1. **What is the different between programming language and scripting language?**

|  |  |
| --- | --- |
| **Programming Language** | **Scripting Language** |
| A programming language is a computer language that is used to communicate with computers using a set of instructions. | A scripting language is a type of programming language designed for a runtime system to automate the execution of tasks. |
| It is compiled language or compiler-based language. | It is interpreted language or interpreter-based language |
| It is used to develop an application or software from scratch. | It is used to combine existing components and automate a specific task. |
| It runs or executes independently and does not depend on the parent (exterior) program. | It runs or executes inside another program. |
| It uses a compiler to convert source code into machine code. | It uses an interpreter to convert source code into machine code. |
| As it uses a compiler, hence the complete program is converted into machine code in one shot. | As it uses an interpreter, hence the program is converted into machine code line by line. |
| These languages are required to be compiled. | There is no need for compilation. |
| There is the high maintenance cost. | There is less maintenance cost. |

**2.What is the static typing and dynamic type?**

**Static Typing**

* Static typing means that types are known and checked for correctness before running your program.
* This is often done by the language's compiler

## Dynamic Typing

## while dynamically-typed languages perform type checking at run-time.

## while dynamically-typed languages do not.

**3.Paradigm**

* **Paradigm**can also be termed as method to solve some problem or do some task. Programming paradigm is an approach to solve problem using some programming language or also we can say it is a method to solve a problem using tools and techniques that are available to us following some approach. There are lots for programming language that are known but all of them need to follow some strategy when they are implemented and this methodology/strategy is paradigms. Apart from varieties of programming language there are lots of paradigms to fulfil each and every demand. They are discussed below
* **Imperative programming paradigm:**
* **Procedural programming paradigm**
* **Object oriented programming**
* **Parallel processing approach**
* **Declarative programming paradigm**
* **Logic programming paradigms**
* **Functional programming paradigms**
* **Database/Data driven programming approach**

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

HTTP/1.1

* It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution.
* Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient.
* It is relatively secure since it uses digest authentication, NTLM authentication
* Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags.

**HTTP/2**

* Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested.

Underlying semantics of HTTP such as headers, status codes remains the same.

* Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security.
* HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream.

HTTP/2 utilizes multiplexing and server push to effectively reduce the page load time by a greater margin along with being less sensitive to network delays.

**3.Write a blog about objects and its internal representation in Javascript**

* In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.
* Like all JavaScript variables, both the object name (which could be a normal variable) and property name are case sensitive. You can define a property by assigning it a value. For example, let’s create an object named my car and give it properties named make, model, and year as follows.