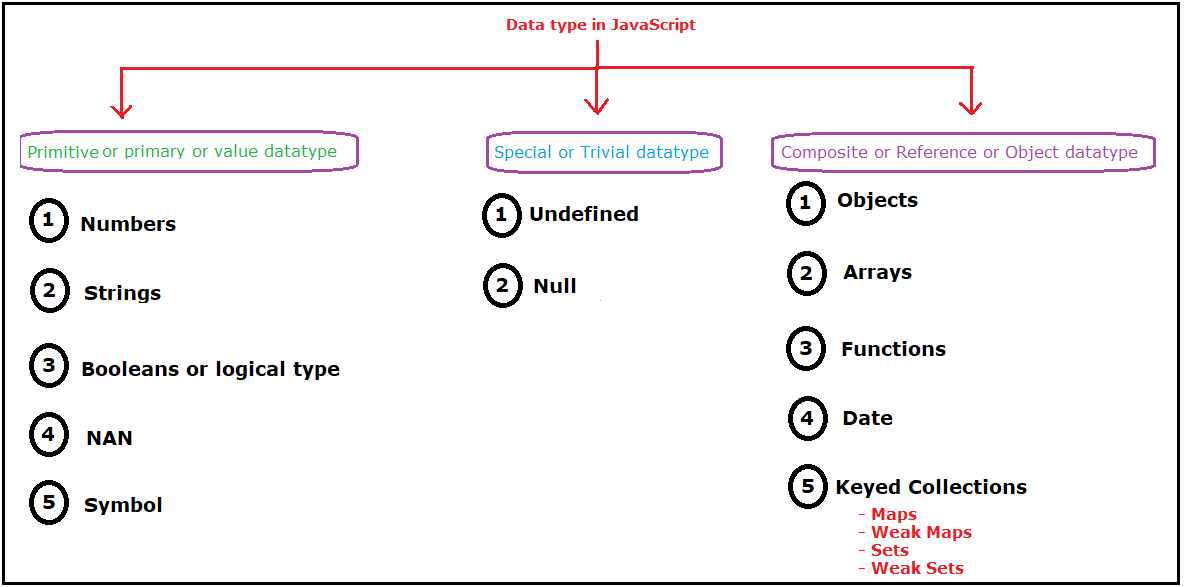
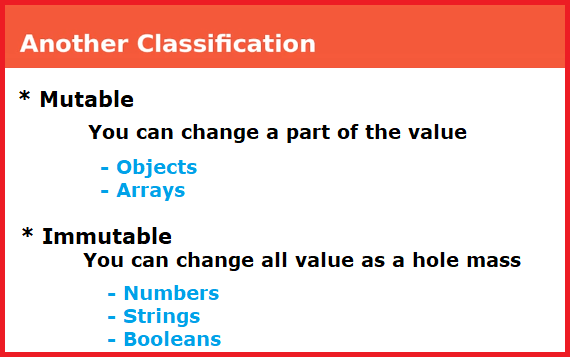
**JavaScript data type**

Data refers to information that can be stored in some memory location.

In JavaScript programming language there are many types of data and each type has its own values. We can classify these types into **three types**:



There is another classification for these data types depending on the ability to change data:



* + 1. **[Primitive or Primary Data Types]**

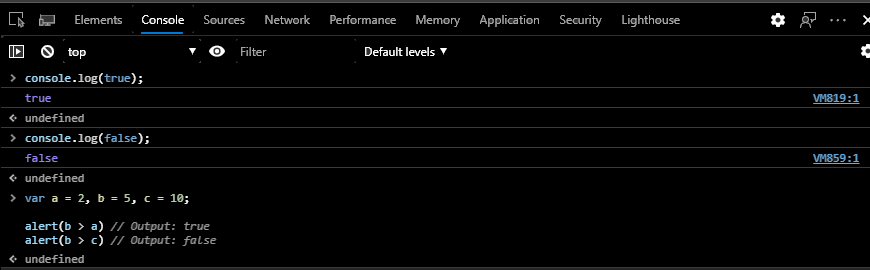
Primitive data types are the basic (fundamental) data type in JavaScript. It may be a

1. [Number](Numbers%20data%20type.docx).
2. [String](String%20data%20type.docx).
3. Boolean.
4. **Boolean**: indicates logical or conditional result. Boolean data type has two values true and false value. In JavaScript true indicates 1 and false indicates 0. It’s often used in conditional testing (if …else). (i.e. true or false).

Boolean data type has only two data type values as shown in this table.

|  |  |  |
| --- | --- | --- |
| Date type | Date type value | Description |
| Booleans or logical type | True | True means the condition was achieved. |
| false | True means the condition wasn’t achieved. |

**Ex:**



Primitive data types has been finished. And we will start with a special data type

* + 1. **[ Special Data Types]**
* Special Data Types are data types that have special uses and cases. It may be undefined or null type.

1. **undefined**: indicates un-initialized data type.

Un-initialized data type: means that there is identifier with an unknown value (memory box has a value (unknown value)).

**Note:** If we store undefined value in a memory box “identifier” then, its type will be set to undefined.

|  |  |  |
| --- | --- | --- |
| Date type | Date type value | Description |
| undefined | undefined | It’s the default value of un initialized variable & un initialized parameter in functions. |

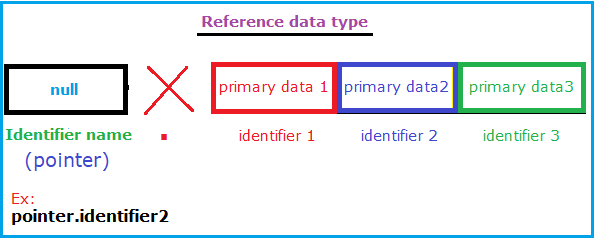


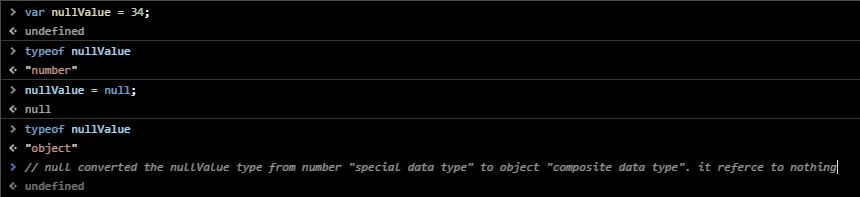
1. **null**: Actually null isn’t a JS data type it’s just a value, but still considered as a data type. it means the variable is **empty**

**null can deals as an object data type.**

If we store null in a value of a specific identifier its type convert from the current type to **object type**.

null is a special data type used to cut the connection between the pointer and the composite data type so it used with composite data type.in the example below we can notice that composite memory still storing it’s data on it, but the only change is the connection (arrow) was cut so no data from composite memory will appear in the browse view. For this reason programmers say that null is used to indicate to “nothing”





**Note:**

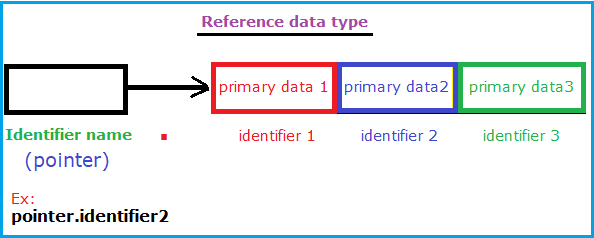
* 1. null reflects a lack of only value (type: object is presence) while undefined reflects a lack of both type and value (both value and type are absence).
  2. null is a primitive value but when you run typeof null you will get “object”, that’s a bug remnant from the first version of JS and one that unfortunately can’t be fixed, because it would break existing code. (search on net about the reason of this bug)

**3-[Composite Data Types]:**

Composite data types also called “**Reference types**” or “**complex types**” or “**Object types**”, are created by using several primitive data types.

Identifiers need to create **two** memory locations to storage this data. One is empty and works as a pointer the other holding composite data.

* The first is a pointer memory location “has the same name of identifier”, it doesn’t hold any data, it just used to refer or point to a specific (data) element in the second memory location.
* The second memory location holds composite data. It divided into many divisions, each division used to store specific data (element) on it.



* Composite data type may be: [Array](Reference/Array/Array.docx), objects…..

**There is a comparison between primitive and composite data types.**

|  |  |
| --- | --- |
| **Primitive value** | **Composite value** |
| hold data on itself | doesn’t hold data on itself |
| It doesn’t reference (point) to another memory box | it referees (point) to another memory box |
| It creates one memory location | It creates two memory location |
| **Examples:** | |