

“Jokes”

Invent Your Own Computer Games with Python

Heejin Park

*College of Information and
Communications
Hanyang University*



Introduction

■ How Programs Run on Computers

■ “Jokes”

- Sample Run
- Source Code

■ Code Explanation

■ Things Covered In This Chapter

How Programs Run on Computers

■ How Programs Run on Computers

- **Operating System (OS)**
 - Ex) Windows, MacOS, Linux or another one.
 - Program that runs other programs called **applications**.
- **Hardware**
 - Parts of the computer that you **can touch**
 - Ex) the monitor, or the keyboard and mouse, or a printer
- **Software**
 - programs like the OS or applications or games that run on the computer.

How Programs Run on Computers

■ How Programs Run on Computers

- **Machine Code**

- very basic instructions.
- simple enough for computer's main microchip to understand.
 - » **CPU or Central Processing Unit**
- written in ones and zeros.
 - » 10101101 00110000 11000000
- These instructions aren't very easy for humans to work with.

How Programs Run on Computers

■ How Programs Run on Computers

- **Assembly language**
 - Ex) MOV, JMP, PUSH or XOR
 - makes reading and writing the instructions easier.
 - but still **long** and **complicated**.
- This is where **higher-level programming languages** come in.

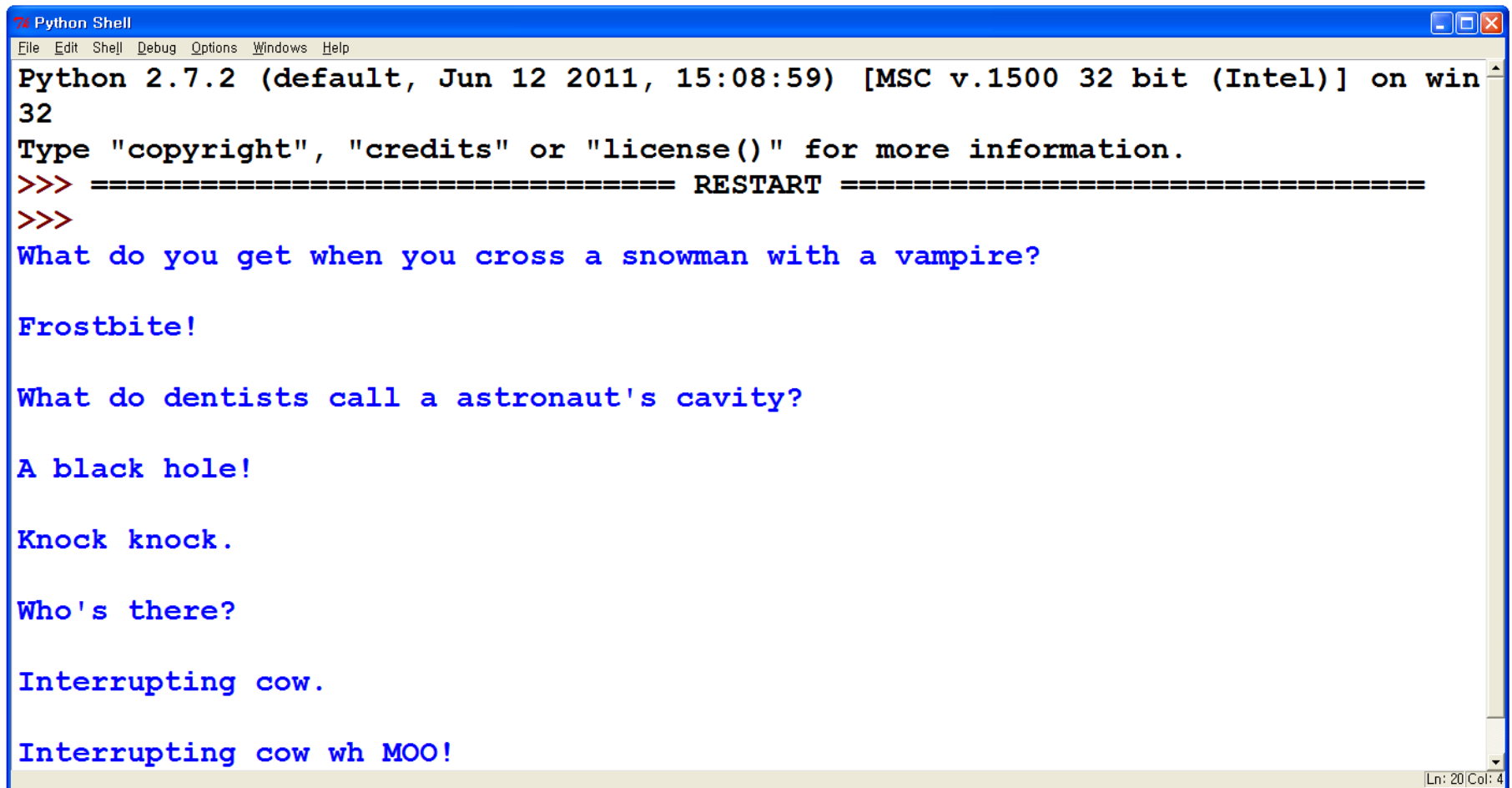
How Programs Run on Computers

■ How Programs Run on Computers

- **High-level languages**
 - Ex) Python, Java, C++, Pascal, Perl, Basic, and many others.
 - take care of many of the details of machine code.
- **Interpreter**
 - translates high-level languages into machine code.

“Jokes”

■ Sample Run

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following output:

```
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win
32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
What do you get when you cross a snowman with a vampire?

Frostbite!

What do dentists call a astronaut's cavity?

A black hole!

Knock knock.

Who's there?

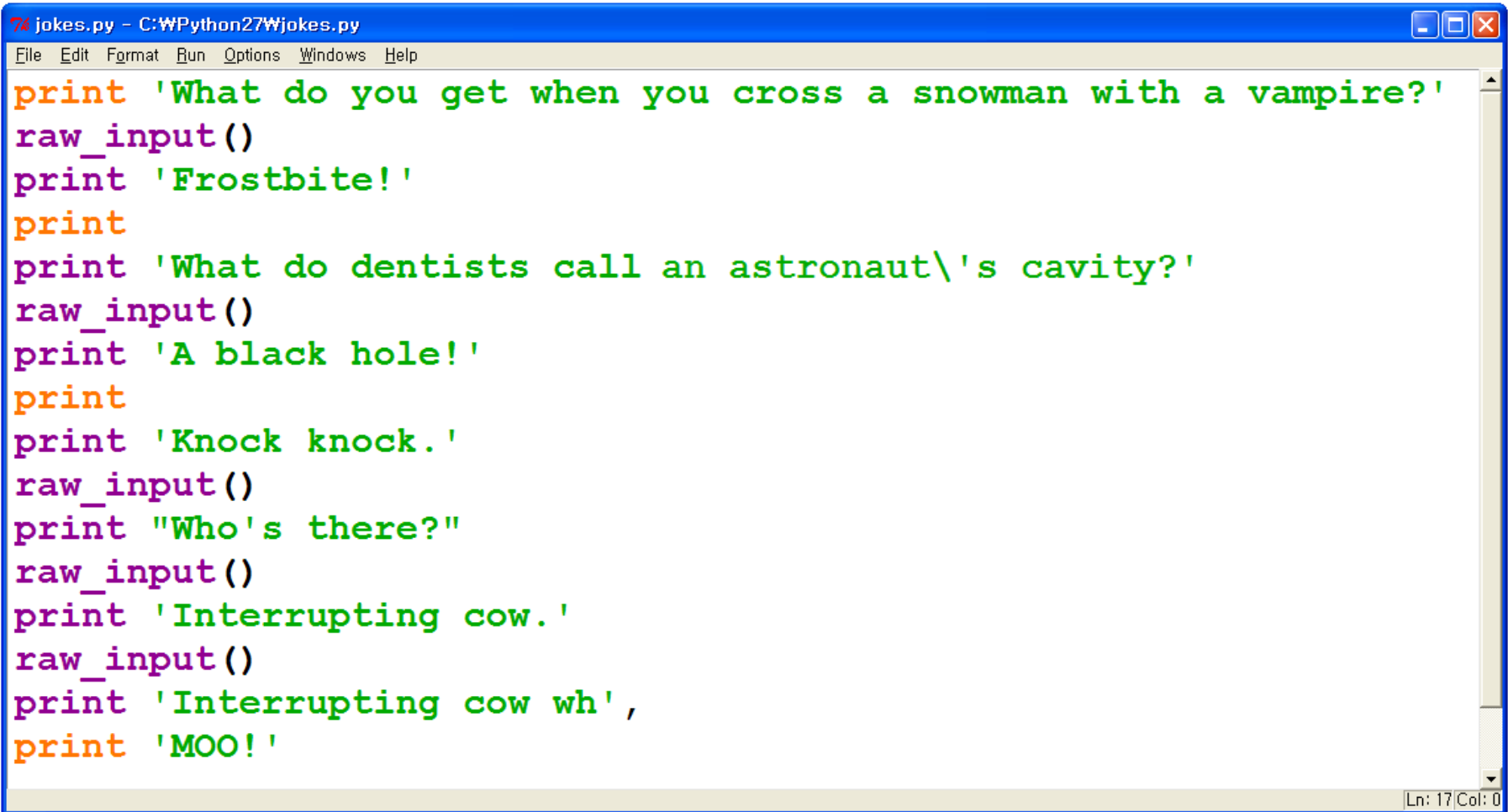
Interrupting cow.

Interrupting cow wh MOO!
```

The status bar at the bottom right indicates "Ln: 20 Col: 4".

“Jokes”

■ Source Code

A screenshot of a Python IDE window titled 'jokes.py - C:\WP\Python27\Wjokes.py'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Windows', and 'Help'. The main text area contains a Python script with three jokes. The first joke asks 'What do you get when you cross a snowman with a vampire?' and answers 'Frostbite!'. The second joke asks 'What do dentists call an astronaut\'s cavity?' and answers 'A black hole!'. The third joke is a knock-knock joke about an interrupting cow. The code uses color-coding: 'print' is orange, strings are green, and 'raw_input()' is purple. The status bar at the bottom right shows 'Ln: 17 Col: 0'.

```
jokes.py - C:\WP\Python27\Wjokes.py
File Edit Format Run Options Windows Help

print 'What do you get when you cross a snowman with a vampire?'
raw_input()
print 'Frostbite!'
print
print 'What do dentists call an astronaut\'s cavity?'
raw_input()
print 'A black hole!'
print
print 'Knock knock.'
raw_input()
print "Who's there?"
raw_input()
print 'Interrupting cow.'
raw_input()
print 'Interrupting cow wh',
print 'MOO!'
```

Ln: 17 Col: 0

Code Explanation

■ Three `print` statements.

```
print 'What do you get when you cross a snowman with a vampire?'  
raw_input()  
print 'Frostbite!'  
print
```

- Read the first line, press Enter, and then read the punch line.
- The user can still type in a string and hit Enter
 - because we aren't storing this string in any variable.
- The last `print` statement call has no string.

Code Explanation

■ Escape Characters

```
print 'What do dentists call an astronaut\'s cavity?'  
raw_input()  
print 'A black hole!'  
print
```

- a **slash** right before the single quote (that is, the apostrophe).
 - \ is a backslash, / is a forward slash.
 - tells us that the letter right after it is an escape character.
 - escape character helps us print out letters.

Code Explanation

■ Some Other Escape Characters

- What if you really want to display a backslash?
- This line of code would not work.

```
>>> print 'He flew away in a green\teal helicopter.'  
He flew away in a green eal helicopter.
```

Code Explanation



■ Quiz

- Instead, try this line

```
>>> print 'He flew away in a green\\teal helicopter.'
```

Code Explanation

■ Escape Characters

Escape Character	What Is Actually Printed
<code>\\</code>	Backslash (\)
<code>\'</code>	Single quote (')
<code>\"</code>	Double quote (")
<code>\n</code>	Newline
<code>\t</code>	Tab

Code Explanation

■ Quotes and Double Quotes

- Strings don't always have to be in between single quotes.
- You can also put them in between **double quotes**.

```
>>> print 'Hello world'
Hello world
>>> print "Hello world"
Hello world
```

Code Explanation



■ Quiz

```
>>> print 'Hello world'
```

Code Explanation

■ Quotes and Double Quotes

- `\'` to have a single quote in a string surrounded by **single quotes**.
- `\''` to have a double quote in a string surrounded by **double quotes**.

```
>>> print 'I asked to borrow Abe\'s car for a week. He said, "Sure."'
I asked to borrow Abe's car for a week. He said, "Sure."
>>> print "He said, \"I can't believe you let him borrow your car.\""
He said, "I can't believe you let him borrow your car."
```

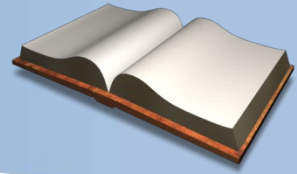

Code Explanation

■ Using Commas

```
print 'Interrupting cow wh',  
print 'MOO!'
```

- **Comma** means we do not want to `print` a newline at the end.
- This is why 'MOO!' appears next to the previous line.

Things Covered In This Chapter



- Using `print` with no parameters to display blank lines.
- Escape characters.
- Using single quotes and double quotes for strings.
- Using commas at the end of print statements.