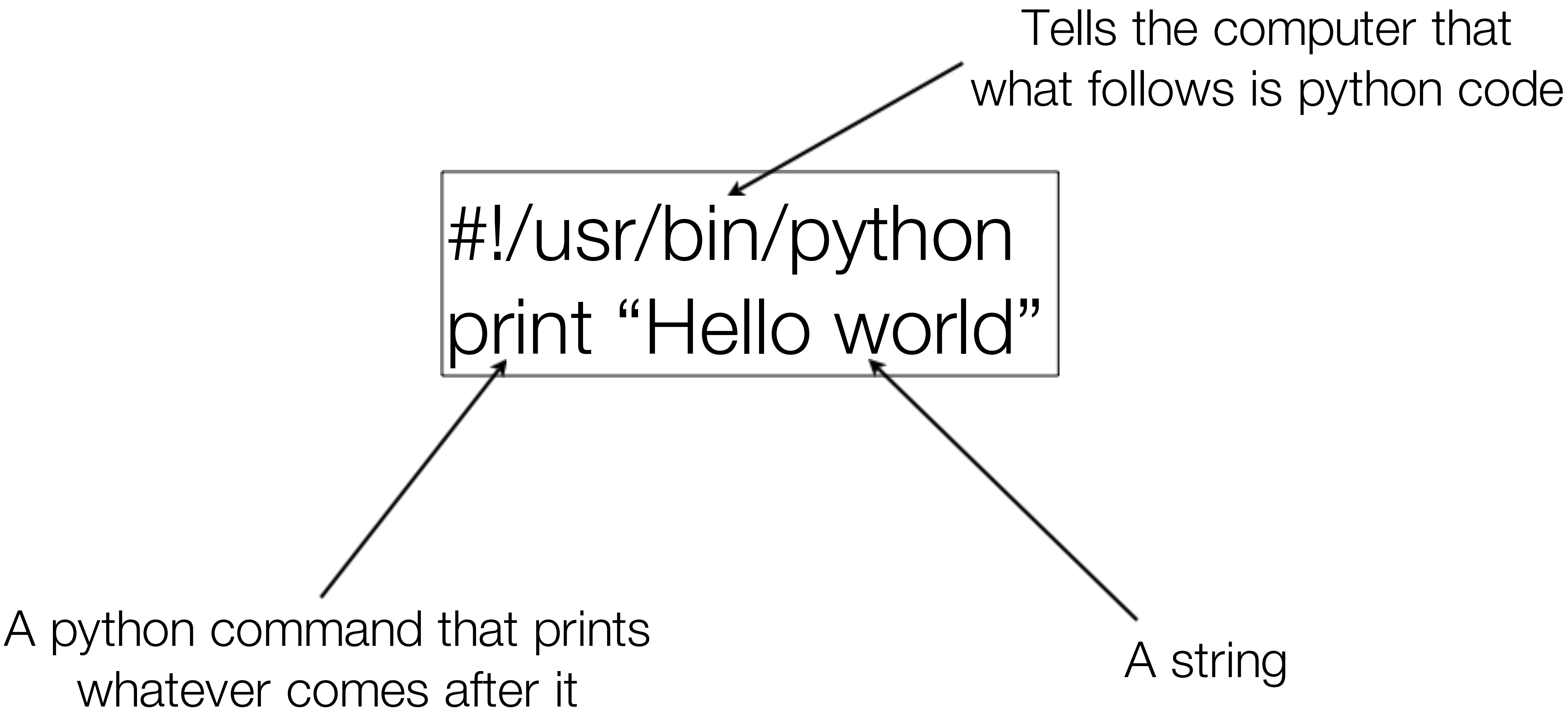
Modules, Functions, Comparators, and Lists

Lecture 8 - COMP 364
February 1, 2010
Derek Ruths

Our first python program

In an executable file:



Tells the computer that what follows is python code

```
#!/usr/bin/python
print “Hello world”
```

A python command that prints whatever comes after it

A string

Modules

- Python keeps things organized in different places called “modules”
- You tell python that you want to use things in a given place by *importing* a module. Some important & useful modules
    - *sys*: system-specific information like command arguments
        - *sys.argv* are the command arguments given to your script
    - *os*: operating-system specific things
        - *os.name* is the name of your operating system

    - To bring a module into your python code, type “*import <module name>*”

```
import sys
print sys.argv
```

Functions

- Functions perform an action and, sometimes, return a result value. Python provides **many** functions, but we will also learn how to define our own (later).

- Functions have the format:

    - <function name>(<argument1>,<argument2>,...,<argument N>)

```
import os
print os.getlogin()
```

```
import os
os.chdir(‘..’)
print os.getcwd()
```

Lists: a container

- Contain multiple objects (in order)
  - *last_names = [‘Smith’, ‘Singer’, ‘Smith’]*
- Can contain objects of different types
  - *x = [‘hello’, 2, True, 2.5]*
- Getting contents out by index: *x[0]*

- Question: *sys.argv* is a list. How would you print out the first argument given to your program?

Comparators

- Equality: x == y
  - Is True if x and y are equal, False otherwise
- Inequality: x != y
  - Is True if x and y are not equal, False otherwise
- Some other operators:
  - x > y
  - x >= y
  - x < y
  - x <= y

Comparator practice

- What are the results of these different comparisons?
  - *2 == 3*
  - *‘hello’ == “hello”*
  - *‘x’ < ‘y’*
  - *2 < 2.2*
  - *2 == 2.0*
  - *True != False*

Exercise: write a program that tests whether its first and second arguments are equal

Files in Python

Lecture 11 - COMP 364
February 8, 2010
Derek Ruths

File operations

- Reading content
- Writing content

- *The file workflow*
  - *Open the file*
  - *Read/write content*
  - *Close the file*

Reading Files - Technique 1

```
fh = open(<path to file>, ‘r’)

content = fh.readlines()

fh.close()

for line in content:

<do stuff with line>
```

Exercise: Print lines that are less than 10 characters in length.

Reading Files - Technique 2

```
fh = open(<path>, ‘r’)

for line in fh:

<do stuff with the line>

fh.close()
```

Exercise: Print lines that are less than 10 characters in length.

This is the better technique for handling large files.

Writing content

```
fh = open(“/home/druths/blah.txt”, ‘w’)

fh.write(‘hello world!’)

fh.close()
```

Exercise: Write command line arguments that are less than 10 in value

Exercise: Write all lines in ***source.txt*** that have less than 10 characters to file ***output.txt***