

Chapter 14. Script Languages

Byoung-Tak Zhang

TA: Hanock Kwak

Biointelligence Laboratory

School of Computer Science and Engineering

Seoul National University

<http://bi.snu.ac.kr>

Script Languages

- **Unix shells: sh, ksh, bash**

- job control

- **Perl**

- Slashdot, bioinformatics, financial data processing, cgi

- **Python**

- System administration at Google
- BitTorrent file sharing system



- **Ruby**

- Various blogs, data processing applications

- **PHP**

- Yahoo web site



- **JavaScript**



Characteristics

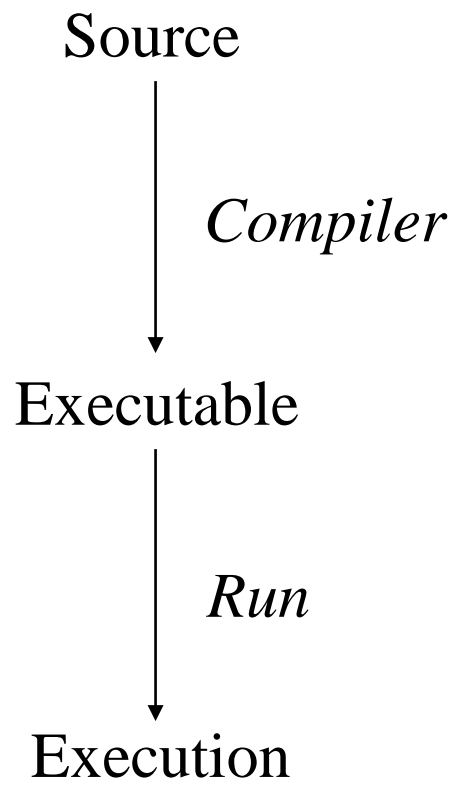
- Interpreted (no compilation step)
- Easy to learn
- Dynamically typed
- High-level model of underlying machine
- Built-in Data Structures
- Garbage collected
- Don't have to declare variables

- **Designed to support “quick programming”**

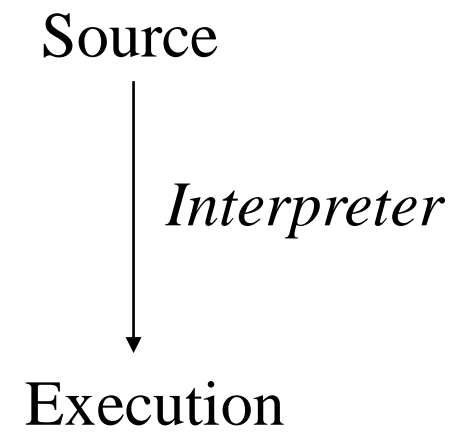
Design philosophy

Often people, especially computer engineers, focus on the machines. They think, "By doing this, the machine will run faster. By doing this, the machine will run more effectively. By doing this, the machine will something, something, something." They are focusing on the machines. But in fact we need to focus on humans, on how humans care about doing programming or operating the application of the machines. We are the masters. They are the slaves.

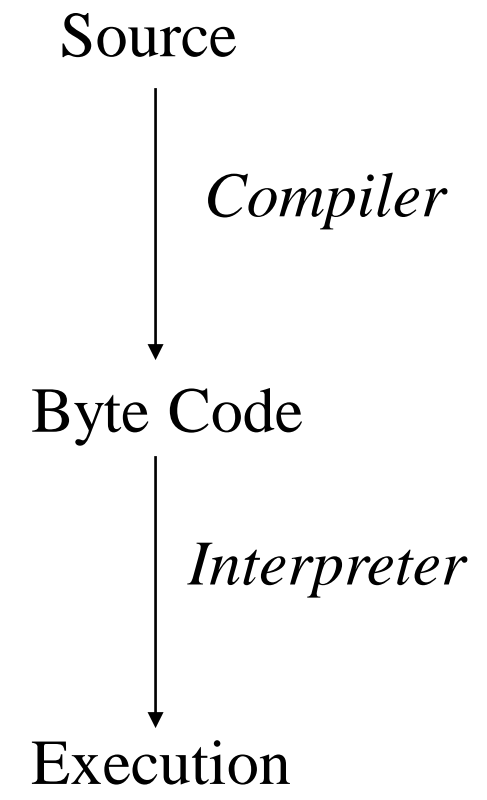
Yukihiro "Matz" Matsumoto
Creator of Ruby



**Compiled
Execution**



**Interpreter
Execution**



**Byte Code
Execution**

Act as Glue

- Scripting languages are often used to combine the functionality of other programs.
 - They act as glue.
- This allows the script to act as the intermediary between programs and pass information between them.

Extending

- External programs can be used to increase the functionality of the scripting language by binding existing programs to it
 - This is called extending.
- This allows function calls to be made directly to the compiled programs instead of through command interface

Dynamic Typing

- Scripting languages often support dynamic typing.
- The system manages the types of variables without the programmer's explicit input on matters of length and type declarations.

```
[javascript]  
b = "abc";  
b = 293; // type changes!!  
b = 1.0; // type changes again!!
```


Memory Managements

- Automatic memory management controls the allocation and freeing (garbage collection) of memory on demand.
 - Objects can grow and shrink as needed and are removed when no longer necessary.

[Matlab]

```
b = zeros(10000000, 1); % very big vector  
% no need to free explicitly in the code
```

Object Oriented

- Many scripting languages adopt object-oriented structures.
- Traditional scripting languages tend to become difficult to manage when used to write larger programs and the inclusion of OO is an attempt to address the problem.

[Python]

```
class Complex:
```

```
    def __init__(self, realpart, imagpart):
```

```
        self.r = realpart
```

```
        self.i = imagpart
```

Dynamic Codes

- Many scripting languages can dynamically create and execute code during the execution of the script.
 - This generally cannot be done by a non-scripting language.

```
[php]  
$number = 1;  
eval ("$number = $number + 1");  
echo $number;
```

Data Structures

- Most modern scripting languages have built-in support for high level data structures.

```
[php]
```

```
$myarray = array("a"=>"Apples", "b"=>"Oranges", "c"=>"Pears");  
$a = $myarray["a"];  
echo $a; // Apples
```

String Operations

- Most modern scripting languages supports high-level string operations.

```
[javascript]  
x = 5 + 5;  
y = "5" + x;  
z= "Hello" + y; // Hello510
```

```
[python]  
x = "a"*10 // aaaaaaaaaa
```

Debugging

- Errors are easily found in script languages because of its flexibilities in the grammars.
- With some managed runtime environment, debugging is also possible for some script languages.

Debugging Javascript with Chrome

The image shows the Chrome Developer Tools interface with several annotations:

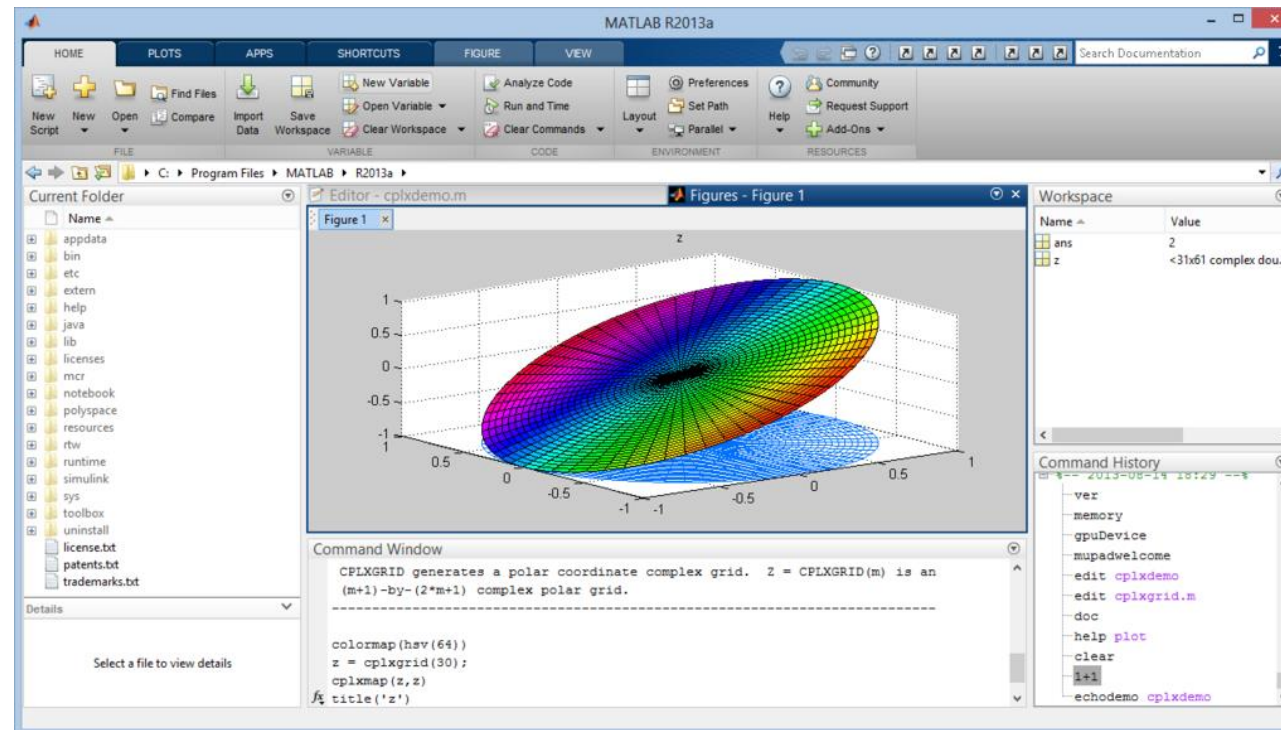
- Use Pause/Resume, Step Over, Step Into & Step out to debug js code**: A red circle highlights the execution control buttons (Pause, Step Over, Step Into, Step Out) in the top toolbar.
- Add js objects that you want inspect**: A red circle highlights the "Add" button in the Watch Expressions panel, which currently shows "myObj: undefined".
- Add breakpoints by clicking on the line number**: A red circle highlights the line number "18" in the source code editor, next to the line `var myObj = new myCFCProxy();`.
- View all the breakpoints**: A red circle highlights the "Breakpoints" section in the right-hand sidebar, which shows a list of breakpoints including the one at line 18.
- Dock to the main window, show console, pause on exceptions**: A red circle highlights the bottom toolbar icons for Docking, Console, and Pausing on exceptions.

```
1 <script type="text/javascript"> /*  */_cf_loadingtexthtml="i
2 _cf_contextpath="";
3 _cf_ajaxscriptsrc="/CFIDE/scripts/ajax";
4 _cf_jsonprefix="//";
5 /* ]]]&gt; */&lt;/script&gt;&lt;script type="text/javascript" src="/CFIDE/scripts/
6 &lt;script type="text/javascript" src="/CFIDE/scripts/ajax/package/cfaj
7
8 &lt;script type="text/javascript"&gt; /* <![CDATA[ */
9 ColdFusion.Ajax.importTag('CFAJAXPROXY');
10 /* ]]]&gt; */&lt;/script&gt;
11
12 &lt;script type="text/javascript"&gt; /* <![CDATA[ */
13 var _cf_TestCFC=ColdFusion.AjaxProxy.init('/tutorials/AjaxProxy/T
14 _cf_TestCFC.prototype.retrieveAuthorDetails=function() { return
15 /* ]]]&gt; */&lt;/script&gt;
16
17 &lt;script type="text/javascript"&gt;
18 var myObj = new myCFCProxy();
19
20
21 &lt;/script&gt;
22
23
24
25</pre></div>
```

MATLAB

■ MATLAB

- MATLAB (matrix laboratory) is a multi-paradigm numerical computing **environment** and fourth-generation programming language.

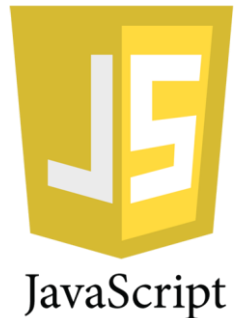


MATLAB

- MATLAB provides a language and environment for numerical computation, data analysis, visualization and algorithm development
- MATLAB provides functions that operate on
 - Integer, real and complex numbers
 - Vectors and matrices
 - Structures

JavaScript

- JavaScript is a dynamic programming language
- It is most commonly used as part of web browsers
 - whose implementations allow client-side scripts to interact with the user, control the browser,
- It is also used in server-side network programming with runtime environments such as Node.js, game development and the creation of desktop and mobile applications.
- JavaScript is not a Java !!



Python

- Python is a widely used general-purpose, high-level programming language.
- Design philosophy
 - code readability,
 - its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.

Python is powerful... and fast;
plays well with others;
runs everywhere;
is friendly & easy to learn;
is Open.

These are some of the reasons people who use Python would rather not use anything else.



Python

- Python is a multi-paradigm programming language
 - object-oriented programming
 - structured programming
 - functional programming
 - aspect-oriented programming



[Guido van Rossum](#), the creator of Python.

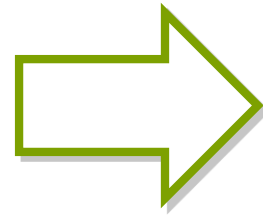
More Topics: Functional Languages

- It is a programming paradigm that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.
 - Its like a math
- Most popular functional languages
 - Haskell
 - Erlang



Functional Languages

```
total = 0;  
for (i = 1; i ≤ 10; ++i)  
    total = total+i;
```



```
sum[1..10]
```

Functional Languages

```
x :: Int
```

```
x = 4
```

- = does not denote “assignment” like it does in many other languages
- = denotes definition, like it does in mathematics.
 - $x = 4$ should not be read as “x gets 4” or “assign 4 to x”,
 - but as “x is defined to be 4”.

Functional Languages

■ Quiz: What's the meaning of the following codes?

- $x:xs$: element x goes inside the list xs
- $++$: list concatenation

$$f [] = []$$
$$f (x:xs) = f ys ++ [x] ++ f zs$$

where

$$ys = [a \mid a \leftarrow xs, a \leq x]$$
$$zs = [b \mid b \leftarrow xs, b > x]$$

More Topics: Aspect Oriented Programming

- Suppose that you want to measure times of all the functions you made.
 - Then you need to put time measuring codes to all that functions 😞
 - In aspect oriented programming, something like following can be done.

```
long startTime = getCurrentTime();  
invocation.invokeNext(); // function call !  
long elapsed = startTime - getCurrentTime();  
print(elapsed);
```

Aspect Oriented Programming

- Aspect-oriented programming (AOP) is a patented programming paradigm that aims to increase modularity by allowing the **separation of cross-cutting concerns**.
- In very simplified view, you can do, for instance
 - check the integer arguments of every functions whose names start with 'calculate'.
 - throw an exception when classes in package named 'kr.ac.snu.bi.prog' are constructed at night (10p.m ~ 6a.m)