Python with Rasberry Pi

What is a Raspberry Pi ?

- A Raspberry Pi is a very inexpensive, fully programmable computer that is small enough to fit into the palm of your hand.
- The Raspberry Pi is small in size but mighty in potential. It can be used as a regular desktop computer or you can create super-cool projects with it, like games.

Raspberry Pi History

- The Raspberry Pi is still a fairly young device. It was created in the United Kingdom by Eben Upton and a few colleagues.
- The first commercial version, Model A, was officially offered for sale in early 2012 at the low price of $25.
- There are other different names given to the Raspberry Pi such as Rpi and just Pi.
- More on Raspberry Pi can be found on the Raspberry Pi foundation's website

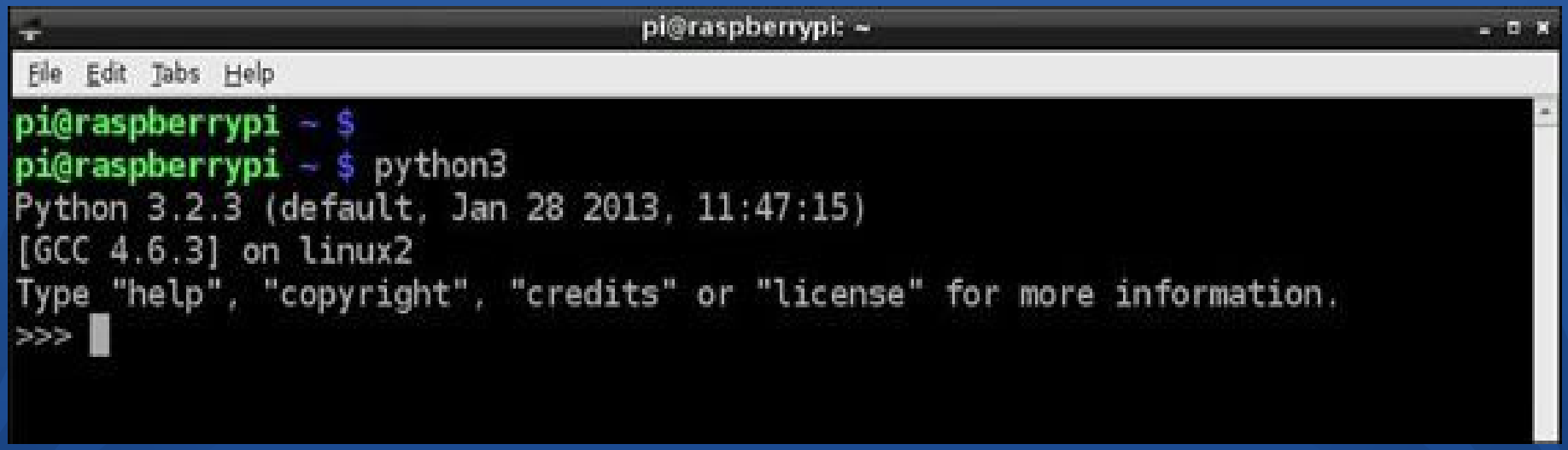
  Raspberry Pi website

Why Learn to program Python on a Raspberry Pi ?

- Python allows a Raspberry Pi owner to increase the field of project possibilities to an incredible size.
- The Raspberry Pi offers an incredibly cheap development platform for Python programming. Though Python can be considered “educational” because it is easy to learn, by no means is Python wimpy.
- You can write games in Python and run them on gaming consoles controlled by your Raspberry Pi.
- Armed with Python and Pi, your only limit is your imagination.

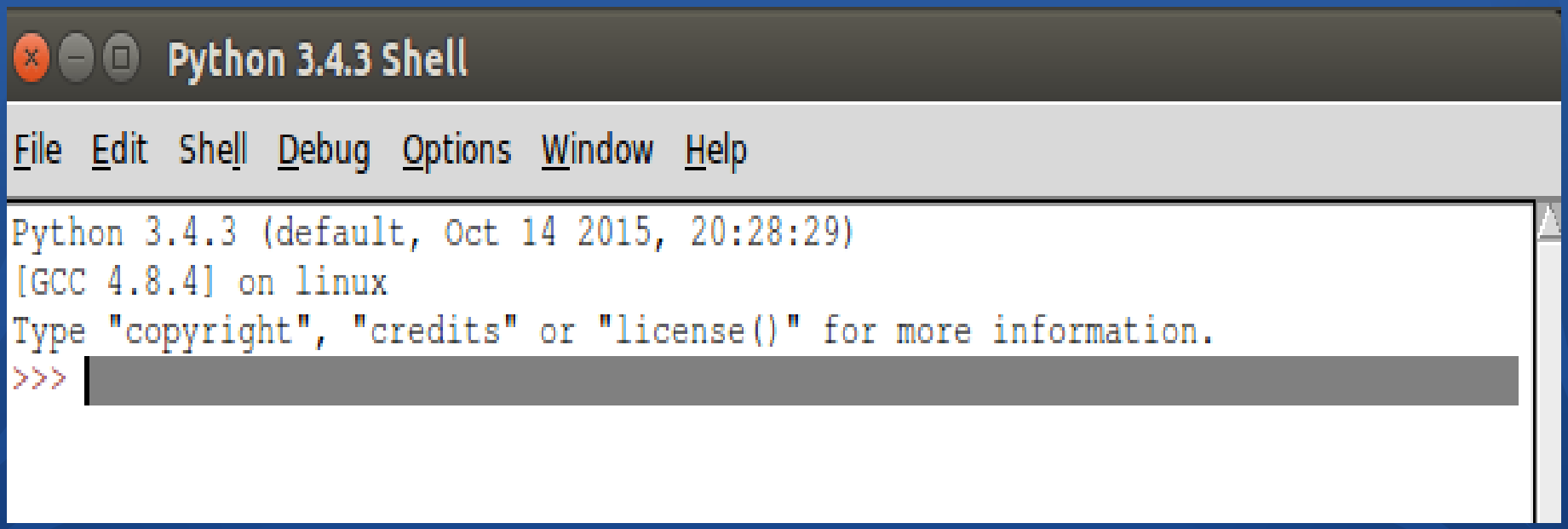
Python Interactive Shell

- To enter the interactive Python shell, type in the command “python3” (for python version3) and “python2” (for python version2) at the command line and press enter.
- You simply enter a Python statement and press Enter. The Python interpreter checks the statement’s syntax. If the syntax is correct, the statement is translated into binary code and executed.



Python Development Environment Shell (IDLE)

- IDLE stands for Interactive DeveLopment Environment.
- This development environment provides a built-in text editor and many features that assist in the creation and testing of Python scripts.
- To start up IDLE you just double-click the IDLE 3 icon on the desktop. You can also find it under the LXDE Programs Menu icon. The figure below shows the IDLE shell for Python v3.



Creating Python Scripts

- You can create whole files of Python statements and then run them. These whole files of Python statements are called Python scripts.
- Python scripts can be run from either the Python interactive shell of from IDLE.
- Python scripts always end with .py, like hello.py
- Use any Text Editor to create a python script
- This is how you run a python script :

  python3 script.py

  where script.py is the file containing your python code

```
Python 3.4.3 (default, Oct 14 2015, 20:28:29)
[GCC 4.8.4] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> python3 script.py
```

print()

- print() is used to print anything on the screen.
- print() can print numbers and strings.

```
1
2 #printing a String
3
4 print("Python is great!")
5
6 #printing a number
7
8 print( 5 )
9
```

print()

```python
#using single Quotes inside a print statement

print("It's raining!")

#using double quotes inside a print statement

print(""" "Python" is a programming language! """)

#Concatenating two strings

print("Hello " + "world")

# Using an Escape Sequence to Add a Linefeed

print("This is line one.\nThis is line two\nAnd this is line three")
```

Escape sequences

- \' → Displays a single quote
- \” → Displays a double quote in output
- \\ → Displays a single backslash in output
- \a → produces a “bell” sound with output
- \f → Inserts a formfeed into the output
- \n → Inserts a linefeed into the output
- \t → Inserts a horizontal tab into the output

Comments

- A comment's purpose is to provide understanding of the script's syntax and logic.
- To add a comment to a script, you precede it with the pound or hash symbol (#). The Python interpreter ignores anything that follows the hash symbol.

```
1
2 # Class: Python Programming
3 # Tutor: Jeremy Pedersen
4 # program purpose : understanding the use of comments
5
6 print("Comments explains how the program works!")
```

Variables

- A variable is a name that stores a value for later use in a script.
- A variable can store a number, a string, a list, etc...

```
#the variable name is "m" and the value 6
m  = 6

#A variable can contain a decimal number

decimal_number = 7.5

#A variable can contain a string

string_variable = "I am a string"

# A variable can contain a list

list_variable = [1, 2, 3, 4, 5]
```

Python Data Types

- You can determine what data type Python has assigned to a variable by using the type function.

```
# Using the Type function

first = "I am a string"
second = 5
third = 6.7

print( type (first) )
print( type (second))
print( type (third) )
```

```
<class 'str'>
<class 'int'>
<class 'float'>
```

input()

- There will be times that you need your program to stop and ask for help from a human. The input function is a built-in function you can use for this, and it has the following syntax:

variable = input(user prompt)

```
#The use of the input function

name = input("Enter your full name please!")
print(name)
```

Arithmetic operators

```
# The addition(+) operator
print( 5 + 4)

#The substraction(-) operator
print (5 - 4)

#The division(/) operator
print (10 / 2)

#The multiplication (*) operator
print( 4 * 3)

#The modulus (%) operator
print ( 5 % 4) #result : 1

#The exponent (**) operator

print( 5**2) #result : 25
```