The Domain Object Model

(<https://www.w3schools.com/js/js_htmldom_document.asp>)

There are a few helpful functions to access elements on page:

* document.getElementById(id);accesses element with given id
* document.getElementsByTagName(name);accesses all elements with given tag name
* document.getElementsByClassName(name);accesses all elements with given class name

The content or style of elements can then be modified with JavaScript:

* *element*.innerHTML = write/update HTML for selection
* *element*.attribute = update specific attribute value for selection
* *element*.style.*property* = update style for the selection

DOM style properties: <https://www.w3schools.com/jsref/dom_obj_style.asp>

Example:

In html:

<html>

<head> <script src="example.js" defer(want unfll the html content is landed first)></script> </head>

<body>

<p>Welcome to the page!</p>

<div id="pic"></div>

<p>I hope you enjoyed it</p>

</body>

</html>

In example.js:

document.getElementByld(“pic”).inner.HTML= ‘<img scr= “face.png” id= “face”>;(we are writing HTML for the page and giving itar id)

document.getElementByld(“face”).style.borderStyle= “solid”;

document.getElementByld(“face”).style.borderColor= “red”;

Functions

A function is a named block of code designed to perform a task

function greeting{

alert(“welcome”[only ever executes when called--->from a button click] );

}

Greeting( ); calls the greeting( ) function

Function syntax:

function fiName(){

statement;

statement2;

....

}

Buttons

Useful attributes:

* type: submit | reset | button

Submit: submit the form data(input bases within form tags )

Reset: clear all the form data

Button: just a button--->we can control its effect with JavaSeript

* onclick:

Want to do when the button is clicked.(often this is calling a function)

Variables

Variables associate a name with a storage location that holds data

Syntax to declare a new variable:

* old method:
* current method:
* assignment:

Example:

Output

* console.log()

Inputs (<https://www.w3schools.com/html/html_form_input_types.asp>)

Text box:

HTML:

<form>

Enter a your name:<br>

<input id="nameEntry" type="text" placeholder="example: Ali">

</form>

<button onclick="greeting()">Submit</button>

JavaScript:

function greeting() {

let userName = document.getElementById("nameEntry").value;

alert("Hello " + userName);

}

Radio buttons:

HTML:

<form onchange="getInput()" id="question1">

<input type="radio" name="answer" value="a">Agree<br>

<input type="radio" name="answer" value="b">Neutral<br>

<input type="radio" name="answer" value="c">Disagree<br>

</form>

JavaScript:

function getInput() {

let form = document.getElementById("question1");

let response = form.elements.answer.value;

if (response === "a") {

alert("I'm glad you agree");

}

}

Dropdown menus:

HTML:

<select name="dropdown" size="1">

<option value="1" selected>January</option>

<option value="2">February</option>

<option value="3">March</option>

<option value="4">April</option>

...

</select>

Conditional Statements

Conditional statements check if a specific condition is true, and perform different tasks based on the result of the condition.

Syntax of an if-statement:

More conditions

We can add 'else' and 'else if' to do more complex conditional operations

Syntax:

Comparison Operators: Example:

* Less than:
* Greater than:
* Less than or equal:
* Greater than or equal:
* Is equal to:
* Is NOT equal to:

Logical Operators:

* the *and* operator:
* the *or* operator:
* the *not* operator: