|  |  |  |  |
| --- | --- | --- | --- |
| xJS Primitive :They are immutable and it comes in pass by value | 1. String: Seq of char 2. Boolean: true/False 3. Number: 64-bit floating format 4. BigInt: 5. Symbol: 6. Undefined: 7. Null: 8. Object | | |
| Pass by value : |  | | |
| Pass by reference: Abstract Dt |  | | |
| IIFE() |  | | |
| HOF |  | | |
| this | At global level “this” is equal to global object.  Value of “this” inside the function is usually defined by the functions call.  In constructor a new obj Is created and set as the fn “this” argument  When calling a fn as a method of an obj, “this” refers to the obj.  In apply,call and bind “this” inside the fn is the obj that is passed in as the argument | | |
| Currying Fn | When a function instead of taking all arguments at one time, it takes the first one and returns a new function that takes a second one and returns a new function which takes the third one and so on.  function Myfunction(a) {          return (b) => {             return (c) => {               return a \* b \* c               }              }           } | | |
| Scope | Accessibility of variable | | |
| Null | Falsy value  It is assigned but there is no value  (Null === undefined) => false | | |
| Undefined | NaN value  It is not assigned or It is not yet defined  (Null == undefined) => true | | |
| Break   * Is used to terminate the loop * Used in loop and switch cases | | | Continue   * Used to continue the next iteration in the loop * Only used in loop |
| What is Generator function | | A new way of working with functions and iterators in the form of Generators.  A generator is a function that can stop midway and then continue from where it stopped  A generator appears to be a function but it behaves like an iterator. | |
| Class expression | | Is one way to define a class in ECMAScript. Similar to function expressions, class expressions can be named or unnamed. | |
| Weak Map | | Provides a way to extend objects from the outside without interfering with garbage collection. | |
| Template Literals | | String literal allowing embedded expressions.Enclosed by back-tick,  Enables dynamic expressions:  var greeting = `Welcome to JS World, Mr. ${firstName} ${lastName}.`  multi-line strings is possible | |
| Spread | | Allows iterable (arrays / objects / strings ) to be expanded into single arguments/elements.  function calculateSum(x, y, z) {  return x + y + z;  }  const numbers = [1, 2, 3];  console.log(calculateSum(...numbers)); | |
| pES6 Features | | 1. Support for constants or immutable variables 2. Block-scope support for variables, constants and functions 3. Arrow functions 4. Default parameters 5. Rest and Spread Parameters 6. Template Literals 7. Multi-line Strings 8. Destructuring Assignment 9. Enhanced Object Literals 10. Promises 11. Classes 12. Modules | |
| Arrow Functions | | Is a shorter syntax for a function expression and does not have its own this, arguments, super, or new.target  These fn are best suited for non-method functions, and they cannot to be used as constructor  Clean code, short syntax, best for call backs, | |
| Classical Inheritence   * limited to classes inheriting from other classes | | Prototype Inheritance   * Supports the cloning of any object using an object linking mechanism | |
| How to acces Prototype Object | | 1. Object.getPrototypeOf(myObj) 2. myObject.proto | |