**Tasks: Day01**

1. To read:
   1. <https://stackoverflow.com/questions/1517582/what-is-the-difference-between-statically-typed-and-dynamically-typed-languages>
   2. <https://stackoverflow.com/questions/17253545/scripting-language-vs-programming-language>
   3. <https://cs.lmu.edu/~ray/notes/paradigms/>
2. Write a blog on Difference between HTTP1.1 vs HTTP2
3. Write a blog about objects and its internal representation in JavaScript
4. Write a blog about objects and its internal representation in JavaScript
5. codekata practice

Q.1. To read

Ans: Read a., b., c. by opening given links.

Q.2. Write a blog on Difference between HTTP1.1 vs HTTP2

Ans:

* HTTP 2.0 is a binary protocol that numerous streams going over a single TCP connection
* The contents of each stream are HTTP 1.1 requests and responses, just encoded and packed up differently
* HTTP 2.0 can push resources to the client even without the client requesting them
* HTTP/1.1 allows only one request
* HTTP/2 allows clients to make multiple requests over one TCP connection
* http/2 is evolution, and brings new features like web push. And http/2 much faster than http/1.1, better data compression, better security
* The contents of each stream are HTTP 1.1 requests and responses, just encoded and packed up differently. HTTP2 adds a number of features to manage the streams, but leaves old semantics untouched.

Q.3. Write a blog about objects and its internal representation in JavaScript

Ans: Objects are the representation of real-world entities in any language representing things by defining its properties along with their values. In Javascript, objects may be defined as an unordered collection of related data, of primitive or reference types, in the form of **“key: value”** pairs.

Ways to create an object in JavaScript are as follows-

* Object literal

object literal is a comma-separated list of name-value pairs wrapped in curly braces. Object literals encapsulate data, enclosing it in a tidy package.

var car={id:1 , name:’abc’ , display: function() }

As evident from the above example property values can be of any data type, including array literals, functions, nested object literals, or primitive data type.

* Object.create()

The method creates a new object, using an existing object as the prototype of the newly created object.

using the object literal example as prototype-

var car2 = Object.create(car);

car.id=2;

car.name = ‘xyz’;

* Object constructor

Useful when we require to create multiple objects of similar type. In this case, a constructor (kind of blueprint) is created and multiple objects can be initialized using the new keyword using the constructor as a wrapper for the newly created objects.

construction function-

function Person(name, age, eye) {  
this.Name = name;  
this.age = age;  
this.eyeColor = eye;  
}

creating objects using constructor-

var p1= new Person(“John”, 50, “blue”);  
var p2= new Person(“Sally”, 48, “green”);

* Object.assign()

It is used to copy the values and properties from one or more source objects to a target object. It invokes getters and setters since it uses both [[Get]] on the source and [[Set]] on the target.

Here is an example where properties from three source objects are getting assigned to a new object using Object.assign()

Input : var obj1 = { a: 10 };  
var obj2 = { b: 20 };  
var obj3 = { c: 30 };  
var new\_obj = Object.assign(o1, o2, o3);  
console.log(new\_obj);  
Output : Object { a: 10, b: 20, c: 30 }

* Object.fromEntries

This method transforms a list of key-value pairs into an object.

const entries = new car([  
[‘id’, 4],  
[‘color’, ‘blue’]  
]);

const car1= Object.fromEntries(entries);

console.log(car1);  
output: Object { id: 4, color: ‘blue’}

Unlike other object-oriented programming languages, javascript doesn’t have classes instead of that javascript is a prototype-based language allowing all the functionalities as in other class-based programming languages like JavaScript allows you to create hierarchies of objects and to have the inheritance of properties and their values and all this is done mainly using the constructor functions.

Q.4. codekata practice

Ans: I am practicing in codekata.