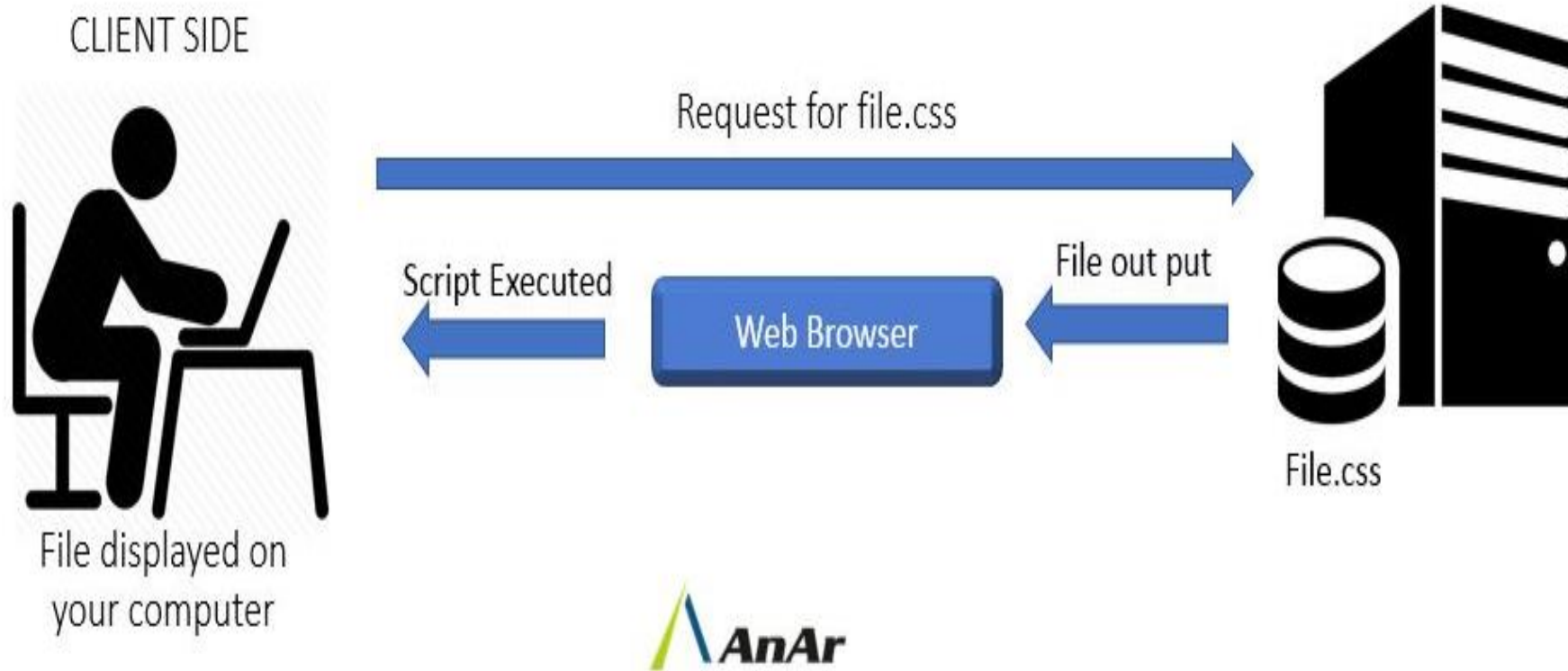


Handling HTML Events



<https://anarsolutions.com/client-side-scripting/>

Overview

- In this module we are going to cover about how to Handle HTML Events, Animations through JS. Reading element state & data using JS. Various form handling components and validations. And in the last section we will discuss about Handling Cookies and Session Data with JS.

Agenda

- Handling HTML Events
- Animations
- Reading element state & data
- form handling and validations
- Handling Cookies and Session Data

Learning Objectives

- Javascript basics and Handling HTML events
- Design webpage using Javascript functions
- Handle session and cookies using JS

Handling HTML Events

- JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

Handling HTML Events

```
<script>
  function showAlert() {
    alert('Clicked!');
  }
</script>
<input type="button" value="Save" onclick="showAlert()">
```

<https://cope-ali.github.io/cope-ali261.github.io/img/HTMLEvents.png>

Animations

JavaScript animations are done by programming gradual changes in an element's style. The changes are called by a **timer**.

```
function myMove() {  
  let id = null;  
  const elem = document.getElementById("animate");  
  let pos = 0;  
  clearInterval(id);  
  id = setInterval(frame, 5);  
  function frame() {  
    if (pos == 350) {  
      clearInterval(id);  
    } else {  
      pos++;  
      elem.style.top = pos + 'px';  
      elem.style.left = pos + 'px';  
    }  
  }  
}
```

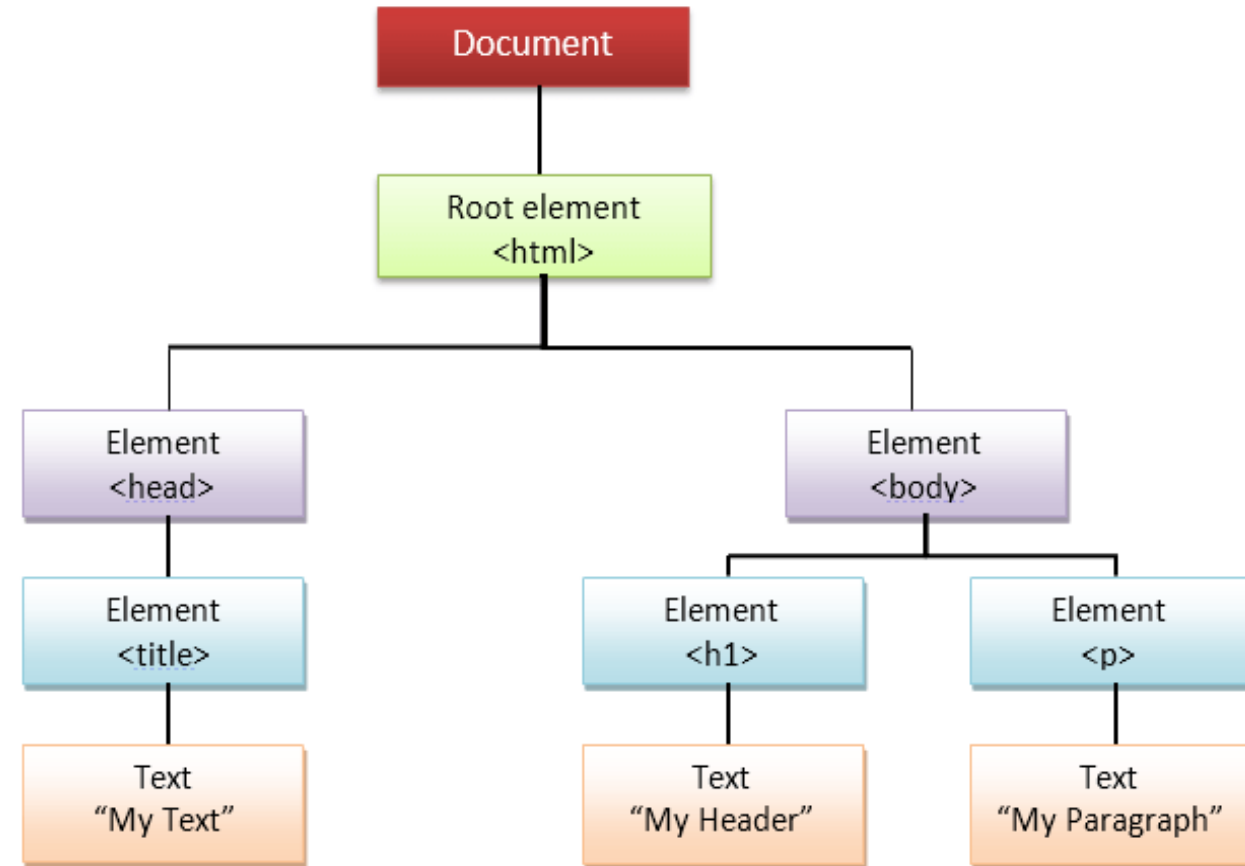
Click Me



https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_animate_3

What is DOM in JavaScript?

JavaScript can access all the elements in a webpage making use of **Document Object Model (DOM)**.



https://www.guru99.com/images/JavaScript/javascript8_1.png

Finding HTML Elements

JavaScript, you want to manipulate HTML elements

- Finding HTML Element by Id
- Finding HTML Elements by Tag Name
- Finding HTML Elements by Class Name
- Finding HTML Elements by CSS Selectors

Finding HTML Elements

```
<p id="mark">This is a paragraph of text.</p>  
<p>This is another paragraph of text.</p>
```

```
<script>  
// Selecting element with id mark  
var match = document.getElementById("mark");  
  
// Highlighting element's background  
match.style.background = "yellow";  
</script>
```

This is a paragraph of text.

This is another paragraph of text.

Form handling and validations

Forms are used in webpages for the user to enter their required details that further send it to the server for processing. A form is also known as a web form or HTML form.

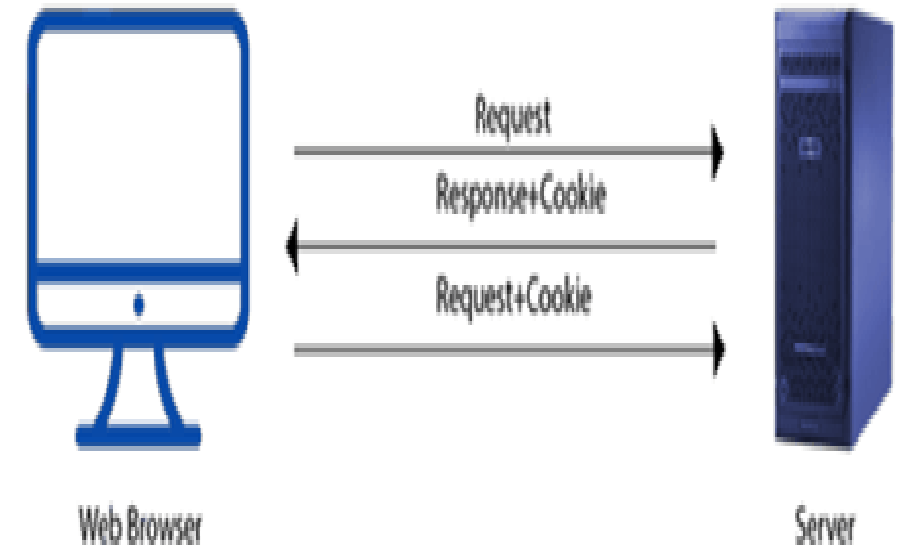
Test JavaScript Form Validation

Name*	<input type="text"/>	Please enter your name!
Address	<input type="text"/>	
Zip Code*	<input type="text"/>	
Country*	<input type="text" value="Please select..."/>	
Gender*	<input type="radio"/> Male <input type="radio"/> Female	
Preferences*	<input type="checkbox"/> Red <input type="checkbox"/> Green <input type="checkbox"/> Blue	
Phone*	<input type="text"/>	
Email*	<input type="text"/>	
password (6-8 characters)*	<input type="text"/>	
Verify password*	<input type="text"/>	
	<input type="button" value="SEND"/> <input type="button" value="CLEAR"/>	

https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/images/JS_FormValidation.png

Cookies in JavaScript

- A cookie is a piece of data that is stored on your computer to be accessed by your browser.
- Cookies are saved as **key/value pairs**.



<https://www.javatpoint.com/javascript-cookies>

Cookies in JavaScript

Cookies consist of 5 variable-length fields:

- **Expires** – This shows the date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- **Domain** – The domain field provides the domain name of your site.
- **Path** – It is the path to the directory or web page that set the cookie.
- **Secure** – cookie may only be retrieved with a secure server.
- **Name=Value** – This depicts the cookies that are set and retrieved in the form of key-value pairs.

Cookies in JavaScript

```
<input type="button" value="setCookie" onclick="setCookie()">
<input type="button" value="getCookie" onclick="getCookie()">

<script>
function setCookie()
{
    document.cookie="username=Duke Martin";
}
function getCookie()
{
    if(document.cookie.length!=0)
    {
        var array=document.cookie.split("=");
        alert("Name="+array[0]+" "+"Value="+array[1]);
    }
    else
    {
        alert("Cookie not available");
    }
}
</script>
```

<https://www.javatpoint.com/javascript-cookies>

Session Storage

The session Storage object stores data for only one session (the data is deleted when the browser tab is closed).

Syntax for SAVING data to sessionStorage:

```
sessionStorage.setItem("key", "value");
```

Syntax for READING data from sessionStorage:

```
var lastname = sessionStorage.getItem("key");
```

Syntax for REMOVING saved data from sessionStorage:

```
sessionStorage.removeItem("key");
```

Syntax for REMOVING ALL saved data from sessionStorage:

```
sessionStorage.clear();
```

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

Conclusion

We have learned and covered the following topics in this module:

- Basics of Javascript and different event handling functions
- Build web page design using different Javascript functionalities
- How to manage cookies and session using Javascript

THANK YOU