Javascript

Comments

```
// this is a single line comment


/*
*
* This is a multi line comment
*
*/
```

Variable, arrays, objects

```
/*
*
* Javascript variables are not statically
* typed, so you can declare a variable
* without stating what type of data it is.
*
*/

var a = 25; // a number variable
var name = 'Renz' // a stringed variable
```

```
var aChar = '*'; // a character
var pie = 3.14; // float number

/*
*   Arrays concists of a set of data
*   or ordered lists of value
*/

var anArray = [25, 30, "Hello world"];
console.log(anArray[1]); // 30


/*
*
*   Objects are a declared by a variable
*   and inside that variable has a number
*   of sets of variables
*
*/

var theObjects = {
    fname: "Renz",
    lname: "Pulvira",
    age: 18
}

// outputting a certain data in an object
console.log(theObjects.fname);
```

Expressions & Operators

You can do basic **arithmetic** in javascript such as using +,-,/,* in manipulating or calculating data.

```
// You can do basic arithmetic in javascript
// and can use variables to manipulate data

var a = 20;
var b = 30;
var result = a + b; // (a)20 + (b)30 = result
console.log(result); // result = 50

// Addition
var result = a + b; // result = 50
// Subtraction
var result = a - b; // result = -20
// Division
var result = a / b; // result = 1
// Multiplication
var result = a * b; // result = 600
// Modulo division
var result = a % b; // 0.66
```

Console commands

```

// outputting data to the console
console.log("Hello world"); // >Hello world

// An object
var object = {
    fname: "Renz",
    lname: "Pulvira"
};

// Outputting data in a data tree view
console.dir(object);

// Show an alert message to the webpage
alert("This is an alert message");
```

Loops

```
/* FOR LOOP
*   A for loop needs 3 arguments
*   initialization, condition, expression
*/
for(int x = 0; x < 5; x++){
    console.log(x);
```

```
}


/* WHILE LOOP
*   A while loop, loops through a block of code
*   and will not stop depending on the condition given.
*/

while(x != 25){
    console.log(x);
}

/* DO WHILE LOOP
*   A do while loop, has a little resemblance to the whil
e
*   loop. do while loop runs the code atleast once. then
loops
*   through.
*/

do {
    console.log(x);
} while(x != 25);
```

Functions

```
// A simple function with no arguments
```

```
function aFunction(){
    console.log("Hello world");
}

// A function with arguments
function aFunction(a, b){
    var result = a + b;
    console.log(result);
}

// An anonymous function

/*
*   Anonymous functions consists
*   are activated/run automatically
*   when the page loads.
*/
(function(){
    console.log("hello world");
});

/*
*
*   There are already
*   Premade functions
*   i.e,
*/
```

```
function runThis(){
    console.log("HELLO");
}

/*
*   This will run every
*   3(3000) seconds
*
*/
setInterval(runThis, 3000);
```

Setters and getters

```
/*
*
*   Setters and getters are very good practice
*   Especially on OO programming. setters
*/

// get
var obj = {
    fname: "Renz",
    lname: "Pulvira",
    get fullName(){
        return this.fname + ' ' + this.lname;
```

```
    }
}

console.log(obj.fullName); // Renz Pulvira



// set
var obj = {
    age: null,
    set herAge(age){
        this.age = age;
    }
}

obj.herAge = 25;
console.log(obj.age); // 25
```

Classes

Javascript is a **class-less** language. you can achieve to create classes by using functions or objects. classes in javascript are **Special functions**.

```
function student(firstname, lastname, theDom){
    this.fname = firstname,
    this.lname = lastname,
```

```
    this.emailDom = theDom;
    this.theEmail = emailAdd;
}

function emailAdd(){
    return this.fname + '.' + this.lname + this.emailDom;
}

var theStudent = new student('Renz','Pulvira','@gmail.com
');
alert(theStudent.theEmail());
```