Introduction to Python

IDLE and Python 2.7

Lecture notes of Alexander Wood
awood@citytech.cuny.edu

New York City College of Technology

What is a program?

A **program** consists of instructions that specify how to perform some task. There are many types of tasks a program can perform:

- Mathematical computation
  - Solving systems of equations
- Symbolic computation
  - Finding all instances of a word in a text document
- Graphical computation
  - Image recognition
  - Playing videos online

Input and Output

Many programs take **input** and/or yield some **output**.

- **Input:** Get data from the user, from a file, etc.
- **Output:** Print data, save data in a file, etc.

Program Flow

We utilize various code structures to control which lines of code execute.

- **Conditional execution:** Evaluate certain code blocks under certain conditions. For instance, `if` and `else` statements.
- **Repetition Structures:** Repeat some action.

Running Python

- You can run Python from your browser
  - PythonAnywhere
- The Python interpreter
  - Accessed by typing `python` or `python3` from command line
- IDLE, an integrated development environment

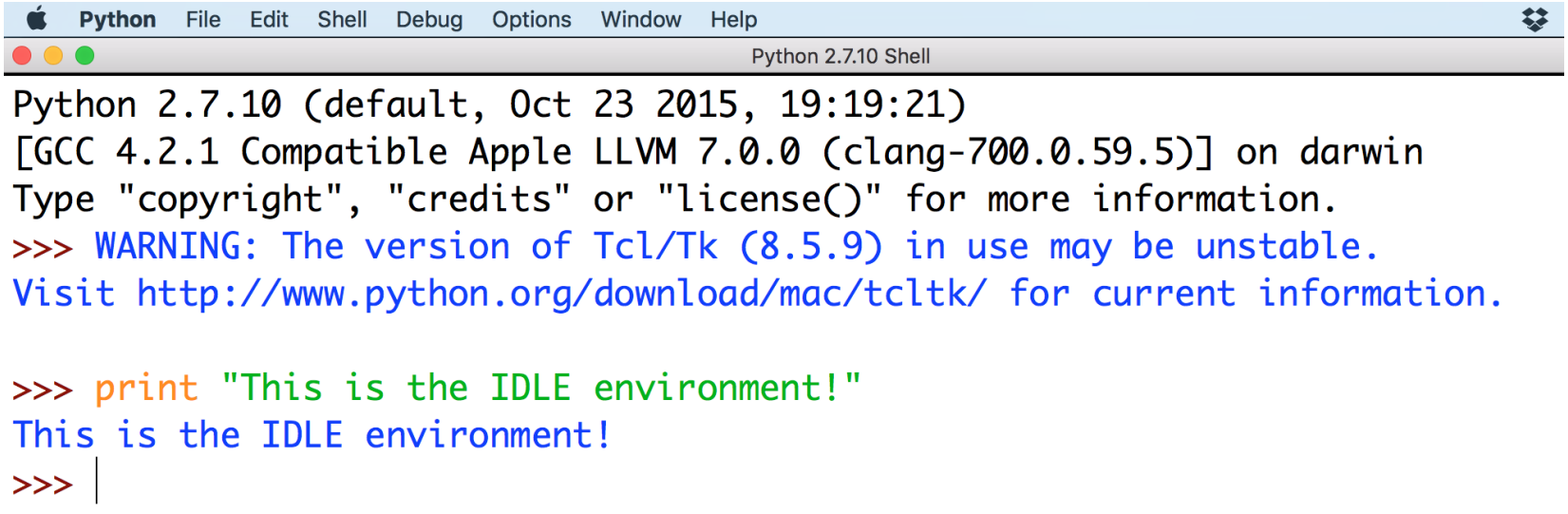
IDLE

IDLE combines:

- Interactive Python shell
- Color-coding text editor
- “Check module" for syntax errors
- Search tools
- Text formatting for auto-indentation
- Debugger tool

Opening IDLE

- Mac: Go to applications, select the Python folder, and double-click “IDLE"
- PC: Select Python from Start menu, select “IDLE"



Using IDLE

The >>> prompt indicates that you may write a Python statement. When you press `Enter`, the statement is executed.

```
>>> name = 'Rihanna'
>>> song = 'Work'
>>> print name + ' is the singer of the song ' + song
Rihanna is the singer of the song Work
>>>
```

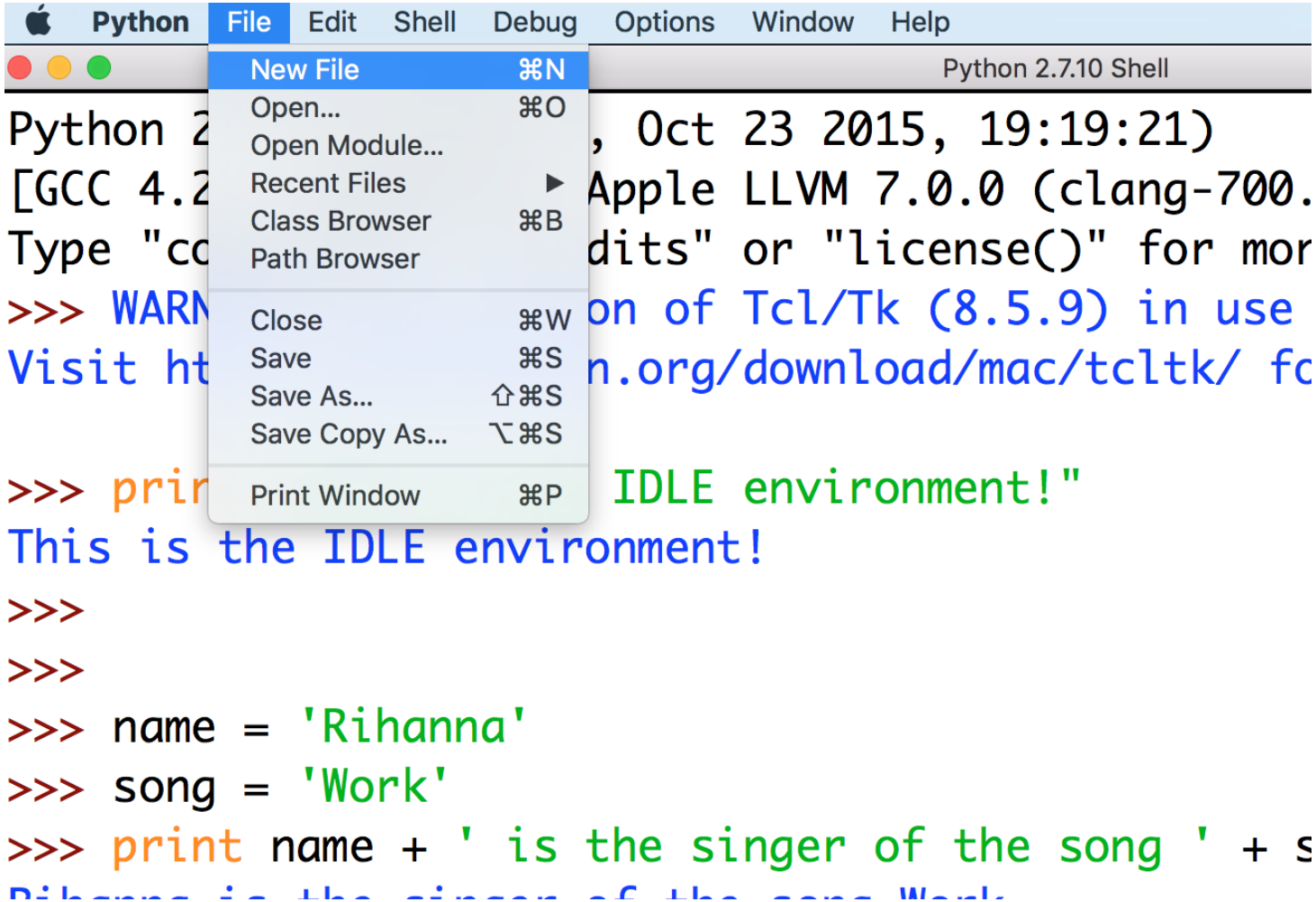
Using IDLE: Multiline statements

Multiline statements in IDLE:

```
>>> for x in range(6):
        print x


0
1
2
3
4
5
>>> quote = """To be, or not to be
That is the question."""
>>>
```

Writing a Python program using IDLE

Open a new editing window



Example Program in IDLE

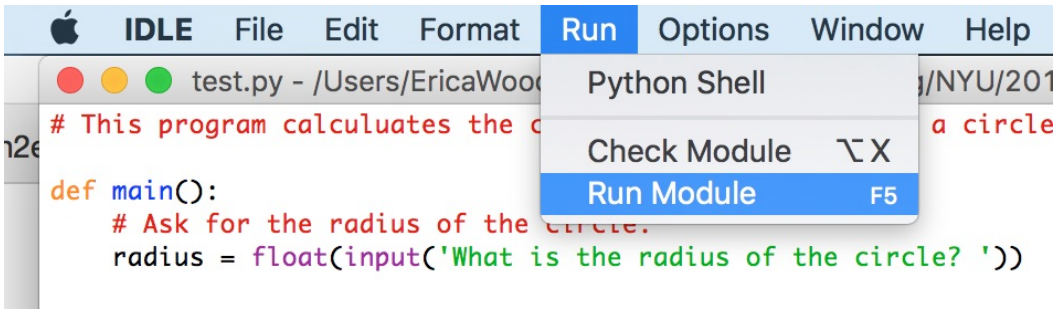
```
# This is an example of a program

def hotline_bling():
    lyrics = "You used to call me on my cell phone\n"

    for x in range(3):
        print lyrics

hotline_bling()
```

Color Coding in IDLE

- **Blue:** Defined names, such as functions and classes
- **Orange:** Python keywords
- **Red:** comments
- **Green:** String literals
- **Purple:** Built-in functions
- **Black:** Everything else!

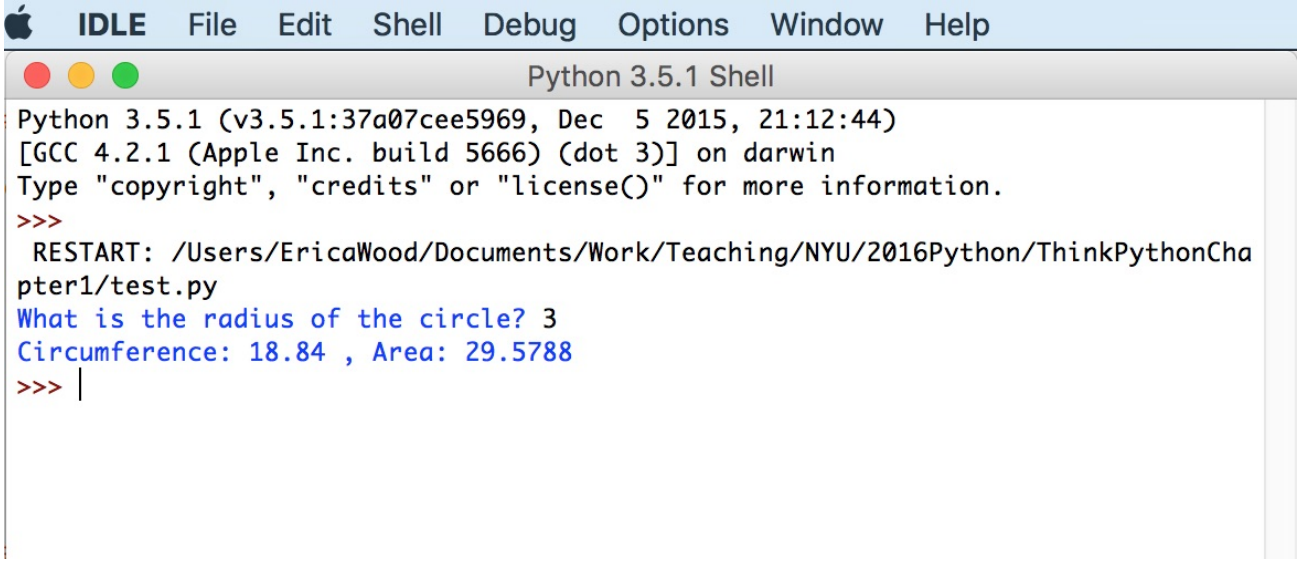
Running the test program in IDLE

- First, save the program!
- Second, press F5 or Run > Run Module



Running the test program in IDLE

The program then runs in IDLE's Python Shell.



Two Modes of Python

- **Interactive mode** is using python interactively, by typing directly into the python interpreter. Open up IDLE and type away!
- **Python Programming**, is programming in the python language, where .py scripts are run.

Expressions: Values and Types

- **Value:** The basic building blocks of a program! For instance, a letter, or a number.
  - Examples: `42`, `3.14`, `'hotline bling'`
- **Type:** Every value belongs to a specific type. For instance,
  - `42` is an `int`, `3.14` is a `float`, and `'hotline bling'` is a `string`.

Data Types

A few data types available in Python are:

- **Booleans**, which take value `True` or `False`
- **Integers**, whole numbers
- **Floats**, numbers with decimals
- **Strings**, sequences of text characters

Determining Types

Determine the type of a value by calling the built in function `type()`

```
>>> type(42)
<class 'int'>
>>> type('the answer to life, the universe, and everything')
<class 'str'>
>>> type(True)
<class 'bool'>
>>> type(true)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    type(true)
NameError: name 'true' is not defined
>>> type('True')
<class 'str'>
>>> type(3.14)
<class 'float'>
>>>
```

Debugging

When you make a mistake in your code, you will receive an error. These errors are called **bug**. Getting rid of the bugs in your code is called **debugging**. Sometimes the mistake is as simple as a misspelled word, whereas other times the error is in the code structure itself. Tracking down bugs can be time-consuming and stressful, and it is important to learn various debugging skills. We will discuss various techniques throughout the course.

Python 2.x VS Python 3.x

- Python 3 is backwards incompatible with Python 2 and introduces new features.
- `print` is a function in Python 3 but a statement in python 2
- For more information on the differences see `https://docs.python.org/3.1/whatsnew/3.0.html`

References

- How to think like a computer scientist: Learning with Python, chapter 1 `http://www.openbookproject.net/thinkcs/python/english2e/ch01.html`
- Official Python 2 documentation `https://docs.python.org/2/`