

STUDENT DEVELOPMENT PROGRAM

Java Script And Jquery

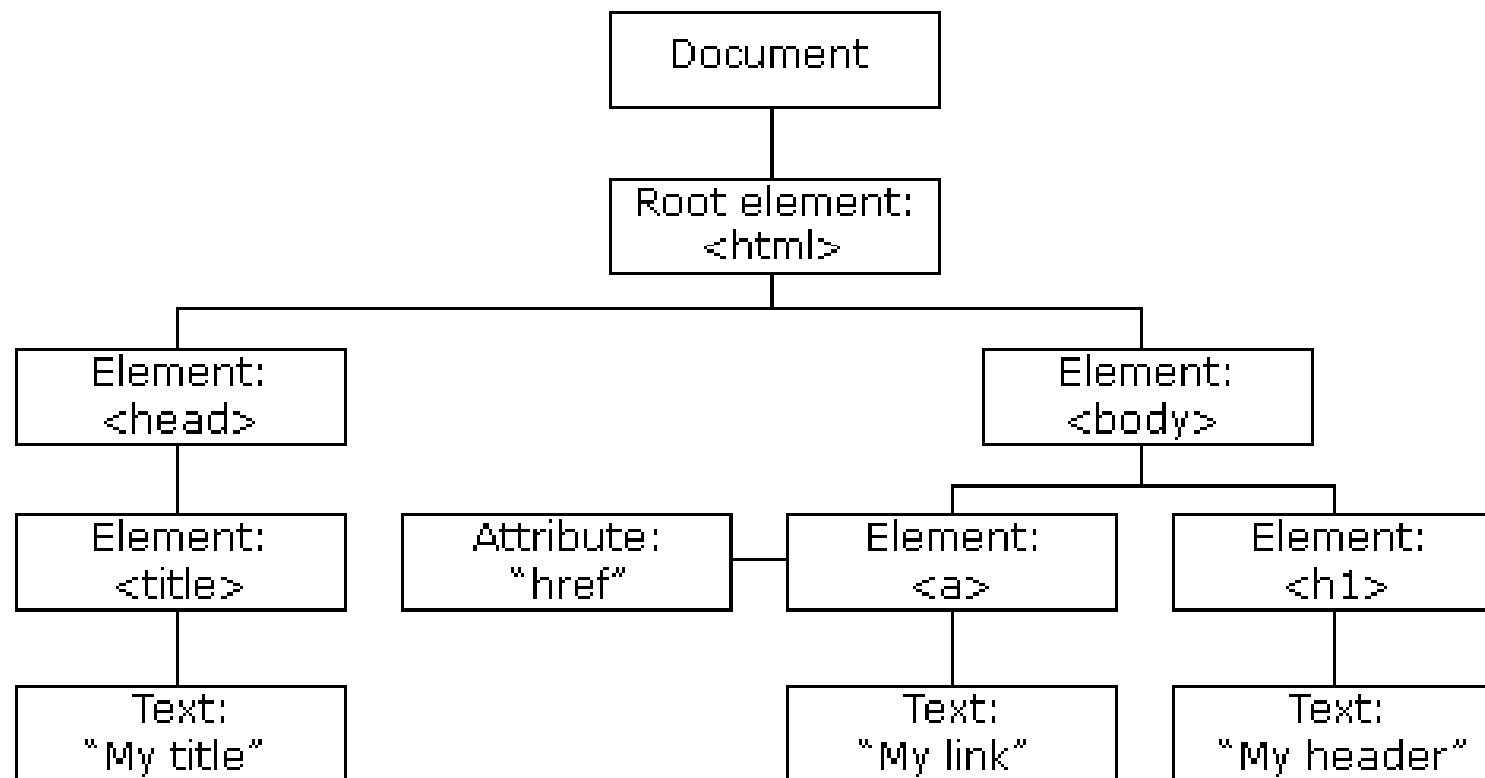
PREPARED BY: PROF. KINJAL PATEL

JAVASCRIPT

- ◉ JavaScript is the world's most popular programming language.
- ◉ JavaScript is the programming language of the Web.
- ◉ JavaScript is easy to learn.
- ◉ I will teach you JavaScript from basic to advanced.

GETTING STARTED WITH DOM

◉ The HTML DOM Tree of Objects



WHAT IS THE DOM?

- ◉ The DOM is a W3C (World Wide Web Consortium) standard.
- ◉ The DOM defines a standard for accessing documents:
- ◉ *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*
- ◉ The W3C DOM standard is separated into 3 different parts:
 - ◉ Core DOM - standard model for all document types
 - ◉ XML DOM - standard model for XML documents
 - ◉ HTML DOM - standard model for HTML documents

WHAT IS THE HTML DOM?

- ◉ The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:
- ◉ The HTML elements as **objects**
- ◉ The **properties** of all HTML elements
- ◉ The **methods** to access all HTML elements
- ◉ The **events** for all HTML elements
- ◉ In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

DOM HTML

- ◉ The HTML DOM allows JavaScript to change the content of HTML elements.
- ◉ Changing HTML Content
- ◉ The easiest way to modify the content of an HTML element is by using the innerHTML property.
- ◉ To change the content of an HTML element, use this syntax:
- ◉ `document.getElementById(id).innerHTML = new HTML`
- ◉ This example changes the content of a <p> element:

EXAMPLE:

```
<html>
  <body>

    <p id="p1">Hello World!</p>

    <script>
      document.getElementById("p1").innerHTML = "New
      text!";
    </script>

  </body>
</html>
```

EXAMPLE EXPLAINED:

- ◉ The HTML document above contains a `<p>` element with `id="p1"`
- ◉ We use the HTML DOM to get the element with `id="p1"`
- ◉ A JavaScript changes the content (`innerHTML`) of that element to "New text!"

VARIABLE DECLARATION IN JAVASCRIPT

- ◉ 4 Ways to Declare a JavaScript Variable:
- ◉ Using var
- ◉ Using let
- ◉ Using const
- ◉ Using nothing

WHEN TO USE JAVASCRIPT VAR?

- ◉ Always declare JavaScript variables with var, let, or const.
- ◉ The var keyword is used in all JavaScript code from 1995 to 2015.
- ◉ The let and const keywords were added to JavaScript in 2015.
- ◉ If you want your code to run in older browser, you must use var.
- ◉ Syntax: var identifier=value;
- ◉ Example: var Collegenname="Silveroak"
- ◉ var age=15

JAVASCRIPT LET:

- ◉ The let keyword was introduced in ES6 (2015).
- ◉ Variables defined with let cannot be Redeclared.
- ◉ Variables defined with let must be Declared before use.
- ◉ Variables defined with let have Block Scope.
- ◉ Example: let x=50

BLOCK SCOPE:

- ◉ Before ES6 (2015), JavaScript had only **Global Scope** and **Function Scope**.
- ◉ ES6 introduced two important new JavaScript keywords: `let` and `const`.
- ◉ These two keywords provide **Block Scope** in JavaScript.
- ◉ Variables declared inside a `{ }` block cannot be accessed from outside the block:

EXAMPLE:

- ◉ {
 let x = 2;
}
// x can NOT be used here
- ◉ Variables declared with the var keyword can NOT have block scope.
- ◉ Variables declared inside a { } block can be accessed from outside the block.
- ◉ Example
- ◉ {
 var x = 2;
}
// x CAN be used here

DIFFERENCE BETWEEN LET AND VAR:

Var	Let
In var, Redeclaring a variable inside the block will also declare outside the block.	In let, Redeclaring a variable inside the block will not declare outside the block.
Redeclaring a JavaScript variable with var is allowed anywhere in a program: Example: var x=50; var x=15;	With let, redeclaring a variable in the same block is NOT allowed: Example: let x=10; let x=15; is not allowed
In var, You can use the variable before it is declared: Example: carName:"BMW" Var carName;	In let, Using a let variable before it is declared will result in a Error: Example: carName:"BMW" let carName; is Not allowed
Prepared By:Prof. Kinjal Patel	

JAVASCRIPT CONST:

- ◉ The const keyword was introduced in ES6 (2015).
- ◉ Variables defined with const cannot be Redeclared.
- ◉ Variables defined with const cannot be Reassigned.
- ◉ Variables defined with const have Block Scope.
- ◉ Syntax: `const identifier=value;`
- ◉ Example: `const pi=3.14;`

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Variables</h1>

<p>In this example, price1, price2, and total are variables.</p>

<p id="demo"></p>

<script>
const price1 = 5;
const price2 = 6;
let total = price1 + price2;
document.getElementById("demo").innerHTML =
"The total is: " + total;
</script>

</body>
</html>
```


EXPLANATION:

- ◉ The two variables price1 and price2 are declared with the const keyword.
- ◉ These are constant values and cannot be changed.
- ◉ The variable total is declared with the let keyword.
- ◉ This is a value that can be changed.

JAVASCRIPT IDENTIFIERS:

- ◉ All JavaScript **variables** must be **identified** with **unique names**.
- ◉ These unique names are called **identifiers**.
- ◉ Identifiers can be short names (like x and y) or more descriptive names (age, sum, totalVolume).
- ◉ The general rules for constructing names for variables (unique identifiers) are:
 - ◉ Names can contain letters, digits, underscores, and dollar signs.
 - ◉ Names must begin with a letter
 - ◉ Names can also begin with \$ and _ (but we will not use it in this tutorial)
 - ◉ Names are case sensitive (y and Y are different variables)
 - ◉ Reserved words (like JavaScript keywords) cannot be used as names
- ◉ Note
- ◉ JavaScript identifiers are case-sensitive.

JAVASCRIPT DATA TYPES:

- ◉ JavaScript variables can hold numbers like 100 and text values like "John Doe".
- ◉ In programming, text values are called text strings.
- ◉ JavaScript can handle many types of data, but for now, just think of numbers and strings.
- ◉ Strings are written inside double or single quotes. Numbers are written without quotes.
- ◉ If you put a number in quotes, it will be treated as a text string.

```
<!DOCTYPE html>
<html>
<body>

<h1>JavaScript Variables</h1>

<p>Strings are written with quotes.</p>
<p>Numbers are written without quotes.</p>

<p id="demo"></p>

<script>
const pi = 3.14;
let person = "Johnny Lever";
let answer = 'Yes I am!';

document.getElementById("demo").innerHTML =
pi + "<br>" + person + "<br>" + answer;
</script>

</body>
</html>
```