# what is JAVASCRIPT?

|  |  |
| --- | --- |
| ☐ | JavaScript is the programming language of the Web. |
| ☐ | All modern HTML pages are using JavaScript. |

# Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. **HTML** to define the content of web pages

   2. **CSS** to specify the layout of web pages

   3. **JavaScript** to program the behavior of web pages

InnerHTML: document.getElementById("demo").innerHTML = "Hello JavaScript";

HTML Styles (CSS) :document.getElementById("demo").style.fontSize = "25px";

# JavaScript Functions and Events?

A JavaScript **function** is a block of JavaScript code, that can be executed when "asked" for.

For example, a function can be executed when an **event** occurs, like when the user clicks a button.

# JavaScript Display Possibilities?

* Writing into an alert box, using **window.alert()**.
* Writing into the HTML output using **document.write()**.
* Writing into an HTML element, using **innerHTML**.
* Writing into the browser console, using **console.log()**.

The document.write() method should be used only for testing.

# JavaScript variable?

JavaScript variables are containers for storing data values.

Example:

var x = 5;  
var y = 6;  
var z = x + y;

# JavaScript Keywords?

|  |  |
| --- | --- |
| **Keyword** | **Description** |
| break | Terminates a switch or a loop |
| continue | Jumps out of a loop and starts at the top |
| debugger | Stops the execution of JavaScript, and calls (if available) the debugging function |
| do ... while | Executes a block of statements, and repeats the block, while a condition is true |
| for | Marks a block of statements to be executed, as long as a condition is true |
| function | Declares a function |
| if ... else | Marks a block of statements to be executed, depending on a condition |
| return | Exits a function |
| switch | Marks a block of statements to be executed, depending on different cases |
| try ... catch | Implements error handling to a block of statements |
| var | Declares a variable |

# JavaScript Operators?

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulus |
| ++ | Increment |
| -- | Decrement |

# JavaScript Data Types?

JavaScript variables can hold many **data types**: numbers, strings, arrays, objects and more:

var length = 16;                               // Number  
var lastName = "Johnson";                      // String  
var cars = ["Saab", "Volvo", "BMW"];           // Array  
var x = {firstName:"John", lastName:"Doe"};    // Object

# JavaScript Objects?

Real Life Objects, Properties, and Methods

In real life, a car is an **object**.

A car has **properties** like weight and color, and **methods** like start and stop:

# JavaScript Scope?

In JavaScript, scope is the set of variables, objects, and functions you have access to.

1. Local Variable
2. Global variable
3. Local variables are deleted when the function is completed.
4. Global variables are deleted when you close the page.

# JavaScript Events?

HTML events are **"things"** that happen to HTML elements.

HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

|  |  |
| --- | --- |
| **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 |

# JavaScript Number Methods?

Global Methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| Number() | Returns a number, converted from its argument. |
| parseFloat() | Parses its argument and returns a floating point number |
| parseInt() | Parses its argument and returns an integer |

Number Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| toString() | Returns a number as a string |
| toExponential() | Returns a string, with a number rounded and written using exponential notation. |
| toFixed() | Returns a string, with a number rounded and written with a specified number of decimals. |
| toPrecision() | Returns a string, with a number written with a specified length |
| valueOf() | Returns a number as a number |

# JavaScript Date Methods?

Date get methods:

|  |  |
| --- | --- |
| **Method** | **Description** |
| getDate() | Get the day as a number (1-31) |
| getDay() | Get the weekday as a number (0-6) |
| getFullYear() | Get the four digit year (yyyy) |
| getHours() | Get the hour (0-23) |
| getMilliseconds() | Get the milliseconds (0-999) |
| getMinutes() | Get the minutes (0-59) |
| getMonth() | Get the month (0-11) |
| getSeconds() | Get the seconds (0-59) |
| getTime() | Get the time (milliseconds since January 1, 1970) |

Date Set Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| setDate() | Set the day as a number (1-31) |
| setFullYear() | Set the year (optionally month and day yyyy.mm.dd) |
| setHours() | Set the hour (0-23) |
| setMilliseconds() | Set the milliseconds (0-999) |
| setMinutes() | Set the minutes (0-59) |
| setMonth() | Set the month (0-11) |
| setSeconds() | Set the seconds (0-59) |
| setTime() | Set the time (milliseconds since January 1, 1970) |

# JavaScript Arrays?

JavaScript arrays are used to store multiple values in a single variable.

var cars = ["Saab", "Volvo", "BMW"];

# JavaScript Array Methods?

1. Converting Arrays to Strings : **valueOf()**
2. join() method joins array elements into a string.
3. The pop method removes the last element from an array.
4. The push method appends a new element to an array.
5. The shift method removes (shifts) the first element of an array.
6. The unshift method adds new elements to the beginning of an array.
7. Changing ElementsArray elements are accessed using their index number.
8. Delete method Deleting elements leaves undefined holes in an array.
9. The splice() method adds new elements to an array.
10. The sort() method sorts an array alphabetically.
11. The **reverse()** method reverses the elements in an array.
12. The **concat()** method creates a new array by concatenating two arrays:
13. The **slice()** method slices out a piece of an array:

# JavaScript Array Methods?

|  |
| --- |
|  |
| abstract | arguments | boolean | break | byte |
| case | catch | char | class\* | const |
| continue | debugger | default | delete | do |
| double | else | enum\* | eval | export\* |
| extends\* | false | final | finally | float |
| for | function | goto | if | implements |
| import\* | in | instanceof | int | interface |
| let | long | native | new | null |
| package | private | protected | public | return |
| short | static | super\* | switch | synchronized |
| this | throw | throws | transient | true |
| try | typeof | var | void | volatile |
| while | with | yield |  |  |

## JavaScript Objects, Properties, and Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Array | Date | eval | function | hasOwnProperty |
| Infinity | isFinite | isNaN | isPrototypeOf | length |
| Math | NaN | name | Number | Object |
| prototype | String | toString | undefined | valueOf |

## Windows Reserved Words

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lert | all | anchor | anchors | area |
| assign | blur | button | checkbox | clearInterval |
| clearTimeout | clientInformation | close | closed | confirm |
| constructor | crypto | decodeURI | decodeURIComponent | defaultStatus |
| document | element | elements | embed | embeds |
| encodeURI | encodeURIComponent | escape | event | fileUpload |
| focus | form | forms | frame | innerHeight |
| innerWidth | layer | layers | link | location |
| mimeTypes | navigate | navigator | frames | frameRate |
| hidden | history | image | images | offscreenBuffering |
| open | opener | option | outerHeight | outerWidth |
| packages | pageXOffset | pageYOffset | parent | parseFloat |
| parseInt | password | pkcs11 | plugin | prompt |
| propertyIsEnum | radio | reset | screenX | screenY |
| scroll | secure | select | self | setInterval |
| setTimeout | status | submit | taint | text |
| textarea | top | unescape | untaint | window |

## HTML Event Handlers

|  |  |  |  |
| --- | --- | --- | --- |
| onblur | onclick | onerror | onfocus |
| onkeydown | onkeypress | onkeyup | onmouseover |
| onload | onmouseup | onmousedown | onsubmit |

# JavaScript CLOSURES?

Closures are nothing but local and global variables.

Local variables have scope of particular function and it’s deleted when the function completed.

Global variables have scope of globally and it’s deleted when you close the page.

# difference between e.propagation and immediate propagation? event bublling?

event.stopPropagation will prevent handlers on parent elements from running.  
event.stopImmediatePropagation will also prevent other handlers on the same element from running.

$("p").click(function(event){

event.stopImmediatePropagation();

});

$("p").click(function(event){

// This function won't be executed

$(this).css("background-color", "#f00");

});

# e.preventdefault ?

The event.preventDefault() method stops the default action of an element from happening.

$("a").click(function(event){  
    event.preventDefault();  
});

# what $(this); will do ?

this is a reference to the member that invokes the current function...