ECMAScript-6

1.What is ECMAScript?

ECMAScript (ES) is a scripting language specification standardized by ECMAScript International. It is used by applications to enable client-side scripting. The specification is influenced by programming languages like Self, Perl, Python, Java etc.

2.New features in ECMAScript:

* Support for constants
* Block Scope
* Arrow Functions
* Extended Parameter Handling
* Template Literals
* Extended Literals
* Enhanced Object Properties
* De-structuring Assignment
* Modules
* Classes
* Iterators
* Generators
* Collections
* New built in methods for various classes
* Promises

3.Let , Const and Var keyword:

*let* gives you the privilege to declare variables that are limited in scope to the block, statement of expression unlike *var*.

*var* is rather a keyword which defines a variable globally regardless of block scope.

var and let are both used for variable declaration in javascript but the difference between them is that var is function scoped and let is block scoped.

It can be said that a variable declared with var is defined throughout the program as compared to let.

4. Object-Oriented Programming Concepts:

* **Object** − An object is a real-time representation of any entity.
  + **State** − Described by the attributes of an object.
  + **Behaviour** − Describes how the object will act.
  + **Identity** − A unique value that distinguishes an object from a set of similar such objects.
* **Class** − A class in terms of OOP is a blueprint for creating objects. A class encapsulates data for the object.
* **Method** − Methods facilitate communication between objects.

Prior to ES6, creating a class was a fussy affair. Classes can be created using the class keyword in ES6.

### Example: Declaring a class

class Polygon {

constructor(height, width) {

this.height = height;

this.width = width;

}

}

## 5. Class Inheritance

## ES6 supports the concept of Inheritance. Inheritance is the ability of a program to create new entities from an existing entity - here a class. The class that is extended to create newer classes is called the parent class/super class. The newly created classes are called the child/sub classes.

### Example: Class Inheritance

'use strict'

class Shape {

constructor(a) {

this.Area = a

}

}

class Circle extends Shape {

disp() {

console.log("Area of the circle: "+this.Area)

}

}

var obj = new Circle(223);

obj.disp()

## Output:

Area of Circle: 223

## 6. The Super Keyword

## ES6 enables a child class to invoke its parent class data member. This is achieved by using the super keyword. The super keyword is used to refer to the immediate parent of a class.

class PrinterClass {

doPrint() {

console.log("doPrint() from Parent called…")

}

}

class StringPrinter extends PrinterClass {

doPrint() {

super.doPrint()

console.log("doPrint() is printing a string…")

}

}

var obj = new StringPrinter()

obj.doPrint()

## Output:

doPrint() from Parent called.

doPrint() is printing a string.