* By default during declariation in javascript you have a value to a variable called undefined .we can overwrite using initialization.
* camelCasing is a naming convention in javaScript where if there are more than one word in a variable name then we give all lowercase for the first word and then we capitalize the first letter of the next word .

**VAR;**

* It is a datatype used in Javascript that declares a variable inside a function ,let helps us to define the variable within a block .
* We can create number, fractions, Boolean .
* We can never all ascii values of character to number like we do in C/C++.
* If we add a number with a string then string concatenation takes place instead of addition
* 7+”a”+6= 7a6
* 7+”5”+6=756
* Incase we use other operators like -,/,\* like the previous case we don’t get the result converted to string ,we get the numeric value.
* "37" - 7 // 30
* "37" + 7 // "377"
* parseInt() is used to convert string to numeric value
* parseInt(“94”,10<which base value to convert to >)= (int)94
* parseInt(“9a4”,10)=(int)9 //it stops with 9 because it encounters a non integer value a.
* parseInt(“s” ,10)= NaN //it means that the input is Not a Number.
* parseInt(3.14,10)=3 //since it doesn’t leave a number as a float.
* Parse float is used to convert input to float .
* parseFloat(3.14,10)= 3.14
* parseFloat(“3.14”,10)=3.14

**HOISTING:**

* It is a process in Javascript where the declarations(variable/function) are moved up to the first even if they are declared at the bottom.

**VARIABLE HOISTING:**

Refer variable hoisting in <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Grammar_and_types> .

**FUNCTION HOISTING:**

* first you must understand that the function can be declared as an expression also

function foo() {

console.log("bar");

}

* function declaration:

function foo() {

console.log("bar");

}

* with regard to hoisting functions declaration gets hoisted but function expression doesn’t.

/\* Function declaration \*/

foo(); // "bar"

function foo() {

console.log("bar");

}

/\* Function expression \*/

baz(); // TypeError: baz is not a function

var baz = function() {

console.log("bar2");

};

**IDENTITY OPERATOR(===):**

* IT IS USED TO COMPARE TWO OPERATORS and see if they are equal provided they are of the same type .

**NONIDENTITY OPERATOR(!==):**

* It is used to compare two elements and returns true if they aren’t true or they are not of the same type.

**IF:**

* Any number inside if other than 0 is true while 0 is false.
* Similarly if we have nothing inside “” it is false while any character even “ “ is true
* Undefined value also means false.