**How to access Objects**

**Dot notation**

***below example shows how to access an Object using a dot notation:***

var bob = {

name: "Bob Smith",

age: 30

};

var susan = {

name: "Susan Jordan",

age: 25

};

var name1 = bob.name;

var age1 = bob.age;

var name2 = susan.name;

var age2 = susan.age;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Bracket notation**

***below example shows how to access an Object using a bracket notation:***

var dog = {

species: "greyhound",

weight: 60,

age: 4

};

var species = dog["species"];

// save the weight and age using bracket notation

var weight = dog["weight"];

var age = dog ["age"];

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Methods**

***Methods*** are an important part of object oriented programming (OOP). OOP is an important part of programming, and it’s similar to ***functions***, but it’s associated with objects. Example of a method below:

var bob = new Object();

bob.name = "Bob Smith";

bob.age = 30;

// this time we have added a method, setAge

bob.setAge = function (newAge){

bob.age = newAge;

};

bob.setAge(20);

// bob's age is now ‘20’

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Why are ***methods*** important?

* Methods serve several important purposes when it comes to objects. The method ***setAge*** allows us to update ***bob.age.***
* They can be used to make calculations based on object properties. Functions can only use parameters as an input, but methods can make calculations with object properties. For example, we can calculate the year bob was born based on his ***age*** with our ***getYearOfBirth*** method.

**“This” keyword**

The keyword this acts as a placeholder, and will refer to whichever object called that method when the method is actually used. By using the keyword this, ***setAge*** will change the age property of any object that calls it. Previously, we had a specific object ***bob***instead of the keyword this. But that limited the use of the method to just ***bob***. Then when we say bob.setAge = ***setAge;*** it means whenever we type bob.setAge( ), this.age in the setAge method will refer to bob.age. ***Example below:***

var setAge = function (newAge) {

this.age = newAge;

};

var bob = new Object();

bob.age = 30;

bob.setAge = setAge;

bob.setAge(50); /// 🡨 This is how you change bob’s age to ‘50’

**Custom Constructors**

This constructor is used to make ***Person*** objects. Notice it uses the keyword ***this*** to define the ***name*** and ***age*** properties and set them equal to the parameters given.

Example below:

function Person(name,age) {

this.name = name;

this.age = age;

}

// Let's make bob and susan again, using our constructor

var bob = new Person("Bob Smith", 30);

var susan = new Person (“Susan Jordan”, 25);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Arrays of Objects**

Remember that an object is just another type, like a string or number but more complex. This means that just as we can make arrays of numbers and strings, we can also make arrays of objects. Example below:

// Our person constructor

function Person (name, age) {

this.name = name;

this.age = age;

}

// Below is how you make an array of people

var family = new Array();

family[0] = new Person("alice", 40);

family[1] = new Person("bob", 42);

family[2] = new Person("michelle", 8);

// add the last family member, "timmy", who is 6 years old

family[3] = new Person("timmy", 6);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Properties**

Properties are like variables that belong to an object, and are used to hold pieces of information. Properties can be accessed in two ways:

**Dot notation**: ObjectName.PropertyName;

**Bracket notation**: ObjectName["PropertyName"];

***Example below:***

var snoopy = new Object();

snoopy.species = "beagle";

snoopy.age = 10;

// save Snoopy's age and species into variables

// use dot notation for snoopy's species

var species = snoopy.species;

// use bracket notation for snoopy's age

var age = snoopy["age"];

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*