



Android Development Tutorial (Basics)

PRESENTER : JACK LIU ZHEMIN

Agenda

- Compiled vs Interpreted Languages (20 Minutes)
- Communications between Activities (15 Minutes)
- Activity Interoperability (20 minutes)

Disclaimer

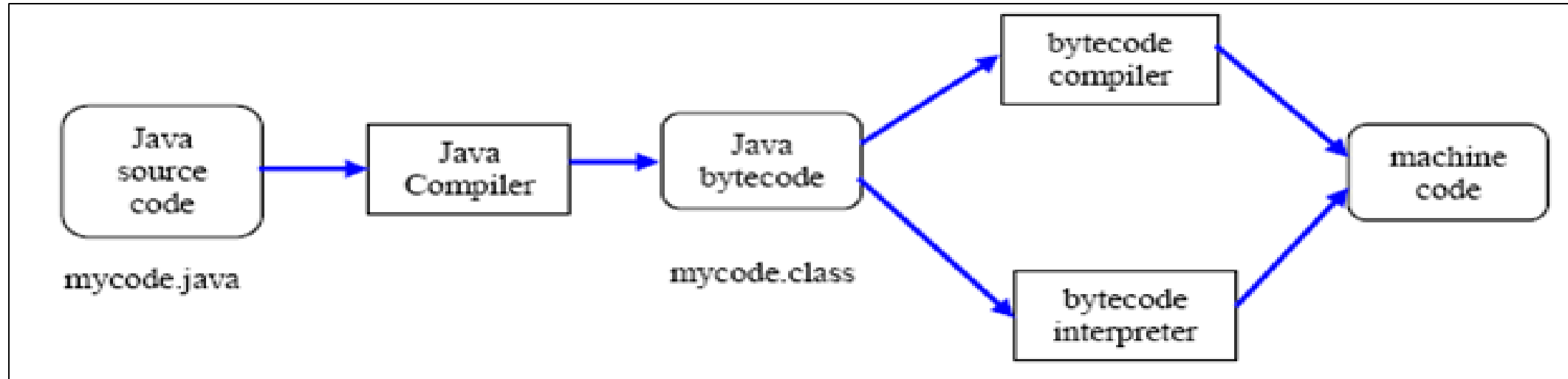
I will be using Java and Python as the example since most of you are developing in Python.

However, in no way or manner am I indicating that either Python or Java is the superior language.

It all depends on needs and use case and what problems are you tackling.

Compiled vs Interpreted Languages

Compiling Process



Compiled Languages : Source code (.java) to Byte code (.class) to Machine code

Interpreted Languages : Source code executed directly without compilation

Compiled vs Interpreted Languages

Pros & Cons

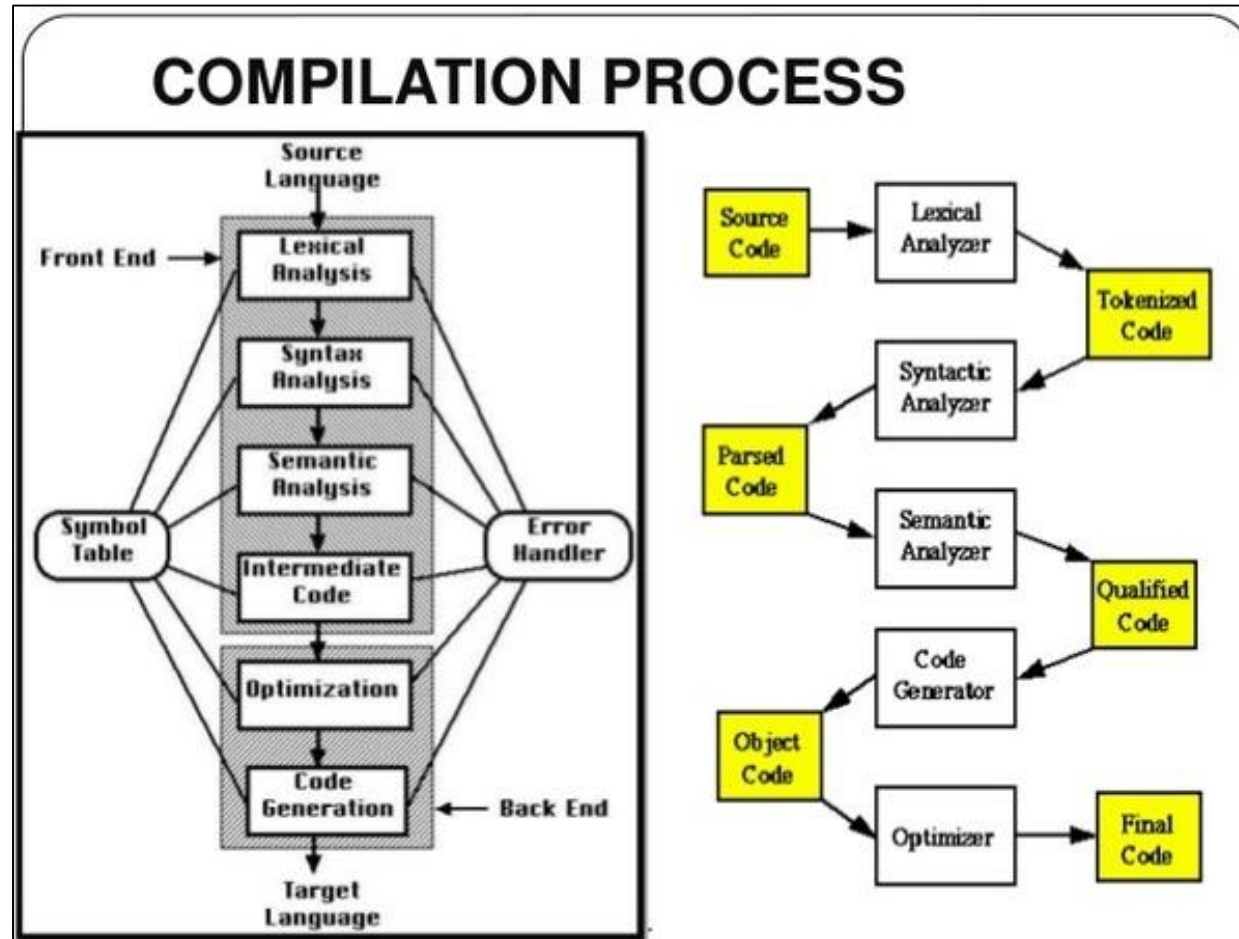


Why Fast?

Compiled		Interpreted	
PROS	CONS	PROS	CONS
ready to run	no cross platform	cross-platform	interpreter required
often faster	inflexible	simpler to test	often slower
source code is private	extra step	easier to debug	source code is public

Compiled vs Interpreted Languages

Compilation Process



Compiled vs Interpreted Languages

Compiler as Optimizer

```
public class MyClass {  
    public static void main(String args[]) {  
        for (int i=0; i<5; i++){  
            System.out.println("i = " + i);  
        }  
    }  
}
```

i = 0
i = 1
i = 2
i = 3
i = 4

Compiled vs Interpreted Languages

Compiler as Optimizer

```
1 i = 0
2 Printing_Loop:
3     condition = i >= 5
4     if condition GOTO End
5     sys.out "i = " + i
6     i = i + 1
7     GOTO Printing_Loop
8 End:
9     return
```

Some overheads:

- Same as recursion, jumping and goto will incur some overheads due to pointer arithmetic
- End of loop test after each iteration
- Reading Data from memory

Compiled vs Interpreted Languages

Loop Unrolling (Basic Example)

Optimize a program's execution speed at the expense of its binary size (space-time tradeoff)

Programs actually spend a lot of time in loops

```
1 i = 0
2 sys.out "i = " + i
3 i = i + 1
4 sys.out "i = " + i
5 i = i + 1
6 sys.out "i = " + i
7 i = i + 1
8 sys.out "i = " + i
9 i = i + 1
10 sys.out "i = " + i
11 i = i + 1
12 End:
13 return
```

i = 0
i = 1
i = 2
i = 3
i = 4

- Same output but visually more LOC
- No End of loop test after each iteration
- Imagine if i goes towards ("inf")!
 - Saving time at the expense of binary file size
 - File size grows towards ("inf") as well

Compiled vs Interpreted Languages

More complicated optimizations

- Data-flow optimizations
 - Conduct data-flow analysis based on control edges in the control graph (graph theory)
- Constant folding and propagation
 - Replace constant “ $x = 3 + 8$ ” with “ $x = 8$ ” at compile time rather than doing the calculations in run-time
- Removal of recursion
 - Converting tail recursion to iteration
- Many More different techniques :

https://en.wikipedia.org/wiki/Optimizing_compiler

Compiled vs Interpreted Languages

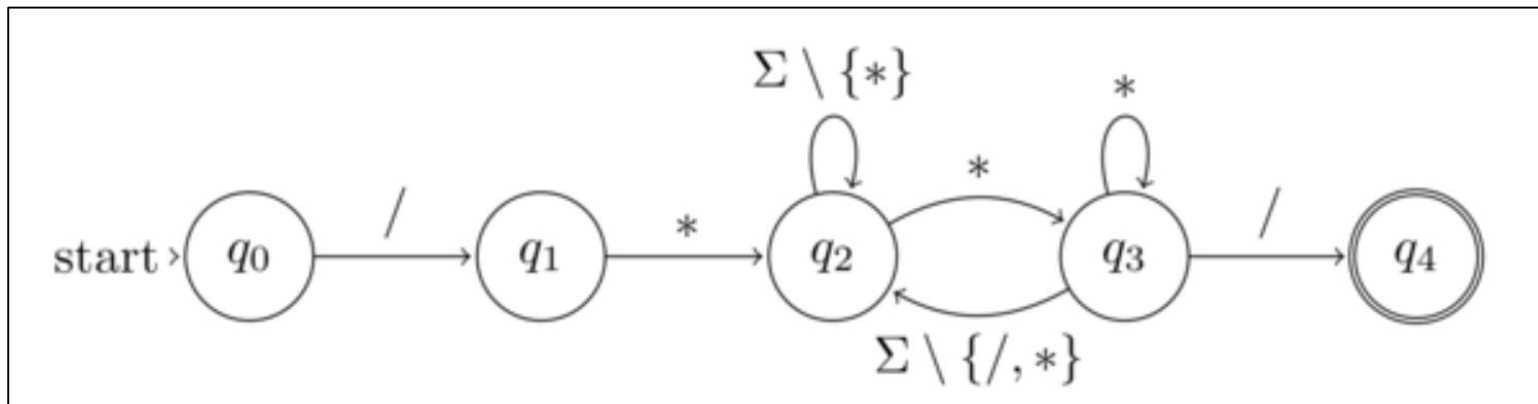
Questions?

Compiled vs Interpreted Languages

Question for you

Does using more comments leads to increase binary file size? **No**

DFA for comments in the scanning process



Continue to tokenize
non-comment codes

Communications between Activities

- In this section, we will create a basic android application with 2 activities and learn how they can communicate using Intent

Communications between Web Pages

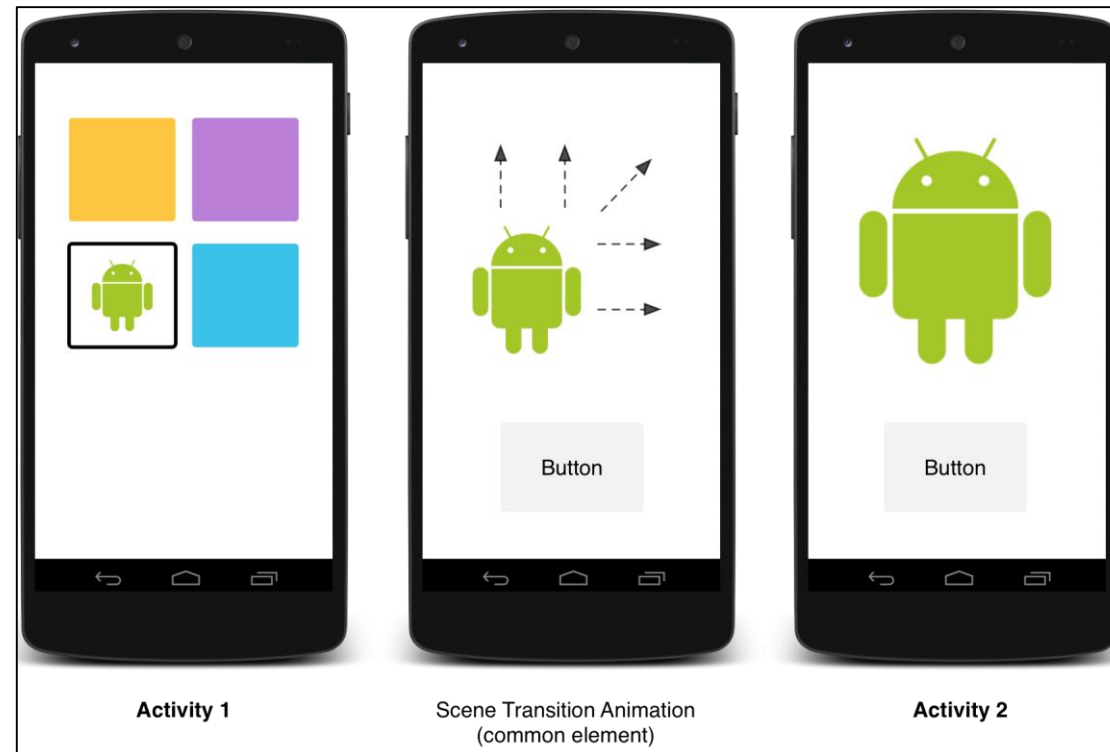
- Typically, templates is not generated on your machine and most of the contents are generated by the back end server
- Communications between pages through params given in the URL to the server, or generated by server

`http://127.0.0.1/?name=Jack`

Communications between Activities

Activities

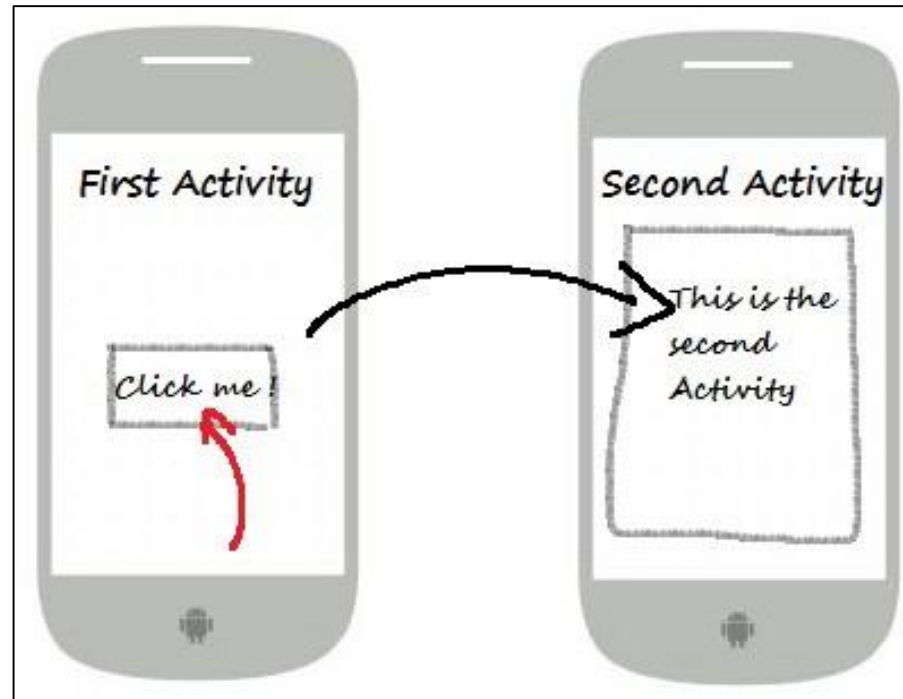
- Can see it as just a screen or User Interface.



Communications between Activities

Communications between Activities

- Achieved through Intents, a “data structure” that can be passed between activities or another application components



Communications between Activities

Live Coding – Simple communication between activities

- Create a simple GUI with a button and message box for user to type a message
- The message will be sent to the second activity(UI) to be displayed to the user

Activity Interoperability

- No Application is fun on its own without any interaction with other systems. In this section, we will create a simple activity to perform simple interoperability actions such as pulling data from an API end point :

`https://reqres.in/api/user/2`

Live Coding - Activity Interoperability

- We will first create a button to start the fetch from the end point and a textView to display the results
- Once the button is clicked, we will begin a async http get from the end point
 - AsyncTask or new Threads should be used for networking calls
 - Must use a separate thread from the UI thread to prevent the UI from freezing and app from crashing while doing the network calls
 - Private Java class is a security mechanism in Java
 - Some Stages: onPreExecute, doInBackground, onPostExecute
- After retrieving the response, we can update the textView we created

References

- <https://stackoverflow.com/questions/28209637/what-does-javac-exe-do-when-compile-a-java-file/28209778>
- <https://learntocodewith.me/programming/source-code/>
- <https://www.quora.com/What-is-role-of-compiler-during-execution-of-program>
- <https://cs.stackexchange.com/questions/396/a-dfa-for-recognizing-comments>