**JavaScript**

Inserting

* JavaScript can be inserted in the body and/or the head
* JavaScript can be inserted externally with a .js file
* Use <script> tags

Output Possibilities

* Alert Box – window.alert()
* HTML Output – document.write()
* HTML Input – innerHTML.
* Writing into a browser console – console.log()

Programs

* Uses variable (var x; x=6; or var x = 6;)
* Text is in quotations (“John Doe”)
* JavaScript is case sensitive

Statements (Instructions)

* Often uses: document.getElementById(“id”).innerHTML = “Display Text.”;
* CSS: function myFunction() {

Document.getElementById(“id”).innerHTML = “Display Text.”;

}

* 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 |

Comments

* Single Line Comment - // (Insert Text)
* Multi-line Comments - /\*

INSERT

TEXT

\*/

* The Comments above can be used to prevent the execution of some code for testing

Variables

* Can be used for algebra functions (var x = 1; var y = 2; var z = x + y; (which displays 3))
* X,Y, and Z above are identifiers and can be replaced with any different name, such as “sample”
* Variable can contain multiple data types, such as numbers and text
* To assign a value to a variable use the equal sign (var sample = “value”;)
* Multiple variables can be defined with one variable (var sample1 = “text”, sample2 = “123”;)

Operators

* Arithmetic Operators :

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

* Assignment Operators :

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |

* Comparison and Logical Operators :

|  |  |
| --- | --- |
| **Operator** | **Description** |
| == | equal to |
| === | equal value and equal type |
| != | not equal |
| !== | not equal value or not equal type |
| > | greater than |
| < | less than |
| >= | greater than or equal to |
| <= | less than or equal to |

Arithmetic

* Arithmetic Operators :

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

* Arithmetic Operators can be literals, variables, or expressions
* Operands in Arithmetic operations are the numbers, and the action performed is the operator
* Arithmetic follows order of operations, PEMDAS
* Operator Precedence Values :

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **Operator** | **Description** | **Example** |
| 19 | ( ) | Expression grouping | (3 + 4) |
|  |  |  |  |
| 18 | . | Member | person.name |
| 18 | [] | Member | person["name"] |
|  |  |  |  |
| 17 | () | Function call | myFunction() |
| 17 | new | Create | new Date() |
|  |  |  |  |
| 16 | ++ | Postfix Increment | ++i |
| 16 | -- | Postfix Decrement | --i |
|  |  |  |  |
| 15 | ++ | Prefix Increment | i++ |
| 15 | -- | Prefix Decrement | i-- |
| 15 | ! | Logical not | !(x==y) |
| 15 | typeof | Type | typeof x |
|  |  |  |  |
| 14 | \* | Multiplication | 10 \* 5 |
| 14 | / | Division | 10 / 5 |
| 14 | % | Modulo division | 10 % 5 |
| 14 | \*\* | Exponentiation | 10 \*\* 2 |
|  |  |  |  |
| 13 | + | Addition | 10 + 5 |
| 13 | - | Subtraction | 10 - 5 |
|  |  |  |  |
| 12 | << | Shift left | x << 2 |
| 12 | >> | Shift right | x >> 2 |
|  |  |  |  |
| 11 | < | Less than | x < y |
| 11 | <= | Less than or equal | x <= y |
| 11 | > | Greater than | x > y |
| 11 | >= | Greater than or equal | x >= y |
|  |  |  |  |
| 10 | == | Equal | x == y |
| 10 | === | Strict equal | x === y |
| 10 | != | Unequal | x != y |
| 10 | !== | Strict unequal | x !== y |
|  |  |  |  |
| 6 | && | And | x && y |
| 5 | || | Or | x || y |
|  |  |  |  |
| 3 | = | Assignment | x = y |
| 3 | += | Assignment | x += y |
| 3 | -= | Assignment | x -= y |
| 3 | \*= | Assignment | x \*= y |
| 3 | /= | Assignment | x /= y |

Assignment

* Assignment Operators :

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |

Data Types