Suppose that you were to embark on creating a new programming language for the Web. From what we have learned about existing languages so far, describe how you would go about it. Include in your discussion necessary components you may need to develop for:

- Language definition
- Processing of programs written in that language
- Execution of processed programs in that language

Language definition: In my language syntactic definition should be rather strict. No independence should be given in the form syntax. For e.g. if I want my users to end a statement with a certain character, they are required to put it at the end of each sentence. For e.g. languages like Javascript & PHP (probably) makes it optional for the users to put a ';' at the end of a statement. This leads to a lot of confusion. When an user does not put a ';' it produces entirely different results compared to when they put it. For the same piece of code, let us look at another example. The for( ; ; ) loop & it \[ z \] & else in 'C' language. Since it is not mandatory not to provide the \[ z \], many users skip it & they have got different results.
from what they are expecting. In my language, the placing of `else & else statement will be a must even if that mean they have to put the braces for a single statement. With `if` they do not have an else logic, they have to create an `else statement & write void. Moreover I would not dynamic instantiation of variable, since I have seen naive programmers trying to instantiate objects/variables in a loop. In my language variable have to instantiated at the beginning of a method.

b) Processing of program written in that language:—

I would have a Compiler. Compiling any code. The Compiler front and would scan from the tokens in the source code. Then the parser would take over. The parser would parse the code for any syntax errors. Next if the code is upto the mark it will create an abstract parse tree. Next the code optimizer would be running on the source code. Next the intermediate code generator would take over. If we take a black-box view of the Compiler, the intent
will be the source code & the output will be intermediate code (bytecode) which is ready to run on a virtual machine.

c) Execution of processed program: Here I would not like to go with the rest. Instead, I would a popular VM to interpret my intermediate code into (?) machine readable format. The reason being very simple. A popular VM like JVM will be installed in almost all the computers, thus the users will not have to worry about installing another VM on his system. For example, the JVM. It has the capability of recognizing hotspots & implement JIT compiling which I would like to incorporate.

_____________________
ATANU