Week 5 Javascript Events

1. 4 ways of writing Events in JavaScript
2. MouseEvent in JavaScript
3. KeyboardEvent in JavaScript
4. InputEvents in JavaScript

1. 4 ways of writing Events in JavaScript

```
HTML events are "things" that happen to HTML elements.
When JavaScript is used in HTML pages, JavaScript can "react" to these
events.


HTML Events
An HTML event can be something the browser does or something a user does.

Here are some examples of HTML events:

An HTML web page has finished loading
An HTML input field was changed
An HTML button was clicked
Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes with JavaScript code to be added to
HTML elements.

1:4 ways of writing Events in JavaScript

1: using inline events alert();
2: By Calling a function (We have already seen and most common way of writing)
3: using Inline events (HTML onclick="" property and element.onclick)
4: using Event Listeners (addEventListener and IE's attachEvent)

                Open the Events HTML File
```

```
<script>
        //2nd way of writing
        const callingFunction = () => {
            alert('most common way of writing functions');
        }
        // calling events by reference
        const thirdWay = document.getElementById('thirdWay');
        thirdWay.onclick = function () {
            alert('most common way of writing functions again');
        }
//event listeners
        const fourthWay = document.querySelector('#fourthWay');
        fourthWay.addEventListener('click', () => { //callback function as im
passing function as argument
            alert('Event Handlers');
        }
        );
    </script>
```

2: MouseEvent in JavaScript

The MouseEvent Object
Events that occur when the mouse interacts with the HTML document belong to the MouseEvent Object.

3: KeyboardEvent in JavaScript

When a user presses a key on the keyboard, events belong to the KeyboardEvent Object.

https://www.w3schools.com/jsref/obj_keyboardevent.asp

5:InputEvents in JavaScript //Important -form validation

The on-change event occurs when the value of an element has been changed.

For radio buttons and checkboxes, the on-change event occurs when the the checked state has been changed.

Open InputEvents.html

```
 const iceCreams=document.getElementById('iceCreams');
 iceCreams.addEventListener('change',()=>{
```

```
 const inputChange=document.getElementById('ice').value;
    const iceCreams=document.getElementById('iceCreams').value;
    const result=document.getElementById('result');
    result.innerHTML=`Mr ${inputChange} select ${iceCreams}  ice-cream
flavour`;
});
```

6. Exercises

click event examples

log to console

```
<button id="say-hello">Log "Hello!"</button>
```

```
const button = document.querySelector("#say-hello")
// click event for a button with a specific ID
button.addEventListener("click", () => {
    console.log("Hello!")
})
```

Toggle a hidden element

```
<h1>Title</h1>
<p class="subtitle">I am the page subtitle</p>
<button id="hide-subtitle">Hide Subtitle</button>
```

```
const button = document.querySelector("#hide-subtitle")
button.addEventListener("click", () => {
    const subtitleElement = document.querySelector(".subtitle")

    // toggle hidden element
    if (subtitleElement.hidden === true) {
        subtitleElement.hidden = false
    } else {
        subtitleElement.hidden = true
    }
})
```

Change a class property

```
<h1>Title</h1>
<p class="subtitle">I am the page subtitle</p>
<button id="hide-subtitle">Hide Subtitle</button>
```

```
.hidden {
  visibility: hidden;
}
```

```
const button = document.querySelector("#hide-subtitle")
button.addEventListener("click", () => {
    const subtitleElement = document.querySelector(".subtitle")
    const elementClasses = subtitleElement.classList
```

```
    elementClasses.toggle("hidden")
})
```

Using the event target

The event callback takes a parameter of "event". This event gives us extra info about the event, including the element that triggered it (e.g. a specific button).

Change a style property

```
<div class="box"></div>
```

```
.box {
  height: 300px;
  width: 300px;
  border: solid 2px #969696;
  margin: 20px auto;
}
```

```
const box = document.querySelector(".box")
box.addEventListener("click", (event) => {
  const box = event.target
  box.styles.backgroundColor = '#990000'
})
```

Applying listeners to many elements

```
<div class="box"></div>
<div class="box"></div>
<div class="box"></div>
```

```
.box {
  height: 300px;
  width: 300px;
  border: solid 2px #969696;
  margin: 20px auto;
}
```

```
const boxes = document.querySelectorAll(".box")
boxes.forEach(box => {
  box.addEventListener("click", (event) => {
    const box = event.target
    box.styles.backgroundColor = '#990000'
  })
})
```

In-Home Exercise: Exercise 1: Validation

Create a form that has these fields, with these validation requirements:

- a text input
    - required field
    - maximum length of 20 characters
- number input
    - that has a range of -10 to 10
    - is not required
- another text input that doesn't let you type the letter "T"
- a checkbox that must be selected
- checkbox 2
- a text input that must have a value if checkbox 2 is selected

Starting Code:

```
<form>
    <!-- required, maxLength 20 -->
    <input id="maxLength" class="required" type="text">
    <!-- value between -10 to 10 -->
    <input id="range" type="number">
    <!-- does not let you type the letter T -->
    <input id="no-t" type="text">
    <!--  required -->
    <input class="required" type="checkbox">
    <input id="checkbox2" type="checkbox">
    <!-- required if checkbox 2 is required -->
    <input type="text">
    <input type="submit">
</form>


<script>

const maxLengthElement = document.querySelector("#maxLength")
maxLengthElement.addEventListener("keydown", (event) => {
    if (maxLengthElement.value.length === 20) {
        preventDefault()
    }
})
</script>
```

Only submit if all the conditions are met. All the validation should be in JS. The user should see an error message if the conditions are not met.