COMP 110/L Lecture 2

Kyle Dewey

Programming Languages as Natural Languages

Syntax

Syntax

Defines what valid sentences are

Syntax

Defines what valid sentences are

Megan goes to the store.

Syntax

Defines what valid sentences are

Megan goes to the store. 

-This is a valid sentence according to the syntactic rules of English

Syntax

Defines what valid sentences are

Megan goes to the store. 

The goes store to Megan.

Syntax

Defines what valid sentences are

Megan goes to the store. 

The goes store to Megan.

-This is not a syntactically valid sentence according to the rules of English
-Programming languages have the exact same sort of rules, though the valid sentences usually don't _look_ like natural language sentences.
-The methods for defining programming language syntax were taken directly from linguistics; the idea of a syntax error predates computers

Semantics

Semantics

Defines what valid sentences *mean*

Semantics

Defines what valid sentences *mean*

Megan goes to the store.

Semantics

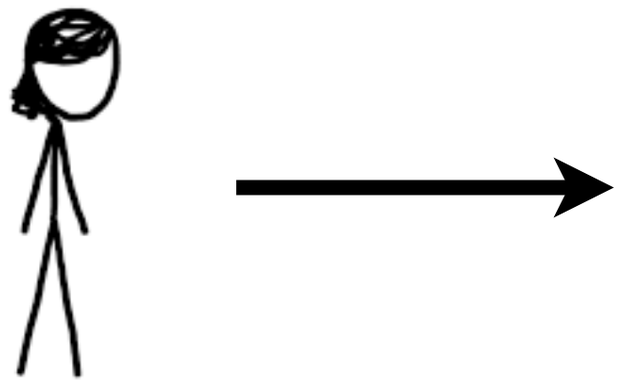
Defines what valid sentences *mean*

Megan goes to the store.

Semantics

Defines what valid sentences *mean*

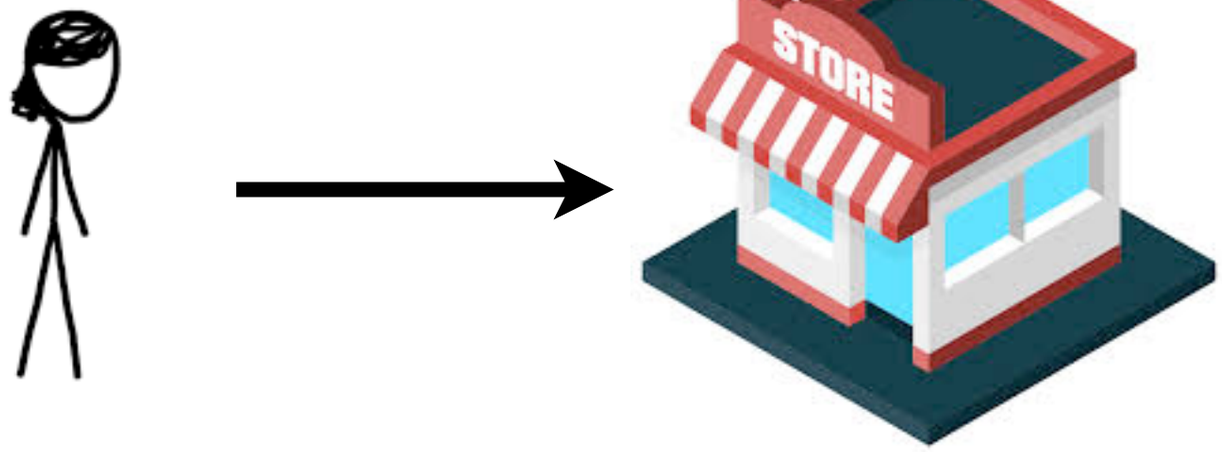
Megan goes to the store.



Semantics

Defines what valid sentences *mean*

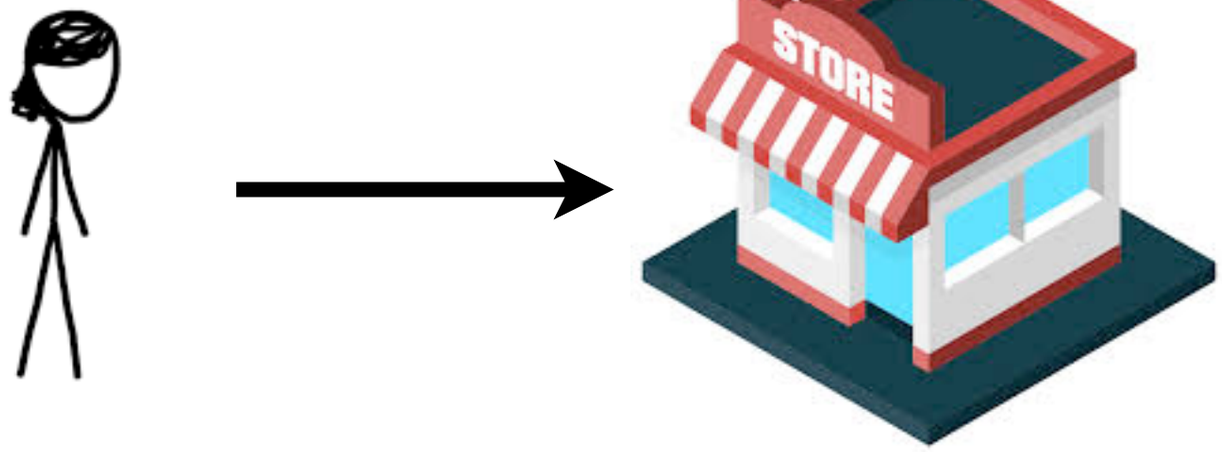
Megan goes to the store.



Semantics

Defines what valid sentences *mean*

Megan goes to the store.

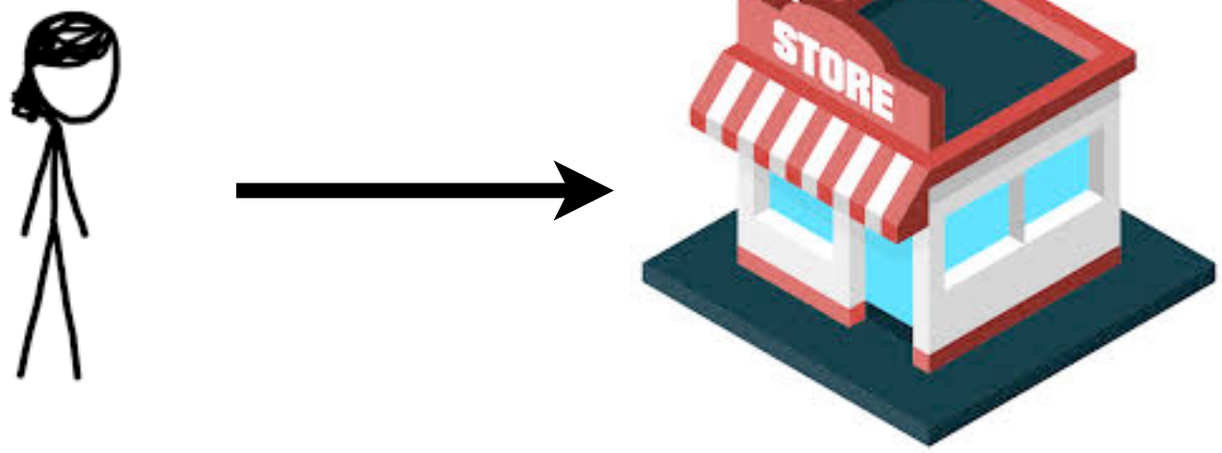


Colorless green ideas sleep furiously.

Semantics

Defines what valid sentences *mean*

Megan goes to the store.



Colorless green ideas sleep furiously.

???

-Natural languages allow us to define sentences that are syntactically valid but semantically nonsensical

Programming Language Semantics

- Some languages have the same problem!
- All syntactically valid sentences in Java have prescribed meaning
  - ...this meaning might not be useful...
  - ...and it definitely could be unintended...

-C/C++ lets you define equally meaningless statements (thanks to undefined behavior)

Learning a Language

Learning a Language

Have to start somewhere

Learning a Language

Have to start somewhere

Megan goes to the store.

Learning a Language

Have to start somewhere

Megan goes to the store.

Need nouns
(that which can act
or can be acted on)

Learning a Language

Have to start somewhere

Megan goes to the store.

Need nouns
(that which can act
or can be acted on)

Need verbs
(the actions)

Learning a Language

Have to start somewhere

Megan goes to the store.

Need nouns
(that which can act
or can be acted on)

Need verbs
(the actions)

Need connections between the two

-This will work

The Point

- To make complete sentences, we need a lot of stuff
- Java requires a *lot* for a complete sentence

The Point

- To make complete sentences, we need a lot of stuff
- Java requires a *lot* for a complete sentence

```
public class MyClass {
   public static void
   main(String[] args) {
      ...
   }
}
```

-For now, the ... is everything you actually want to write, everything else is required but somewhat indirect

The Point

- To make complete sentences, we need a lot of stuff
- Java requires a *lot* for a complete sentence

```
public class MyClass {
   public static void
   main(String[] args) {
      ...
   }
}
```

For now, a magical incantation

-For now, the ... is everything you actually want to write, everything else is required but somewhat indirect

Let's see some code!
(in jGrasp)

Example:

`HelloWorld.java`

Java Coding Process

Text Editor

Text Editor

`emacs, vi, pico, nano...`

Text Editor

`emacs, vi, pico, nano...` 

Microsoft Word, Google Docs

Text Editor

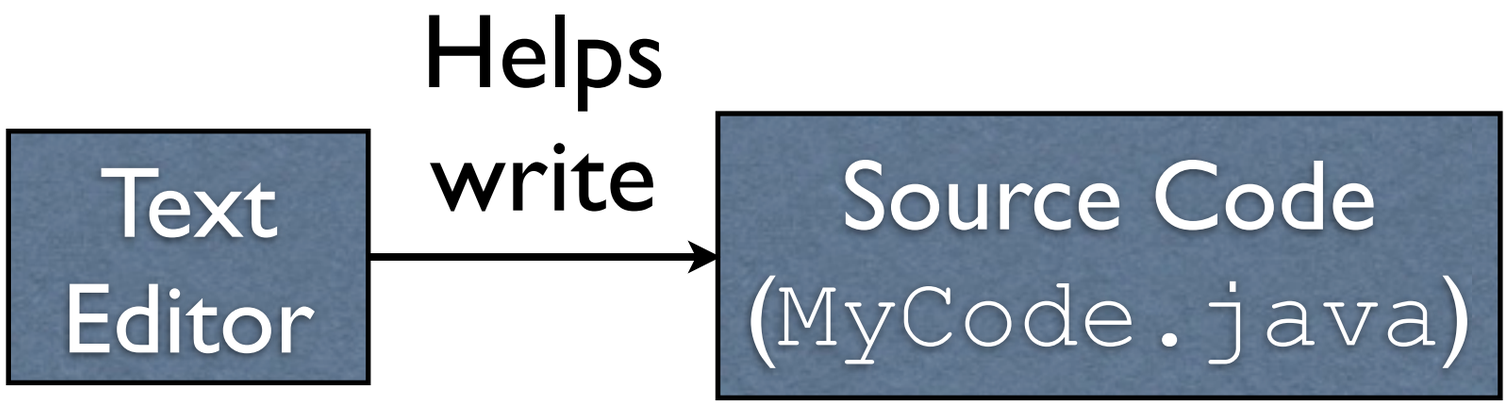
`emacs, vi, pico, nano...` 

~~Microsoft Word, Google Docs~~

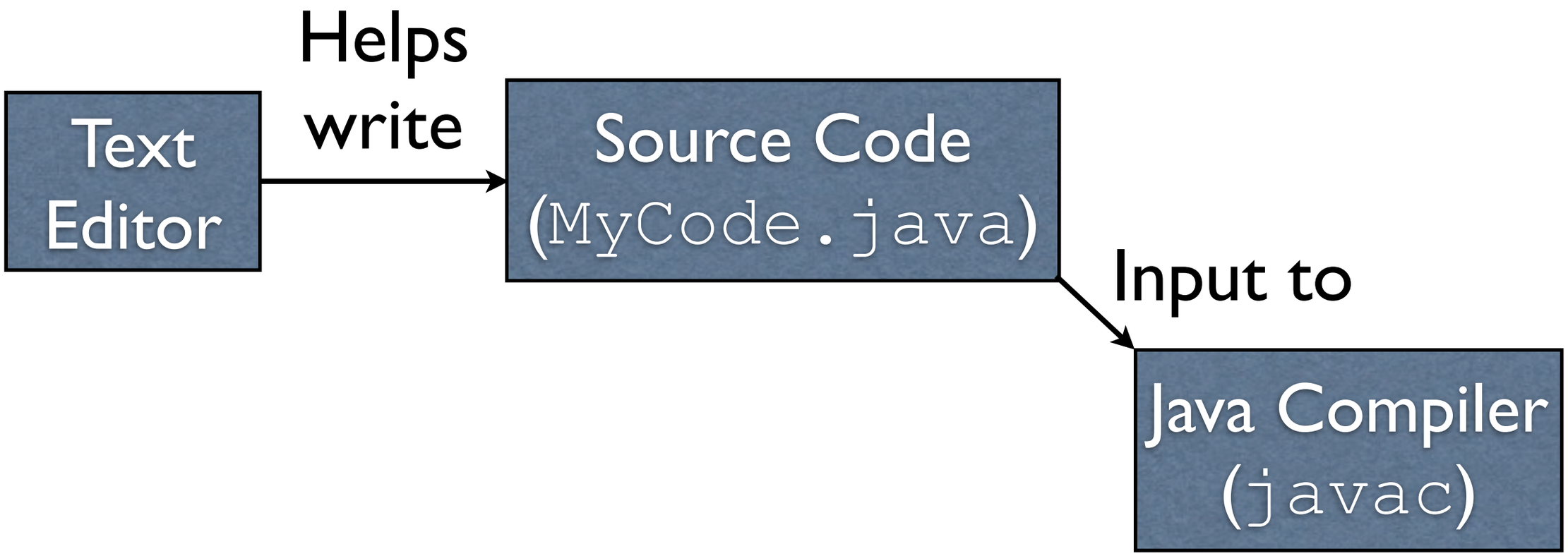
Wanted: plain text without formatting

Also nice: syntax highlighting

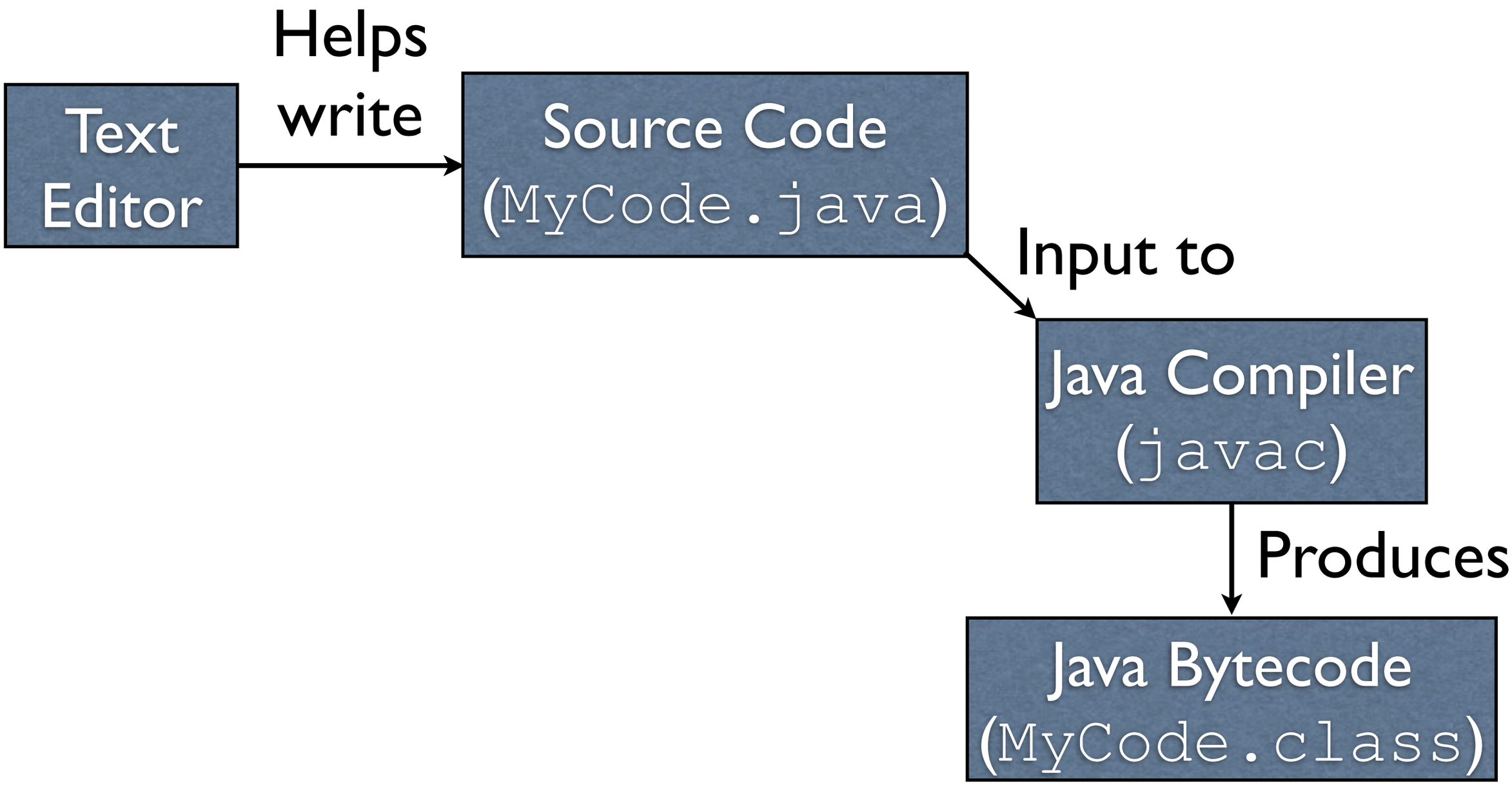
-The text editor is for writing text that is meant for both humans and the computer
-Formatted text is really only for humans



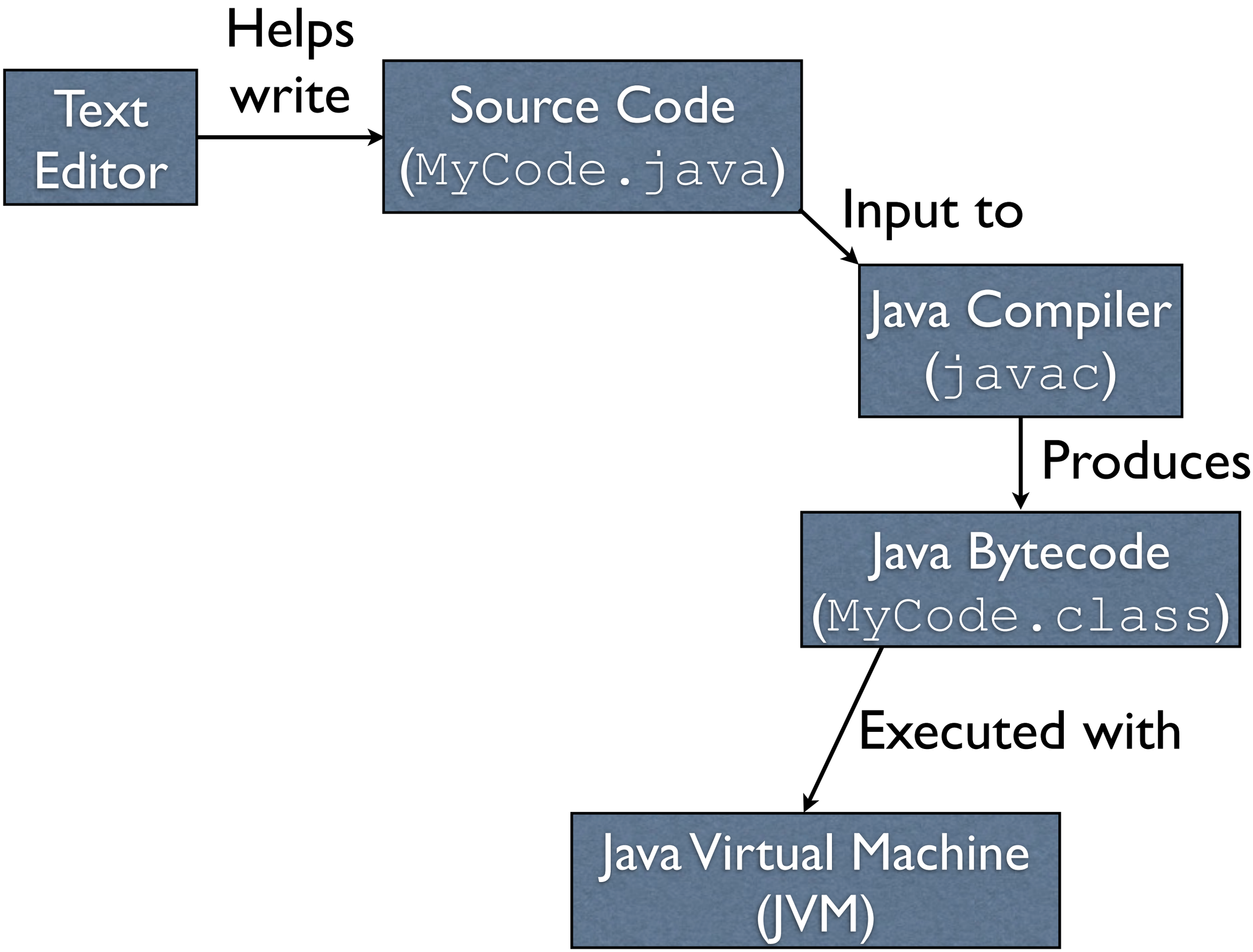
-The source code is usually the thing you see people hacking away at in movies
-The source code is in a format that is intended to be read by both humans and computers



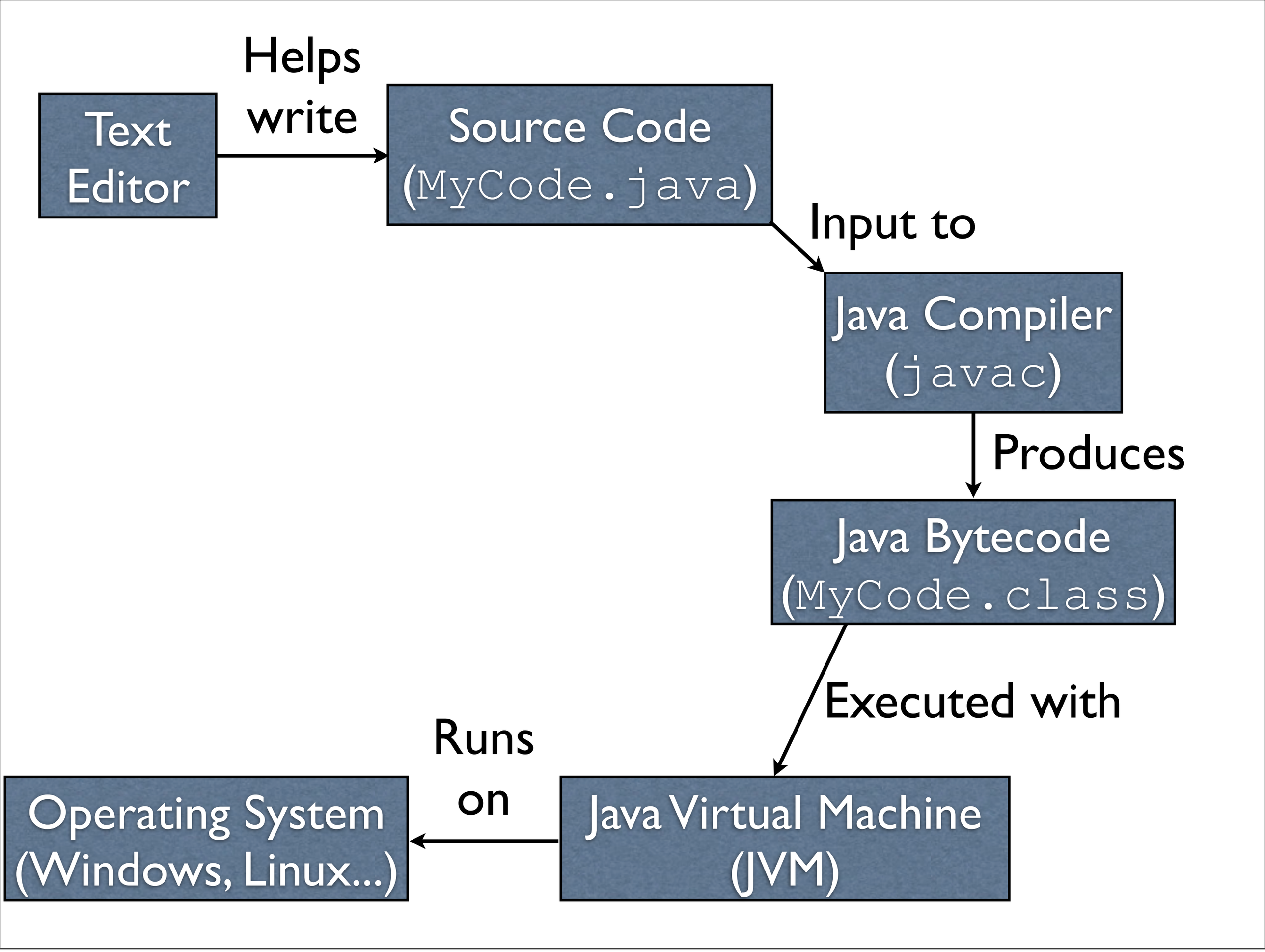
-The Java compiler takes the source code as input
-The compiler reads through the source code. If there are any surface-level problems with the source code, the compiler will reject it. If the code is free of surface-level problems...



-...then the Java compiler will translate the source code into Java bytecode.
-Java bytecode is a representation of the source code that is intended for the machine. While we could technically write directly in bytecode, bytecode is not exactly straightforward, and it wasn't designed to be directly used by humans.



-The JVM will read in the Java bytecode and do whatever the bytecode tells the JVM to do.
-The Java compiler made sure that the Java bytecode produced was an accurate translation of the Java source code you wrote, so ultimately the JVM is doing exactly what you told it to do



-The JVM itself is just a program that runs on whatever operating system you're running. That is, your Java program itself is running on a program (namely the JVM)