**JavaScript Datatypes**

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type. It can hold any type of values such as numbers, strings etc.

For example:

var a=40;//holding number

var b="JavaScript";//holding string

**JavaScript primitive data types:**

There are five types of primitive data types in JavaScript. They are as follows:

1.**String** :

It represents sequence of characters

e.g. “Hello world!”

2.**Number** :

Number represents the numeric value.

e.g. “123456789”

3.**Boolean** :

Boolean datatype contains value of either True or False .

e.g. “true” or “false”

4.**Undefined**:

The undefined type is a primitive type that has one special value undefined. By default, when a variable is declared but not initialized, it is assigned the value of undefined.

**Example:**

{

Let foo;

Console.log(foo); **// undefined**

Console.log(typeof foo); **// undefined**

}

5.**Null** :

The null type is the primitive data type that has only one value: null. Javascript defines that null is an empty object pointer.

**Example:**

{

Let foo= null;

Console.log(typeof foo); **// null**

}

**JavaScript non-primitive data types**

The non-primitive data types are as follows:

1.**Object** :

In JavaScript, an object is a collection of properties, where each property is defined as a key-value pair.

The following example defines an empty object using the object literal form:

var emptyObject = {};

The following example defines the person object with two properties:

var person = {

firstName: 'John',

lastName: 'Doe'

};

A property name of an object can by any string. You can use quotes around the property name if it is not valid [JavaScript identifier](http://www.javascripttutorial.net/javascript-variables/#identifiers).

For example, if you have a property first-name, you must use the quotes such as "first-name" but firstName is a valid JavaScript identifier so the quotes are optional.

If you have more than one property, you use a comma ( ,) to separate the pairs.

JavaScript allows you to nest object as shown in the following example:

var contact = {

firstName: 'John',

lastName: 'Doe',

email: 'john.doe@example.com',

phone: '(408)-555-9999',

address: {

building: '4000',

street: 'North 1st street',

city: 'San Jose',

state: 'CA',

country: 'USA'

}

2.**Array:**

It represents group of similar Values or Datatypes.

**Example:**

Var array\_marks=[85,89,95,86];

Console.log(array\_marks[1]);

**Output:**

89 **// array index always starts from 0 to n-1.**

**3.RegExp :**

A regular expression is a sequence of characters that forms a search pattern.

The search pattern can be used for text search and text replace operations.

**Using String search() with a Regular Expression:**

### **Example**

var str = "Visit Javascript";  
var n = str.search(/Javascript/i);

**The result in *n* will be**:

6 //it prints starting index value

**Using String replace() with a Regular Expression:**

### **Example**

Regular expression is incasesensitive

var str = "Visit Microsoft!";  
var res = str.replace(/microsoft/i, "Javascript");

**The result in *res* will be:**

Visit Javascript!