# **The tasks of Day 1 & Day 2 are both combined in this file**

# 

# ***Javascript - Day -1: Introduction to Browser & web Task***

1. **To read:**
   1. [**https://stackoverflow.com/questions/1517582/what-is-the-difference-between-statically-typed-and-dynamically-typed-languages**](https://stackoverflow.com/questions/1517582/what-is-the-difference-between-statically-typed-and-dynamically-typed-languages)

Statically typed languages:-

* Type of variable is known at compile time.
* Must specify the type of each variable
* Checking is done by the compiler so bugs can be caught early
* Examples: C, C++, Java

Dynamically typed languages:-

* The type of variable is known at run time.
* Speedy coding as not have to specify type each time
* Mostly scripting languages
* You may need to find bugs your-self
* Examples: Python, PHP, JavaScript

* 1. [**https://stackoverflow.com/questions/17253545/scripting-language-vs-programming-language**](https://stackoverflow.com/questions/17253545/scripting-language-vs-programming-language)

Scripting languages:-

* No need for an explicit(direct) compilation step.
* They are interpreted (line by line).
* The interpreter will read and analyze the code statements each time it meets them and halts at that very instance if there is some error
* Used to automate certain tasks in the program, extracting information from a data set, and are less code-intensive.
* Examples: JAvaScript, PHP, Python

Programming languages:-

* The compilation is needed before running as in C.
* Run faster as they are first converted into native code
* Compiler read and analyze the code only once and report errors collectively that the code might have
* Typically run inside a parent program, More compatible while integrating code with mathematical models, languages like JAVA can be compiled and then used on any platform.
* Examples: C, C++

The difference between the two is mixing up or we can say getting blurred due to improved computational capabilities of the modern hardware and advanced coding practices.

1. **Code kata practice**
2. **Difference between HTTP1.1 vs HTTP2**

|  | HTTP 2 | HTTP 1.1 |
| --- | --- | --- |
| Prioritization | The developers have control over prioritization on page load speed. | Not possible or up to an extent as in HTTP2 |
| Multiplexing | It is able to use a single [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection to send multiple streams of data at once so that no one resource blocks any other resource | loads resources one after the other |
| Server Push | Allows a server to "push" content to a client before the client asks for it. | a server only serves content to a client device if the client asks for it |
| Header compression | Uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets | To speed up web performance compress HTTP messages to make them smaller |

***Javascript - Day -2 : Request & Response cycle Task***

1. **List 5 difference between Browser JS(console) v Nodejs**

| BROWSER JS | NODE JS |
| --- | --- |
| It’s a programming language | Its a JRE(Javascript Runtime Environment) |
| Can only be run in browsers | JS can run outside the browser using NODE-JS |
| Mostly used on the client-side for front-end development | Mostly used at server side for server-side development |
| Capable to add HTML and play with DOM (Document Object Model) | Does node have the capability to add HTML tags |
| JS Framework example includes RamdaJS, TypedJS, etc | Node JS module examples include Lodash, express, etc. |

1. **watch & summary 5 points -** [**https://www.youtube.com/watch?v=SmE4OwHztCc&ab\_channel=JSConf**](https://www.youtube.com/watch?v=SmE4OwHztCc&ab_channel=JSConf)
2. **To read -** [**https://stackoverflow.com/questions/5641997/is-it-necessary-to-write-head-body-and-html-tags**](https://stackoverflow.com/questions/5641997/is-it-necessary-to-write-head-body-and-html-tags)

Normally we all write HTML code with Tags. But earlier HTML was written without defining Html, head, and body tags thus HTML 2 was made keeping this in mind. So omitting these tags is certainly allowed by the HTML specs.

But then also most of us write tags for making code look familiar.

1. **Execute the below code and write your description in txt file**
   1. **typeof(1)**
   2. **typeof(1.1)**
   3. **typeof('1.1')**
   4. **typeof(true)**
   5. **typeof(null)**
   6. **typeof(undefined)**
   7. **typeof([])**
   8. **typeof({})**
   9. **typeof(NaN)**

console.log("typeof(1) => " + typeof(1));

console.log("typeof(1.1) => " + typeof(1.1));

console.log("typeof('1.1') => " + typeof('1.1'));

console.log("typeof(true) => " + typeof(true)) ;

console.log("typeof(null) => " + typeof(null));

console.log("typeof(undefined) => " + typeof(undefined));

console.log("typeof([]) => " + typeof([]));

console.log("typeof({}) => " + typeof({}));

console.log("typeof(NaN) => " + typeof(NaN));

\*above code is copied from my vs code thus….

typeof(1) => number

typeof(1.1) => number

typeof('1.1') => string

typeof(true) => boolean

typeof(null) => object

typeof(undefined) => undefined

typeof([]) => object

typeof({}) => object

typeof(NaN) => number

1. **Read what is a prototype**

In software development, a prototype is a rudimentary working model of a product or information system, usually built for demonstration purposes or as part of the development process

Website Prototype

A prototype, in terms of web design, is an interactive mockup of your web design

Object Prototype in JS

Prototypes are the mechanism by which JavaScript objects inherit features from one another