# DOM & Events

## + basic operators, conditionals, and GitHub

Week 5

ReDI

# Agenda

1. DOM - recap
2. JS Events
3. Basic operators + conditions
4. Git & GitHub

# Document Object Model

DOM represents all page content as **objects** that can be modified.

Every HTML tag/element is an **object**.

Nested tags are **“children”**.

# Document Object Model

## 💡 What is an object?

```
const user = {
    name: “Anna”,
    address: “Amerikavej 70”,
    age: “29”,
};
```

# Document Object Model

## DOM can be seen in the Developer Tools.

Elements Console

```
<!DOCTYPE html>
<html lang="en">
 ▶<head> ··· </head>
 ▼<body>
  ▼<div class="container"> flex
   ▼<div class="card"> == $0
      <img src="https://images.unsplash.com/photo-14
      84723091739-30a097e8f929?ixlib=rb-4…8MHxwaG90b
      y1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fi
      t=crop&w=1547&q=80" alt="Image 1">
      <h2 id="card-title-one">Blueberry French Toast
      </h2>
      <p>This is the description for Card 1.</p>
      <button id="order-now-btn">Order Now</button>
     </div>
    ▶<div class="card">···</div>
    ▶<div class="card" id="card-3">···</div>
   </div>
   <script src="script.js"></script>
  </body>
</html>
```

Elements Console

top Filter

Default levels No Issues

```
> document.body
<· ▼<body>
    ▶<div class="container">···</div> flex
     <script src="script.js"></script>
    </body>
>
```

# Document Object Model

## Selecting elements in the DOM - recap

The methods **querySelector** and **querySelectorAll** are the most versatile - they can be used to select an element by a class name, an id, an HTML tag.

```
const el = document.querySelector("div.user-panel.main input[name='login']");
```

# Document Object Model

## 💡 What is a method?

```
objectName.methodName()
```

```
Date()
Date.now()

string.length
string.toUpperCase()
string.split(separator, limit)

number.toString()
```

And many more...

# Exercise

*10 minutes*

Write a function **uppercaseFirstLetter(str)**

that returns the string str with the uppercased first character:

```
uppercaseFirstLetter("iga") === "Iga";
```

# Break

# Document Object Model

## DOM node object properties and methods

### Styling:

```
className
classList
style
```

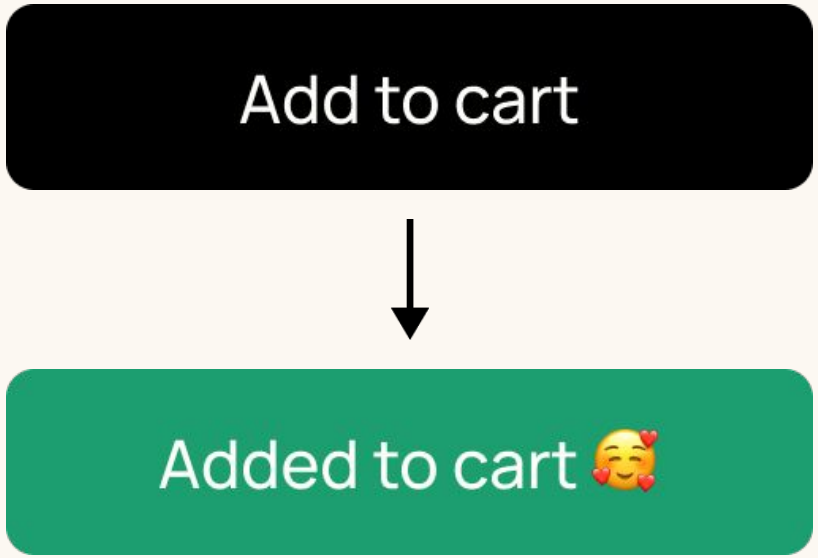
### Creating/removing elements:

```
document.createElement()
node.append/prepend()
node.before/after()
node.cloneNode()
node.remove()
```

# Exercise

*10 minutes*

1. Create this button using ONLY JavaScript.
2. Add styles using CSS.
3. On click change the background color and the text content of the button.
4. Optional - can you make the button change the text and style only for 10 seconds before it goes back to previous state?



# JavaScript events

An event is a signal that something has happened, for instance a user clicked a button, pressed a key, submitted a form.

# JavaScript events

**Mouse events:**
click, mouseover/out, mousedown/up

**Keyboard events:**
keydown/up

**Input events:**
input/change/focus/blur

**Form element events:**
submit

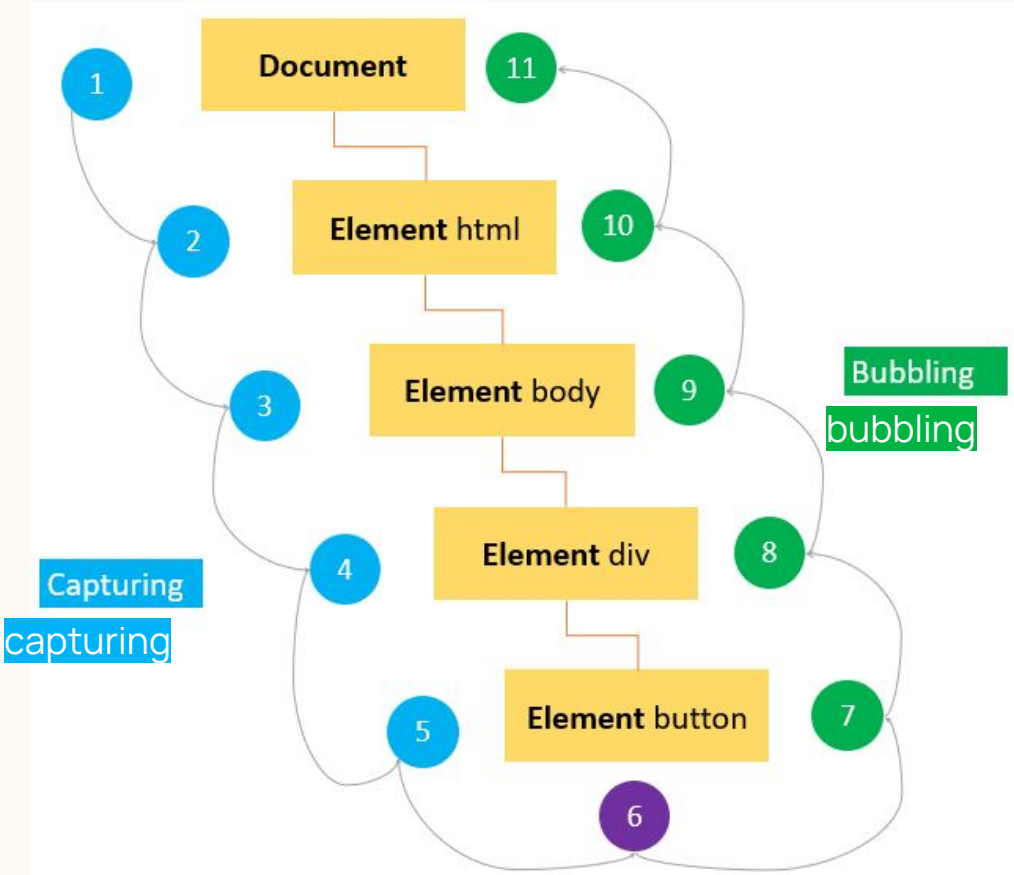
# JavaScript events

## Event flow

1. First, event capturing occurs, which provides the opportunity to **prevent** the event

2. Then, the actual target receives the event

3. Finally, event bubbling occurs, which allows a final response to the event




event.target

# JavaScript events

## Event object

All event objects in the HTML DOM are based on the Event Object. All events have access to the Event Object properties and methods.

```
let btn = document.querySelector('#btn');

btn.addEventListener('click',(event) => {
    console.log(event.type);
});
```

# JavaScript events

## Event listeners

To react on events we can assign a **handler** – a function that runs in case of an event.

A handler can be assigned in a:

| HTML attribute | DOM property | addEventListener() |
| --- | --- | --- |
| fx. input, button onclick | elem.on[event] | |

# JavaScript events

## Adding a handler in a HTML attribute

```
<div class="button-container">
  <button class="round-btn" id="decrement" onclick="alert('Click!')">-</button>
  <span id="count">0</span>
  <button class="round-btn" id="increment">+</button>
</div>
```

# JavaScript events

## Assigning a handler using addEventListener()

event name

event handler function

```
let btn = document.querySelector('#btn');

btn.addEventListener('click',(event) => {
    console.log(event.type);
});
```

# JavaScript events

## Example of mouse events

```
const countElement = document.getElementById("count");
const incrementButton = document.getElementById("increment");

incrementButton.addEventListener("click", () => {
  let count = countElement.innerHTML;
  count++;
  countElement.innerHTML = count;
});
```

```
const countElement = document.getElementById("count");
const incrementButton = document.getElementById("increment");

const increment = () => {
  let count = countElement.innerHTML;
  count++;
  countElement.innerHTML = count;
};

incrementButton.addEventListener("click", increment);
```

# JavaScript events

**What is the difference?**

**What happens in each case?**

```
incrementButton.addEventListener("click", increment);
```

```
incrementButton.addEventListener("click", increment());
```

# JavaScript events

## Example of form element events

stops page from reloading

```
const email = document.getElementById("email");
const carPreference = document.getElementById("car-preference");
const privacyPolicy = document.getElementById("privacy-policy");

const form = document.getElementById("newsletter-form");
form.addEventListener("submit", (event) => {
  event.preventDefault();
  console.log("email:", email.value);
  console.log("car-preference:", carPreference.value);
  console.log("privacy-policy:", privacyPolicy.checked);
  form.reset();
  // Functions sending form data to db, and additional logic (fx validation) go here
});
```

data logging for testing purposes

# JavaScript events

## Example of input events

Triggered immediately after the element's value changes.

```
const email = document.getElementById("email");

email.addEventListener("input", (event) => {
  console.log("email:", event.target.value);
});
```

# JavaScript events

## Example of form + keyboard events (real-time form data)

```
const email = document.getElementById("email");
const carPreference = document.getElementById("car-preference");
const privacyPolicy = document.getElementById("privacy-policy");
const formData = {};

email.addEventListener("input", (event) => {
  formData.userEmail = event.target.value;
});

carPreference.addEventListener("input", (event) => {
  formData.userCarPreference = event.target.value;
});

privacyPolicy.addEventListener("input", (event) => {
  formData.userPrivacyPolicy = event.target.checked;
});

const form = document.getElementById("newsletter-form");
form.addEventListener("submit", (event) => {
  event.preventDefault();
  console.log("email:", formData);
  // Functions sending form data to db, and additional logic (fx validation) go here
});
```

adding a new key and value to the object

# Exercise

*10 minutes*

Create an accordion that expands on click:

1. Build the HTML.
2. Add the functionality with JS:
    a. The icon should be rotated on click.
    b. Toggling the accordion should show/hide the text underneath.

▼ Read more

↓

▲ Read more

When an event happens, the browser creates an event object, puts details into it and passes it as an argument to the handler.

# Basic operators

Basic operators are things like addition +, multiplication *, subtraction -, and so on.

# Basic operators

**Math:**
Addition +, Subtraction -, Multiplication *, Division /, Remainder %, Exponentiation **, Comparison <>

**String concatenation**

**Assignment**

**Increment/decrement:**
++/--

**Other:**
And &, Or |

# Basic operators

**What are the final values of variables a, b, c and d after the code below?**

```
let a = 1, b = 1;
let c = ++a;
let d = b++;
```

# Basic operators

## Incrementation

```
const countElement = document.getElementById("count");
const incrementButton = document.getElementById("increment");

incrementButton.addEventListener("click", () => {
  let count = countElement.innerHTML;
  count++;
  countElement.innerHTML = count;
});
```

# Basic operators

## String concatenation using +

```
const orderNowButton = document.getElementById("order-now-btn");
const cardTitle = document.getElementById("card-title-one").innerHTML;

orderNowButton.addEventListener("click", () => {
  alert("I'm sorry, we don't have " + cardTitle + " at the moment");
});
```

# Basic operators

## Comparison

Greater/less than: a > b, a < b.

Greater/less than or equals: a >= b, a <= b.

Equals: a == b, please note the double equality sign == means the equality test, while a single one a = b means an assignment.

Not equals: In maths the notation is ≠, but in JavaScript it's written as a != b.

Returns a **boolean**: true or false.

# Basic operators

## Conditionals

Sometimes, we need to perform different actions based on different **conditions**.

To do that, we can use the **if statement** and the **conditional operator ?**, that's also called a "question mark" operator.

?

# Basic operators

## Conditionals

```
const countElement = document.getElementById("count");
const decrementButton = document.getElementById("decrement");

decrementButton.addEventListener("click", () => {
  let count = countElement.innerHTML;
  if (count > 0) {
    count--;
    countElement.innerHTML = count;
  }
});
```

# Optional add-ons to the assignment

Let's revisit our counter from the homework and make it and little bit more complex!

1. When the counter goes to 0, it shouldn't be possible to decrease the number anymore.
2. Add a button that resets the number to zero underneath your counter.
3. Add a top limit to your counter and show an alert when the user reaches that number. It should tell the user that they have reached the limit.

# Git & GitHub

**Follow the guide:**

https://docs.google.com/document/d/1_wV9KiVC4342DChy5gzHrb6QJ1QCBzWlXSbBG5CL5Yg/edit?usp=sharing

# Resources

- https://javascript.info/document
- https://javascript.info/events
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Expressions_and_operators
- https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String
- https://www.javascripttutorial.net/javascript-dom/