span>
6     </div>
7     <p class="mt-1 text-gray-500">
8       
9     </div>
10    <div class="border-t border-gray-300 mt-px flex">
11      <div class="w-0 flex-1 flex items-center justify-between py-4 text-sm text-gray-500 focus:outline-none duration-150">
12        <a href="#" class="relative inline-block w-fit h-fit bg-transparent">
```

unsplash.com/photo-1494790108377-be9c29b29330?w=256&h=256&fit=crop

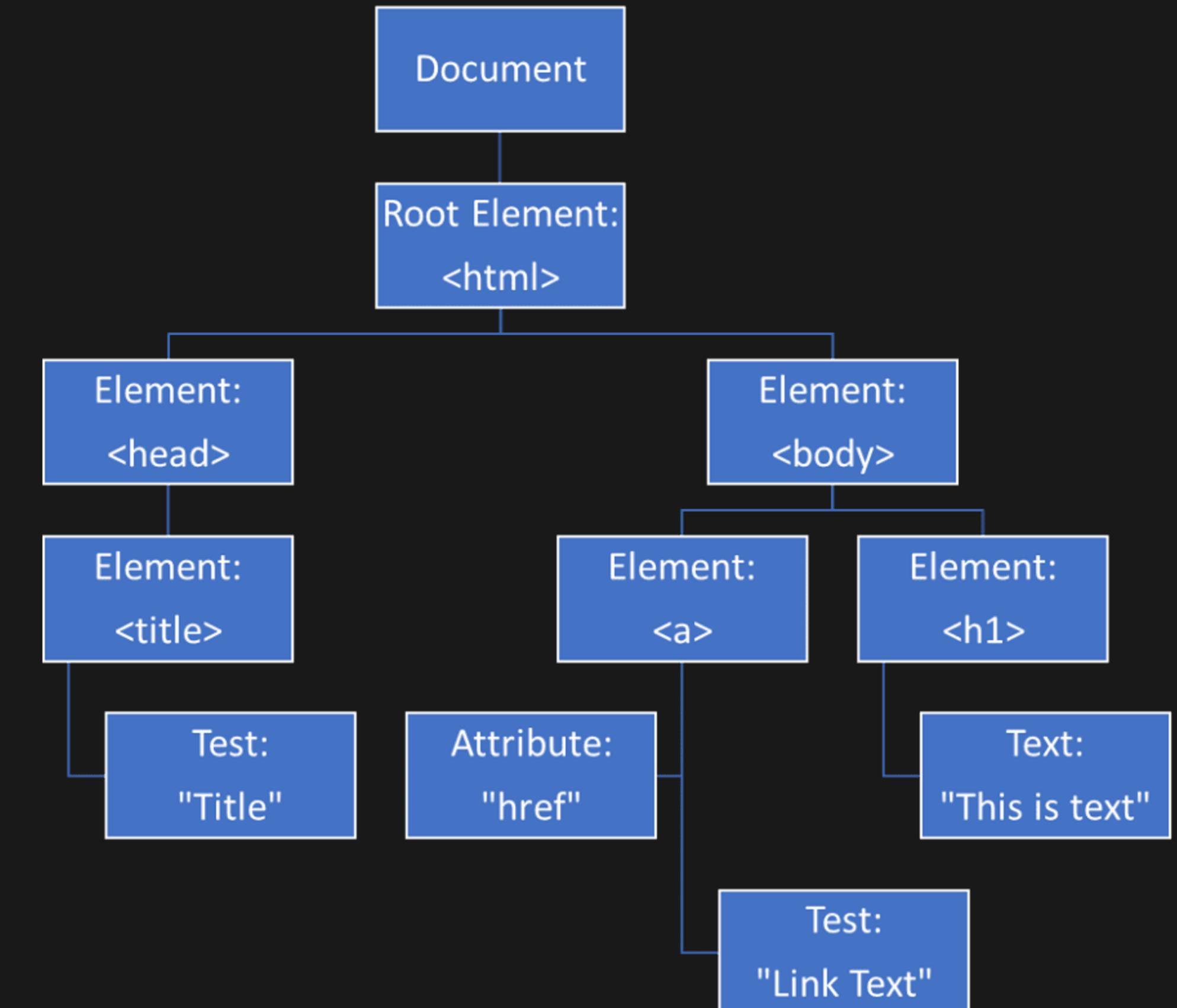
-center py-4 text-sm text-gray-500 focus:outline-none duration-150

 tailwindcss
INTELLISENSE

 Visual Studio Code

in!

→ HTML DOM



Visualizer: <https://codepen.io/pavlovsk/pen/QKGpQr>



```
15  <!DOCTYPE html>
16  <html>
17  <body>
18
19  <h1 id="title">Hello World!</h1>
20  <div class="content">
21    <p>Paragraph 1</p>
22    <p>Paragraph 2</p>
23  </div>
24
25  </body>
26  </html>
```

```
// Get the <div> element with class "content"
let div = document.querySelector('.content');

let firstParagraph = div.firstElementChild;
console.log(firstParagraph.textContent); // Logs "Paragraph 1"
```

```
let parentOfSecondParagraph = secondParagraph.parentNode;
console.log(parentOfSecondParagraph.className); // Logs "content"
```

```
let title = document.getElementById('title');
console.log(title.textContent); // Logs "Hello World!"
```



Latest from nFactorial Podcast



#22 | Айжан Кул-Мухаммед: Как поступить в лучший универ в США и что делать дальше?

932 views • 11 hours ago

in! nFactorial Podcast

Мальчишки и девчонки, а так же их родители, полезные советы услышать, не хотите ли? С нами Айжан Кул-Мухаммед, ...

New

#21 | Мурат Абдрахманов, супер-ангел: Уже инвестировал \$15 млн в 35 стартапов!

2.1K views • 2 days ago

in! nFactorial Podcast

Сегодня с нами Мурат Абдрахманов – человек, которого сложно найти, легко потерять и невозможно забыть. А ещё его в ...

New



Show must go on! | Физмат тогда и сегодня | Какие уроки извлекли из предпринимательског... 31 chapters ▾

JS events

```
document.addEventListener("DOMContentLoaded", function() {  
    // do some magic when the page is loaded  
});
```

in!

event \Rightarrow "submit"
"mouseover"
....


```
1 element.addEventListener("click", function() {  
2   document.getElementById("idName").textContent = "Dalida is queen";  
3 });  
|
```



function that handles
the event

in!

function is called
})

event

```
1 <input type="checkbox" id="myCheck" onmouseover="myFunction()";  
2  
3 <script>  
4 function myFunction() {  
5   document.getElementById("myCheck").click();  
6 }  
7 </script>
```

simulating mouse click =>

in!

```
.example {  
    font-size: 30px;  
    color: purple;  
}
```

```
9 | function myFunction() {  
10|     document.getElementById("myElement").classList.add("example");  
11| }
```

get HTML
element

"example" is
added as a
class

in!

event handling within HTML

```
<body>
    <button onclick="myFunction()">Append</button>
```

```
<script>
function myFunction() {
    const paragraph = document.createElement("p");
    const node = document.createTextNode("This is a paragraph.");

    paragraph.appendChild(node);
    document.body.appendChild(paragraph);
}
</script>
```

```
</body>
```

new <p> is created
with each click to "Append" button

MORE

```
1 document.querySelector(".myClass");
2 document.querySelectorAll(".myClass");
3 document.createElement("p");
4 parent.appendChild(child);
5 parent.removeChild(child);
6 element.classList.remove("myClass");
7 element.classList.toggle("myClass");
8 element.innerText = "my text";
9 element.style.color = "red";
10 element.setAttribute("myAttribute", "myValue");
```

in!

⇒ localStorage

```
localStorage.setItem('key', itemToStore)
```

```
localStorage.getItem('key')
```

Note: JSON conversion may be needed

```
1 var myObj = { "name": "John", "age": 31, "city": "New York" };
2 var myJSON = JSON.stringify(myObj);
3 Var myObj = JSON.parse(myJSON);
4
```

in!

⇒ sessionStorage

data in `localStorage` doesn't expire, data in `sessionStorage` is cleared when the *page session* ends

in!

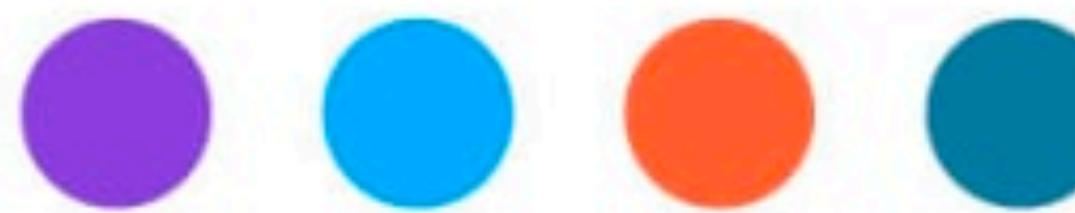
→ HOF (higher order
functions)

- Higher-order function is any function that takes other function as argument or returns another function.
- `map` – transforms one array to another
- `filter` – filters out an array given a **bool** function
- `reduce` – computing **single value** from whole array (e.g. sum of all elements)
- `forEach` – performs action to each element in array [**map copycat:**]

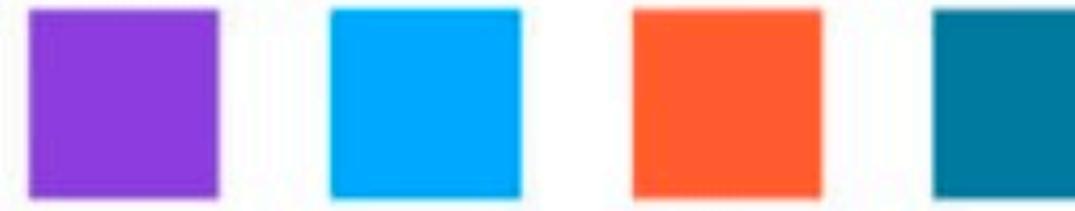
in!



`map(callable)`



`filter(callable)`



`reduce(callable)`





in!

- MAP -

```
1 let numbers = [1, 2, 3, 4, 5];
2
3 let doubled = numbers.map(function(num) {
4   return num * 2;
5 });
6
7 console.log(doubled); // [2, 4, 6, 8, 10]
8
```

```
1 let numbers = [1, 2, 3, 4, 5];
2
3 let doubled = numbers.map(num => num * 2);
4
5 console.log(doubled); // [2, 4, 6, 8, 10]
6
```

Arrow functions!



→ Homework: ToDoList



Todo List

Have fun

Delete

Spread empathy

Delete

Generate value

Delete

in!

Your app should have these core features:

- 1. Adding Todo Items:** Users should be able to add new tasks to their list.
- 2. Removing Todo Items:** Users should be able to remove tasks when they are no longer necessary.
- 3. Checking Off Todos:** Users should be able to mark tasks as completed.

For an extra challenge, try using `localStorage` or `sessionStorage` to keep your TodoList persistent. This way, your users won't lose their tasks, even if they refresh the page! 🚀



06.06.23, 9AM



Raqmet :D

