MODULE TWO - THE DOM & BROWSER EVENTS

BROWSER EVENTS

TODAY'S SCHEDULE

1. Cool Things
2. Concept Review/Review DOM Activities
3. What Are Browser Events?
4. Working With Browser Events
5. Other Event Concepts
6. Recap/ Course Project Phase One, Lab Five & Next Week

LEARNING OBJECTIVES

1. Review & understand the DOM, including selecting DOM elements, DOM traversal and DOM manipulation

2. Understand events & explore events that happen in the web browser

3. Explore, understand and apply knowledge of browser events, event handlers, event objects and propagation.


COOL THINGS

RESOURCES, LINKS TUTORIALS AND OTHER COOL THINGS...

- https://www.kord.app/
- https://wtg.nezia.xyz/
- https://codepen.io/gniziemazity/pen/vYyJGdr
- https://www.java5cript.com/
- https://www.reddit.com/r/coolguides/comments/lpktd3/the_etymology_of_general_computing_terms/
- https://www.youtube.com/watch?v=z6X5oEIg6Ak&t=330s

CONCEPT REVIEW

GETTING TO KNOW THE DOM

- the DOM stands for Document Object Model
- specifies how browser should create a model fo the HTML page & how JS can access and update the contents fo the page
- **not part of HTML or JS**
- called an object model because the DOM tree is made up of objects

GETTING TO KNOW THE DOM

- each object represents a different part of the page loaded in the browser window
- sometimes called an API because the DOM allows JS to interface with the HTML page
- The DOM also defines methods and properties to access and update each object in this model

LET'S REVIEW – ALL ABOUT THE DOM

WEEK FIVE - ALL ABOUT THE DOM

Your mission, should you choose to accept it, is to complete the following learning challenges. Each challenge has a special code that provides access to the instructions. Enter the access code to recieve your directions to complete your task and receive an access code for the next activity. Once you have completed the task, click on the task name to check it off as complete.

ACTIVITY ONE: PICTURE THE DOM

ACTIVITY TWO: NODE OR NOT?

ACTIVITY THREE: SELECT THAT ELEMENT

ACTIVITY FOUR: WALKING THE DOM

ACTIVITY FIVE: MASTER OF (DOM) MANIPULATION

ENTER YOUR ACCESS CODE:

submit

HTTPS://JESSICAGILFILLAN.GITHUB.IO/COMP1073-WINTER2021/WEEK-5/

WHAT IS A BROWSER EVENT?

LET'S TALK ABOUT EVENTS...

Events are basically things that happen in the system that you are programming. When the event happens, the system creates a signal and provides a mechanism that allows some kind of action (or reaction) to happen when the event occurs.

There are lots of web events that happen in browser. With a colleague, **try to think of at least 10 events that happen inside a web browser**.

LET'S TALK ABOUT EVENTS...

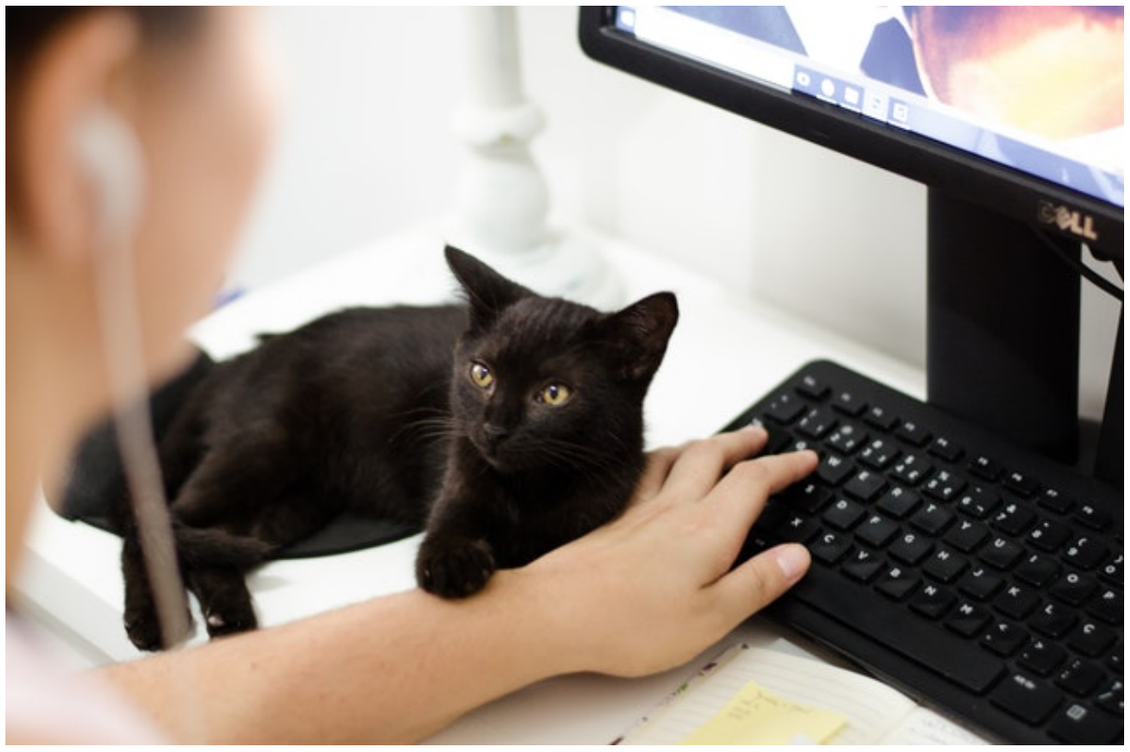
Mouse events

Event Name	Fired When
`auxclick`	A pointing device button (ANY non-primary button) has been pressed and released on an element.
`click`	A pointing device button (ANY button; soon to be primary button only) has been pressed and released on an element.
`contextmenu`	The right button of the mouse is clicked (before the context menu is displayed).
`dblclick`	A pointing device button is clicked twice on an element.
`mousedown`	A pointing device button is pressed on an element.
`mouseenter`	A pointing device is moved onto the element that has the listener attached.
`mouseleave`	A pointing device is moved off the element that has the listener attached.
`mousemove`	A pointing device is moved over an element (fired continuously as the mouse moves).
`mouseover`	A pointing device is moved onto the element that has the listener attached or onto one of its children.
`mouseout`	A pointing device is moved off the element that has the listener attached or off one of its children.

LET'S EXPLORE SOME BROWSER EVENTS...

HOW CAN WE WORK WITH THESE EVENTS?

HOW TO WORK WITH BROWSER EVENTS



1. Inline Event Handlers (Event Attributes)
2. Event Handler Property
3. AddEventListener() Functions

WHICH ONE TO USE?

- **DO NOT** use inline event handlers
- addEventListener() is **more powerful** and allows you to assign multiple event handlers to the same event; **better for larger, more complex programs**
- addEventListener() is newer, so **support may be an issue for older browsers**
- **good rule of thumb - start with event handler properties and experiment with addEventListener( )**

IMPORTANT EVENT CONCEPTS

- **Event handler** : code that runs when an event happens (usually a function defined by the developer)
- **Registering an event handler** : the term used when we define a block of code that will run when the event happens
- **Event Listener**: sometimes used interchangeably with event handler (although these two work together). An event listener listens for an event to occur

CODE EXAMPLE ONE

MODULE 2 > WEEK 6 > CODE EXAMPLES

OTHER EVENT CONCEPTS

OTHER EVENT CONCEPTS

1. Event Objects

- All event objects in the DOM are based on the Event Object.
- all other event objects (like MouseEvent and KeyboardEvent) has access to the Event Object's properties and methods.
- Automatically passed to event handlers to provide extra features and info
- target property of the event object is always a reference to the element that the event has occurred on

2. PreventDefault( )

- sometimes we need to prevent the default behaviour from happening on an object

OTHER EVENT CONCEPTS

3. Propagation: Event Bubbling & Capture

- two mechanisms that describe what happens when two handlers of the same event type are activated on one element

- We can either use these mechanisms or we can stop this from happening with stopPropagation( ).

CODE EXAMPLE TWO

MODULE 2 > WEEK 6 > CODE EXAMPLES

OTHER EVENT CONCEPTS

WORKING WITH BROWSER EVENTS – SPOOKY STORY CHALLENGE

RECAP TIME, ASSIGN TWO,

LET'S TALK MIDTERM

IN MODULE TWO, WE LEARNED

- the DOM (Document Object Model) is a tree like model created to represent the page when the it is loaded in the browser
- the DOM contains different types of nodes and these nodes are objects with methods and properties available
- although the DOM is not part of JavaScript or HTML, we can use JavaScript to interface with our HTML in order to add interactivity

IN MODULE TWO, WE LEARNED

- using DOM queries, we can select different elements on the page
- we can traverse the DOM using available properties and also manipulate the DOM using methods
- events are things that happen in the browser (a click, hover, page load etc.)
- we can use event handler properties or addEventListener in order to work with these events

IN MODULE TWO, WE LEARNED

- All event objects in the DOM are based on the Event Object and have access to Event Object's methods and properties

- e.target returns the element that triggered the event

- sometimes we may need to prevent a default behaviour (i.e. the default behaviour of a form)

- when we want to prevent the default behaviour, we can use preventDefault

COMP1073 - WINTER 2021

MODULE TWO ASSIGNMENT

NEXT WEEK : MIDTERM!

MIDTERM EXAM

- Worth 20% of final grade
- Covers everything from Week 1 - Week 6 (JS syntax, JS basics, objects, working with the DOM, browser events)
- Open Book
- Synchronous (Section 02 - Tuesday, March 2nd, Section 03 - Thursday March 4th) from 12:00 - 2:00pm

  **** if you are unable to attend, please let me know**

HOW TO PREPARE

- Review your code examples and add comments explaining the code
- Review lecture videos and slides
- Practice your JS by creating/coding something!
- Review Course Resource : MDN Web Docs - JavaScript

SAMPLE QUESTIONS

```
//Choose the code snippet that is an object literal

/*a.)*/  const vehicle = {
    type: truck,
    year: 2012,
    color: black,
    engineType: V8,
    vehicleDescription: function() {
        console.log('This type of vehicle is a ' + this.color + ' '  + this.year + ' ' +
        this.type + ' with a ' + this.type + ' engine.');
    }
};

/*b.)*/ let vehicle = ['truck', '2012', 'black', 'V8'];

/*c.)*/function Vehicle(type, year, color, engineType) {
    this.type = type;
    this.year = year;
    this.color = color;
    this.engineType = () => {
        console.log('This type of vehicle is a ' + this.color + ' '  + this.year + ' ' +
        this.type + ' with a ' + this.type + ' engine.');
    };
}
```

SAMPLE QUESTIONS

```
// True or False – Most things in JavaScript are objects and even things
that are not objects may act like an object.
```

```
// True or False – .querySelector() will return a list of nodes which is referred to as a
nodeList
```

```
<!--True or False – The following code is the best way to add JavaScript-->
<button class="btn" onclick = "myFunction()"> Click Me! </button>
```

```
//True or False – JavaScript allows us to add interactivity to our static
webpages and applications.
```

SAMPLE QUESTIONS

```
/*There are two types of errors you will run into when writing JavaScript
code

a.) logic errors and syntax errors
b.) logic errors and silly errors
c.) logic errors and runtime errors
d.) none of the above

*/
/* The JavaScript ________ is an incredibly useful tool for debugging
JavaScript that isn't working as expected
a.) console
b.) error logs
c.) debugger

*/
```

SAMPLE QUESTIONS

```
const vehicle = {
    type: truck,
    year: 2012,
    color: black,
    engineType: V8,
    vehicleDescription: function() {
        console.log('This type of vehicle is a ' + this.color + ' '  + this.year + ' ' + this.type + ' with a ' + this.type + ' engine.');
    }
};

/*Choose the correct way to access the vehicleDescription method */

//a
vehicle.vehicleDescription();
//b
vehicle['vehicleDescription()'];
//c
vehicle => vehicleDescription;
```

SAMPLE QUESTIONS

```
let studentArray = ['Harpreet', 'Jordan', 'Tolu', 'Dae'];
/* choose the correct code to add a new student name to the beginning of the array
*/

//a
studentArray.shift('Megha');

//b
studentArray.unshift('Megha');

//c
studentArray.reverse();

//d
studentArray.push('Megha');
```

SAMPLE QUESTIONS

```
<body>
    <section>
        <h1>Cats Are Great</h1>
        <p>Cats are fluffly and they like to purr</p>
        <p class="meow">Meow!</p>
        <button> Click Me! </button>
    </section>
</body>
</html>

<!-- Provided with the following HTML, choose the correct code to turn the background color ofthe button green when
the user clicks it (choose all applicable answers) -->
<script>
//a
let button = document.querySelector('button');
button.onclick = () => {
    button.style.backgroundColor = 'green';
}
//b
let button = document.querySelectorAll('button')[0];
button.addEventListener('click', function() {
    button.style.backgroundColor = 'green';
});
//c
let button = document.getElementsByTagName('button');
button.click = function() {
    button.style = 'green';
}
</script>
```

EXAM PRACTICE

Game Pin : 07341443

THANKS TO (SOURCES):

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Events

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

https://codepen.io/prosetech/pen/oRxMmZ