Variables

A variable is a “named storage” for data. We can use variables to store goodies, visitors, and other data.

To create a variable in JavaScript, use the “ var “ or “ let “ or “ const “

Ex:

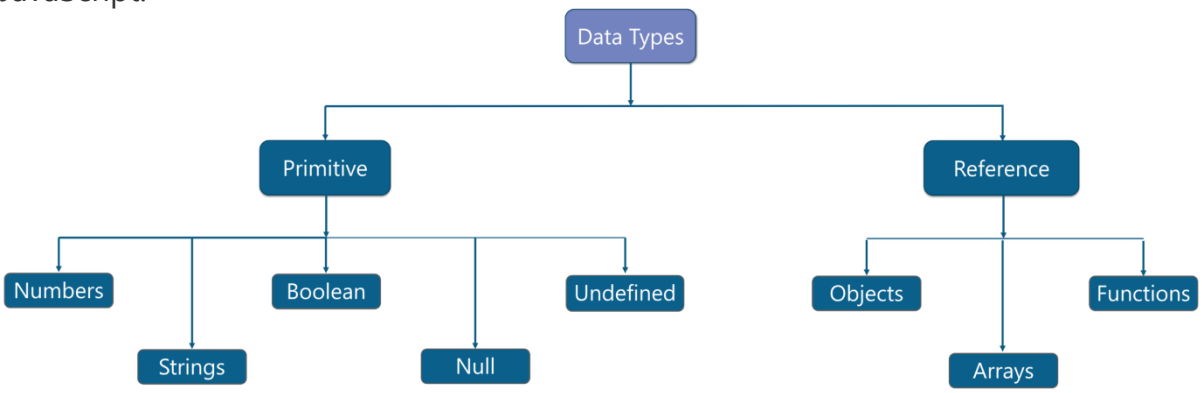
```
var x;

let y;

const z=10;
```

Data types

A value in JavaScript is always of a certain type. There are eight basic data types in JavaScript.



Memory storage

the JS memory model can be understood as having two distinct areas: the call stack and the heap.

The call stack is where primitives are stored (in addition to function calls).

The heap is where non-primitives are stored. The key difference is that the heap can store unordered data that can grow dynamically—perfect for arrays and objects.

Operators

- ✓ Assignment operators
- ✓ Comparison operators
- ✓ Arithmetic operators
- ✓ Bitwise operators
- ✓ Logical operators
- ✓ String operators
- ✓ Conditional (ternary) operator
- ✓ Comma operator
- ✓ Unary operators
- ✓ Relational operators

Conditional branching

If statement:

Syntax:

```
If(condition)
        {
                If-block
        }
```

If-else statement:

Syntax:

```
if(condition)
        {
                If-block
        }
else
        {
               Else block
        }
```

If-else if statement:

Syntax:

```
if(condition-1)
        {
                If-block
        }
Else if(condition-2)
        {

        }
.....................................
```

Switch statement

A `switch` statement allows a program to evaluate an expression and attempt to match the expression's value to a `case` label. If a match is found, the program executes the associated statement.

Syntax:

```
switch (expression) {
  case label_1:
    statements_1
    [break;]
  case label_2:
    statements_2
    [break;]
    …
  default:
    statements_def
    [break;]
}
```

Browser interaction functions

alert(string):

It shows a message and waits for the user to press “OK”.

prompt(string,string):

It shows a modal window with a text message, an input field for the visitor, and the buttons OK/Cancel. The first string is to show text to user. The second is an optional second parameter, the initial value for the input field.

confirm(string):

The function `confirm` shows a modal window with a `question` and two buttons: OK and Cancel.

The result is `true` if OK is pressed and `false` otherwise.

Functions

A function is a "subprogram" that can be *called* by code external (or internal in the case of recursion) to the function. Like the program itself, a function is composed of a sequence of statements called the *function body*. Values can be *passed* to a function, and the function will *return* a value.

A **function definition** (also called a **function declaration**, or **function statement**) consists of the function keyword, followed by:

- The name of the function.
- A list of parameters to the function, enclosed in parentheses and separated by commas.
- The JavaScript statements that define the function, enclosed in curly brackets, {...}.

Syntax:

```
function name-of-function (parameters)
{
        Definition of function
}
```

Primitive parameters (such as a number) are passed to functions **by value**; the value is passed to the function, but if the function changes the value of the parameter, **this change is not reflected globally or in the calling function**.

If you pass an object (i.e. a non-primitive value, such as Array or a user-defined object) as a parameter and the function changes the object's properties, that change is visible outside the function