**FULL-STACK DEVELOPEMENT -- GUVI**

**REGISTERED NAME: RANI D V  
REGISTERED E-MAIL: [ranidvhsr31@gmail.com](mailto:ranidvhsr31@gmail.com)**

**TASK - 1:**

**To read:**

1. **What is the difference between Static type and dynamic typed languages?**

**Static language:**

Its a language where the type of language known at compile time. There are three types of variables in java script they are ‘var’, ‘let’, ‘const’.  
Example: in java: String str = “Hello”; // str is a static variable by default.

Str = 5; //since str is a string error will be occurred,

Advantages:  
 Some trivial errors are got caught by the early stage since it got checked at the time of compilation.

**Dynamically typed language:**

A language is dynamically typed where the type of variable values are known at run time.

Example: Java script, python.

In python, str = ‘Hello”   
 str = 5 // No error occurs until we define with datatype.

1. **Scripting language vs Programming language.**

**Scripting language:**

Scripting language are programming languages that do not requires an explicit compilation step, i.e., we don’t have to compile the scripting language explicitly.

A scripting language is a language that used to modify, customize and manipulate the facility of an existing system.

Scripting language was created for users to create new functionality to provide an interface to their Byte code.

Advantages:

Run time is fast since it hasn’t to be compiled.

**Programming languages:**

It is the language compiled to machine code and run on hardware that is operating system.

A programming language uses the computer as a virtual machine.  
 A programming language is used to code a system that is used to operate the a proposed system.

1. Write a blog on Difference between HTTP1.1 vs HTTP2.  
    HTTP - Hypertext transfer protocol which is the method in computers and servers use to request and send information.

|  |  |
| --- | --- |
| **HTTP/1.1** | **HTTP 2** |
| * The first version of HTTP was HTTP/1.1 and created in the 1997 and still in use. | * The latest version of HTTP is version 2.0, created in the year 2015. |
| * **Prioritization:**   HTTP/1.1 is unaware of the prioritization of the web contents to be displayed to the user,which makes the user unacknowledged about the loading of their required information. | * **Prioritization:**   HTTP/2.0 does anticipates the problems of HTTP/1.1 that user undergone regarding the priority of contents to be displayed, this makes the user to be aware of loading time. |
| * In HTTP/1.1 when the client request for any data it sends streams of data one after another, so it has much loading time. | * It uses the concept of Multiplexing, that when a client request for a data its sends several streams of data at once by each of the streams at a different weighted values |
| * HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. | * In contrast, HTTP/2 is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) (Transmission control protocol) connection to send multiple streams of data at once so that no one resource blocks any other resource by splitting data into binary-code messages |

1. **Programming paradigm:**

A Programming Paradigm is a style, or “way,” of programming.

**Imperative:** Programming with an explicit sequence of commands that update state.  
**Declarative:** Programming by specifying the result you want, not how to get it.  
**Structured:** Programming with clean, goto-free, nested control structures.  
**Procedural:** Imperative programming with procedure calls.  
**Functional:** (Applicative): Programming with function calls that avoid any global state.  
**Functional level**: Programming with no variables at all.  
**Oriented**: Programming by defining objects that send messages to each other. Objects have their own internal (encapsulated) state and public interfaces.  
 Object orientation can be:  
**Class-based**: Objects get state and behavior based on membership in a class.  
**Prototype-based**: Objects get behavior from a prototype object.  
**Event Driven**: Programming with emitters and listeners of asynchronous actions.  
**Flow-Driven**: Programming processes communicating with each other over predefined channels.  
**Logic:** (Rule-based): Programming by specifying a set of facts and rules. An engine infers the answers to questions.  
**Constraints**: Programming by specifying a set of constraints. An engine finds the values that meet the constraints.  
**Aspect-Oriented**: Programming cross-cutting concerns applied transparently.  
**Reflective**: Programming by manipulating the program elements themselves.  
**Array**: Programming with powerful array operators that usually make loops unnecessary.  
  
2. **Objects and its internal representation in JavaScript:**  
 It’s most important data-type and forms the building blocks for modern JavaScript.   
 These objects are quite different from JavaScript’s primitive data-types(Number, String, Boolean, null, undefined and symbol) in the sense that while these primitive data-types all store a single value each.  
**Internal representation:**  
 It contain any combination of these primitive data-types as well as reference data-types.  
 You access the properties of an object with a simple dot-notation.  
 Ex: objectname.propertyname

Unassigned properties of Object are Undefined.  
 Properties of JavaScript objects can also be accessed or set using a bracket notation.  
 Objects are sometimes called *associative arrays*, since each property is associated with a string value that can be used to access it.   
 A property name that has a space or a hyphen, or that starts with a number) can only be accessed using the square bracket notation.  
 myobj.[str]