JavaScript

What you'll learn

- Understand the fundamental concepts in JavaScript
- Learn problem-solving skills
- Learn and apply the best practices
- Avoid common pitfalls and mistakes other JavaScript developers make
- Write solid JavaScript code

What is Javascript?

WHAT CAN YOU DO WITH IT?

WHERE DOES JAVASCRIPT CODE RUNS?

BASICS

Any Questions

The Chronicles of JavaScript Objects

```
l.js ×
(scope) => `<div class="t
p(tag => `
g.classes = (tag.classes |
me.matches('js') ? 'tag-b

.link}" class="${tag.class
v>`;



cope) => `<article>

cope.link}">${scope.title

html.js`)(scope))




e) => `<article>

e.link}">${
```

JavaScript Objects

- Objects in JavaScript, just as in several distinct programming languages, can be linked to objects in real life.
- The JavaScript objects has a property (value) and a method (behavior).
- Unlike C++ or Java, we do not create objects through classes. Instead, we directly create objects.
- Objects in JavaScript are template-based rather than class-based.
- The reason why JavaScript objects are so important is that JavaScript Design depends on a simple object-based paradigm. Meaning everything in JavaScript is an object. Yes! Even the variable with a string value.

Creating JavaScript Objects…

- You can create an object then add properties and methods or you can create an object with properties and methods.
- There are three ways you can define an object:

  - Using Literal Notation

  - Creating an instance of an Object

  - Using Constructor Notation
- Lets create objects with the given example below:

Key	Value
color	string
size	number

JavaScript Objects: Literal Notation

- **This is the simplest way to create an object. But is only recommended when creating small programs.**

```
<html>
    <body>
        <script>
            var myJeans = {color:"Black", size:28}
            //object with key: value pairs
                document.write(myJeans.color +" "+ myJeans.size);
        </script>
    </body>
</html>
```

JavaScript Objects: Instance of an Object

- We use the “new” and Object() constructor function to create an empty object.

```
<body>
    <script>
        var myJeans = new Object();
        myJeans.color = “Black”;
        myJeans.size = 28;
        document.write(myJeans.color +" "+ myJeans.size);
    </script>
</body>
```

JavaScript Objects: Constructor Notation…

- **This method of creating objects also uses the “new” keyword to create an empty object.**
- **The name of the constructor generally begins with a capital letter, unlike other functions that begin with a lowercase letter.**
- **The difference between this technique and the previous one is that it uses “this” keyword for assigning values.**

  We use it instead of the object name to indicate that the property or method belongs to the object that this function creates.

JavaScript Objects: Constructor Notation

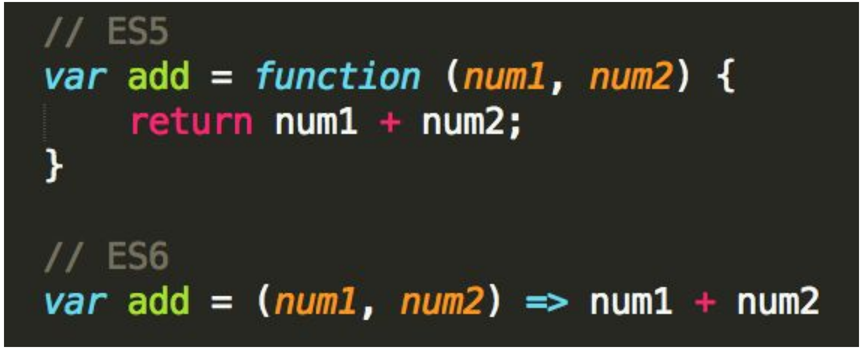
```
<body>
    <script>
        function Jeans(color, size){
            this.color = color;
            this.size = size;
        }
        var myJeans = new Jeans(“Black”, 28);
        document.write(myJeans.color +" "+ myJeans.size);
    </script>
</body>
```

Any Questions

this

JavaScript “this” keyword

- The JavaScript this keyword refers to the object it belongs to.
- Unlike other programming languages, the this keyword behaves differently in JavaScript.
- It has different values depending on where it is used:
  - In a method, this refers to the owner object.
  - Alone, this refers to the global object.
  - In a function, this refers to the global object.
  - In a function, in strict mode, this is undefined.
  - In an event, this refers to the element that received the event.
  - Methods like call(), and apply() can refer this to any object.



```
// ES5
var add = function (num1, num2) {
    return num1 + num2;
}

// ES6
var add = (num1, num2) => num1 + num2
```

Arrow Functions

- **Arrow functions were introduced in ES6.**
- **We use arrow function to write shorter function syntax.**
- **The Arrow function does not bind the “this” keyword.**
- **A regular function tends to bind “this” keyword to its preceding object.**

```
hello = function(){
      return “Hello World”;
}
```

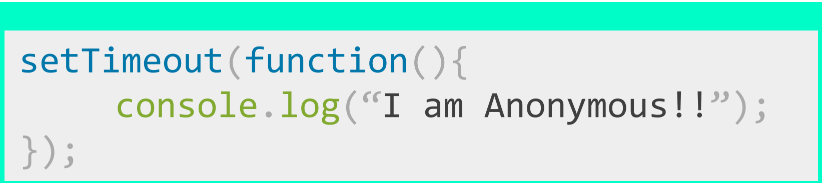
```
hello = () => “Hello World”;
```

Anonymous Functions

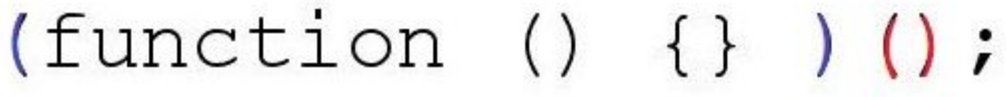
What is an anonymous function?!

Anonymous Function

- An anonymous function is a function that is declared without any named identifier to refer to it.
- An anonymous function is usually not accessible after its initial creation.
- One common use for anonymous functions is as arguments to other functions.



```
setTimeout(function(){
    console.log(“I am Anonymous!!”);
});
```



```
(function () {} )();
```

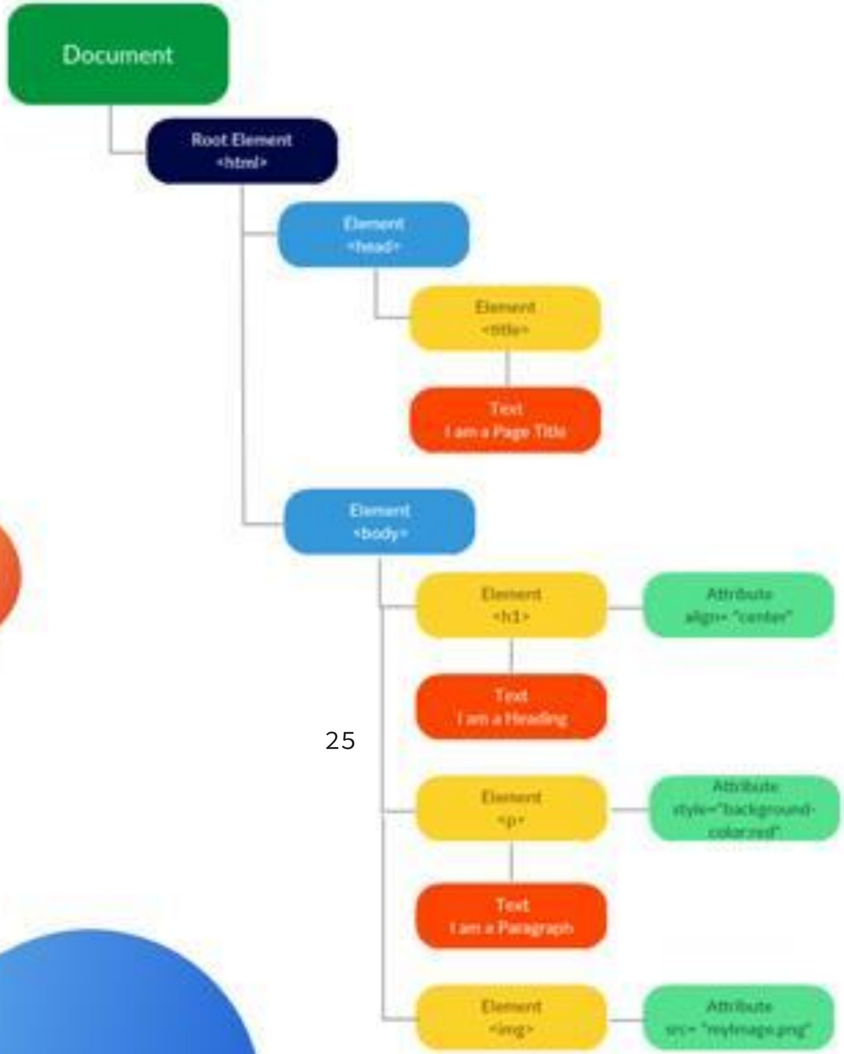
IIFE (Immediately Invoked Function Expression)

- **An IIFE (Immediately Invoked Function Expression) is a JavaScript function that runs as soon as it is defined.**
- **It is a design pattern which is also known as a Self-Executing Anonymous Function.**
- **We simply want to call a function in order to get an output and never want to use it again and don't want our program to ever be able to accidentally access it.**

```
(function(){
     console.log(“I am Anonymous!!”);
})();
```

Any Questions

HTML DOM



Document

Root Element
<html>

Element
<head>

Element
<title>

Text
I am a Page Title

Element
<body>

Element
<h1>

Attribute
align= "center"

Text
I am a Heading

Element
<p>

Attribute
style="background-color:red"

Text
I am a Paragraph

Element
<img>

Attribute
src= "myImage.png"

Understanding DOM…

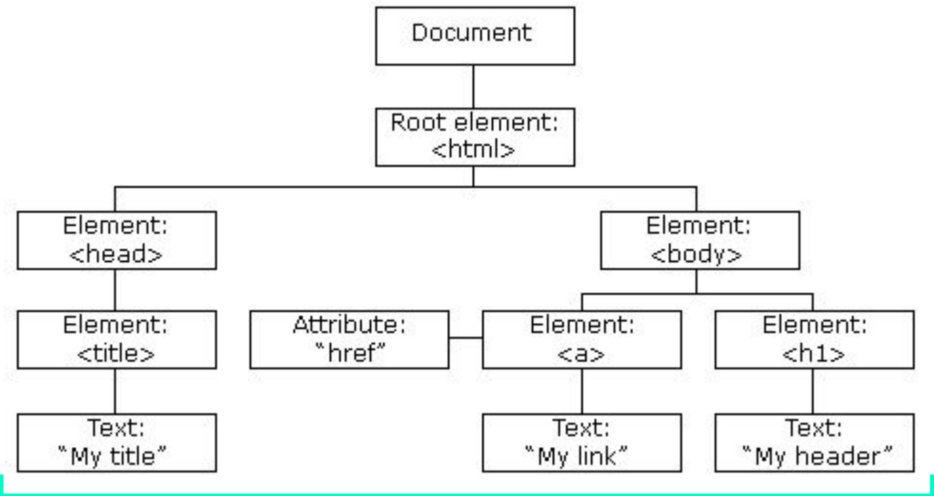
- **DOM stands for Document Object Model.**
- **It is the data representation of the objects that comprise the structure and content of a document on the web.**
- **It is a programming interface for valid HTML and well structured XML documents.**
- **It defines the logical structure of documents and the way a document is accessed and manipulated.**
- **DOM is a way to represent the webpage in the structured hierarchical way.**

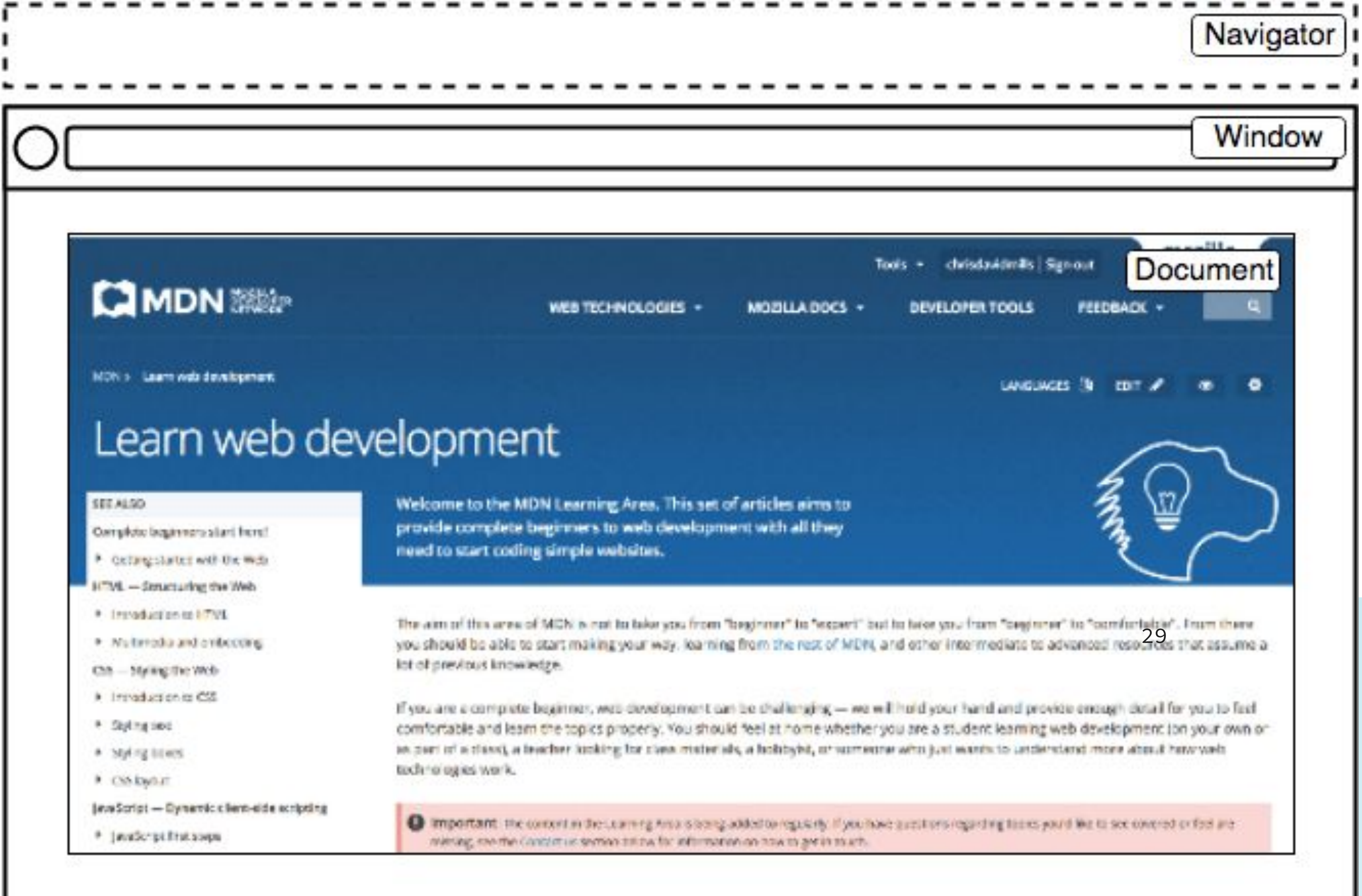
Understanding DOM...

- **DOM can be thought of as a Tree of different html elements, its properties, methods and events.**
- **It is called an “Object Model”, because the document is modeled using objects. The model not only includes the structure but also the behavior of a document.**
- **JavaScript uses this Object Model to interact, manipulate and communicate with the structure of an html page.**

Understanding DOM

```
<!DOCTYPE html>
<html>
     <head>
          <title>My Title</title>
     </head>
     <body>
          <a href=“#”>My Link</a>
          <h1>My Header</h1>
     </body>
</html>
```





Learn web development

SEE ALSO

Complete beginners start here!

- Getting started with the Web

HTML — Structuring the Web

- Introduction to HTML
- Multimedia and embedding

CSS — Styling the Web

- Introduction to CSS
- Styling text
- Styling boxes
- CSS layout

JavaScript — Dynamic client-side scripting

- JavaScript first steps

Welcome to the MDN Learning Area. This set of articles aims to provide complete beginners to web development with all they need to start coding simple websites.

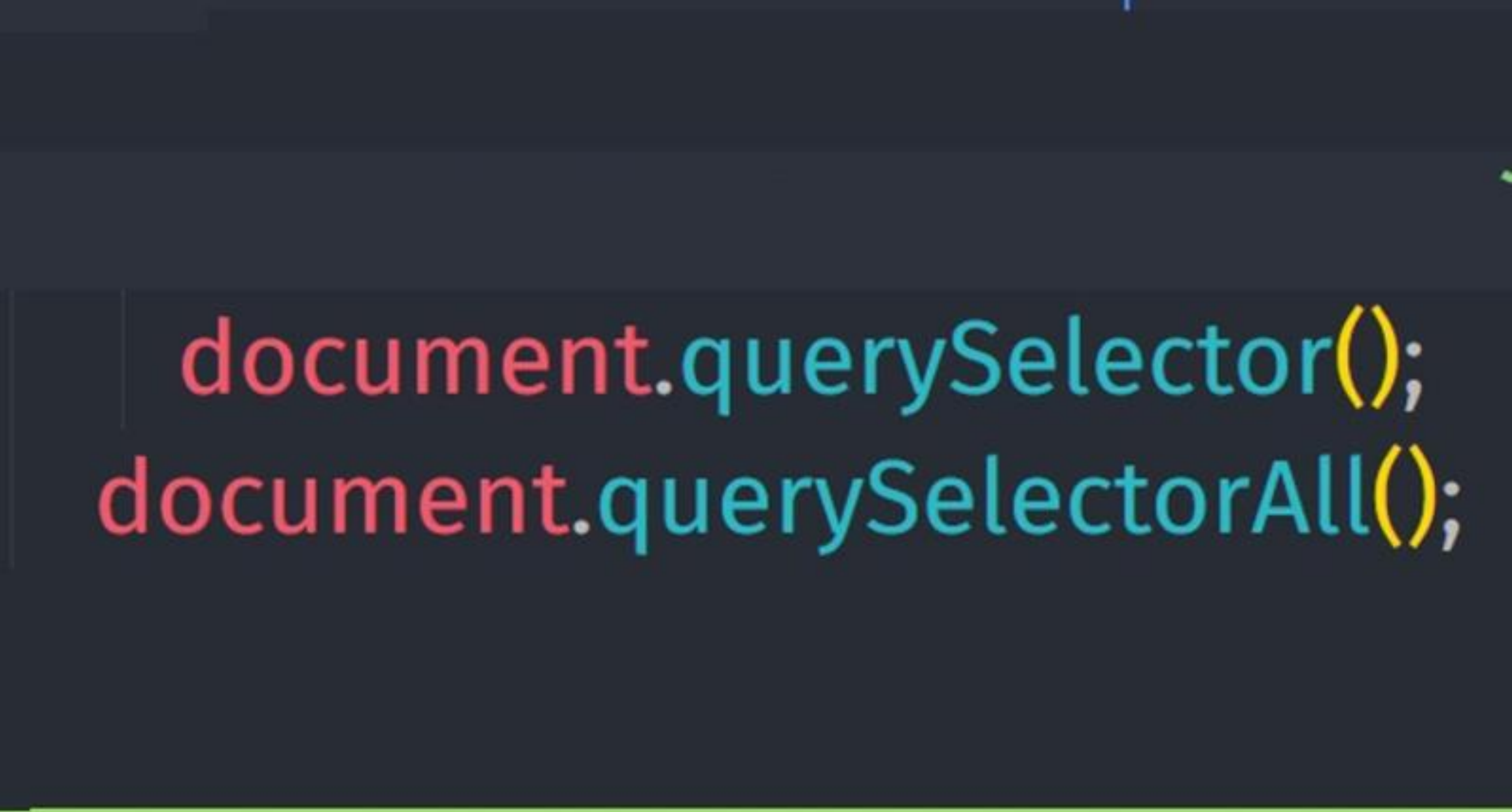
The aim of this area of MDN is not to take you from "beginner" to "expert" but to take you from "beginner" to "comfortable". From there you should be able to start making your way, learning from the rest of MDN, and other intermediate to advanced resources that assume a lot of previous knowledge.

If you are a complete beginner, web development can be challenging — we will hold your hand and provide enough detail for you to feel comfortable and learn the topics properly. You should feel at home whether you are a student learning web development (on your own or as part of a class), a teacher looking for class materials, a hobbyist, or someone who just wants to understand more about how web technologies work.

Important: The content in the Learning Area is being added to regularly. If you have questions regarding topics you'd like to see covered or feel are missing, see the Contact us section below for information on how to get in touch.

Any Questions

be back in 10 minutes…



```
    document.querySelector();
document.querySelectorAll();
```

JavaScript Selectors…

- JavaScript uses the CSS syntax to select and manipulate HTML elements in a DOM.
- Selectors are used to "find" (select) HTML elements based on their tag name, id, classes, types, attributes, values of attributes and much more.
- In JavaScript, we can select elements in multiple ways as follows:
  - getElementById(), getElementsByClassName(), getElementsByTagName(), querySelector(), querySelectorAll()

  We also need to tell JavaScript from where to select these elements. In most cases it is always the “document” object.

JavaScript Selectors...

- **getElementById() and querySelector() returns (selects) a single DOM.**
- **Where as other selectors either returns an HTML collection or an Array of Node Lists, depending upon which selector type you use.**
- **Depending upon which value the selectors return, we can manipulate the HTML DOM.**

querySelector() and querySelectorAll() are the most preferred JS selector to use while programming.

#28
SELECT
ELEMENTS
USING JS

Any Questions

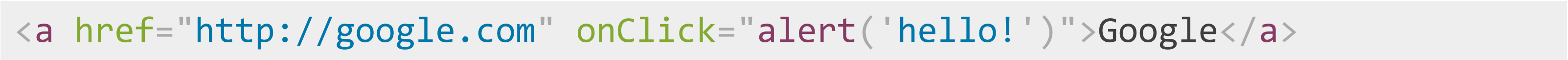
EVENTS

Understanding Events in JavaScript...

- HTML events are things that happens to an HTML element.
- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.
- When the page loads, it is called an event. When the user clicks a button, that click too is an event.
- We can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Exploring JavaScript Events with Event Handlers...

- Events in JavaScript are handled by an Event Handlers.
- They are JavaScript code that are not added inside the <script> tags, but rather, inside the html tags, that execute JavaScript when something happens, such as pressing a button, moving your mouse over a link, submitting a form etc.



```
<a href="http://google.com" onClick="alert('hello!')">Google</a>
```

- Here, onClick is the JavaScript event handler.

While this is a good example of getting to know how events work, JavaScript also has a method called "Event Listener". Lets Explore.

Exploring JavaScript Events with Event Handlers…

- An event listener is a method that attaches an event handler to a specific element.
- This way we do not always have to rely on writing our codes in HTML itself.
- The best way to work around with events is using an event listener method.

To use an event listener, we must first select an element on which to attach the event.

Exploring JavaScript Events with Event Handlers

- Now that we know how to select an element, lets look at the following example:

```
<a href=“#” id=“myBtn”>Show
Alert</a>
```

```
document.getElementById("myBtn").addEventListener("click", () => alert(“hello!”));
```

- We created an <a> element and assigned an ID to it. We then selected that element using the getElementById() method in JavaScript. Finally we attached an event handler (click) to alert “hello!”.

Now, whenever a user clicks the button, it will show an alert with the message “hello!”.

JavaScript Events: Most used event handlers

- **There are numerous event handlers for you to explore. Below are listed some of the most used handlers for you too look into:**
  - onClick
  - ondblclick
  - onkeypress
  - onkeyup
  - onmousein
  - onmouseout
  - onsubmit

Any Questions

JS form events

Form Events...

Object	Event Handler
Button	onClick, onBlur, onFocus
Checkbox	onClick, onBlur, onFocus
File Upload	onClick, onBlur, onFocus
password	onClick, onBlur, onSelect
Radio	onClick, onBlur, onFocus
Reset	onReset
Select	onChange, onBlur, onFocus
Submit	onSubmit
Text	onClick, onBlur, onFocus, onChange
Textarea	onClick, onBlur, onFocus, onChange

Form Events…

Handler	Description
onblur	The JavaScript onblur property runs when the object loses focus.
onchange	The JavaScript onchange property works when the property of an element changes.
oncontextmenu	The oncontextmenu property of the JavaScript works when the user is right-clicked to open the context menu.
onfocus	The Onfocus feature works when it focuses on an object.
oninput	The JavaScript oninput function works when an element is entered or changed.
onreset	The JavaScript onreset property is used to reset a form.
onsearch	The JavaScript onsearch property runs the codes when the user presses the “enter” key.
onselect	The JavaScript onselect property runs after you select the text in an element.

Form Events: preventDefault() Method

- In some cases we need to intercept a form submit. Most of the time we do this for form input validation.
- JavaScript gives us a method named preventDefault() to intercept any default behavior of an element event.
- We can use this method when we submit the form to prevent its default behavior.

```
const form =
document.querySelector('form')
form.addEventListener('submit', (event) =>
{
  // submit event detected
  event.preventDefault()
})
```

Any Questions

THANK
YOU