ASSIGNMENT-3

Adith
ENG18CS0017

1) MVC

The model-view-controller pattern of designing software was one of first architectural design patterns based on the responsibilities of its component software constructs. Trygve Reenskaug introduced the concept of MVC while visiting XEROX Parc in the 1970's

MVC was revolutionary then and is still a key concept when discussing how to create new software. When using an MVC pattern, you can lay out the solution to any problems you expect to solve with the software you intend to build. The three components included by name in the pattren are:

-> The model, which describes the behaviour of the application regarding data, logic and rules.

-> The view, which includes any information that the application may output & the controller, which accepts and translates ~~outp~~ input

The continuing Evolution of MVC and isomorphic Javascript

While the concept of MVC first evolved for desktop applications, the pattern has been updated as a key approach for web based applications. Even today, many common MVC frameworks such as Ruby on Rails, put most of the burden of the model, view & controller logic on the server.

However, modern web applications have evolved to take advantage of maturing client, technologies, allowing new frameworks to handle more of the MVC login on the client. The result is isomorphic Javascript, an approach that enables you to use MVC pattern across both the client & the server.

2) Debugging in JS

Debugging is not easy. But fortunately, all modern browsers have a built in Javascript debugger. Built-in debuggers can be turned on & off, forcing errors to be reported to the user.

3 main methods are

i) Console Log() method
ii) setting breakpoints
iii) Debugger keyword.

i) Console log() method:
This is a function in javascript which is used to print any kind of variables defined before in it or to just print any message that needs to be displayed by the user.
Syntax:
Console.Log(A)

2) Setting breakpoints:

In the debugger window, you can set breakpoints in the javascript code.
At each breakpoint, javascript will stop executing, and let you examine javascript values. After examining values

you can resume the execution of code.

3) The debugger keyword !

The debugger keyword stops the execution of Javascript, and calls (if available) the debugging function
This has the same ~~fnx~~ as ~~a~~ setting a breakpoint in the debugger..
If no debugging is available, the debugger statement has no effect.
with the debugger turned on, this code will stop executing before it executes the third one

Syntax

debugger;