

The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)

# Implementing Python in the Classroom

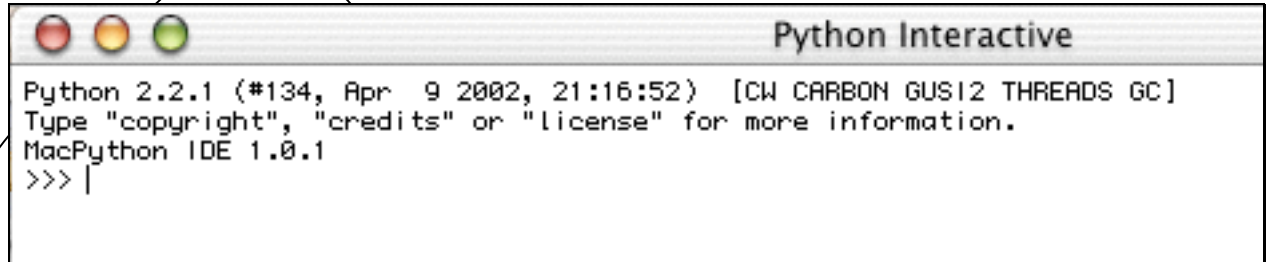
## Contents

Chapter	Topic	Page
1	Using the Python Development Environment	2
2	Sequence Commands	3
3	Binary Selections (IF)	5
4	Multi-way Selections	9
5	Basic data Type	11
	➤ integer	
	➤ string	
	➤ floating point	
	➤ boolean	
	➤ date and currency format	
6	Assignment 1	14
7	Pre-Test Repetition (While)	15
8	Post-Test Repetition	18
9	For Loops	19
10	Assignment 2	21
11	Appendix A	22
12	Appendix B	23

The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)

## Using the Python Development Environment

When you start up the Python program you will get a screen that looks like this (it will vary depending on the version you are using and which operating software you are using).



```
Python 2.2.1 (#134, Apr 9 2002, 21:16:52) [CW CARBON GUS12 THREADS GC]
Type "copyright", "credits" or "license" for more information.
MacPython IDE 1.0.1
>>> |
```

This provides you with a command line that can be used to enter commands. Try it out by typing in the following command followed by enter:

```
print "Hello Computer"
```

You should have seen the words **Hello Computer** on the next line. Now let's try and make our program a little more interactive. Type this line into the Python window:

```
name= raw_input("Please enter your name?")
```

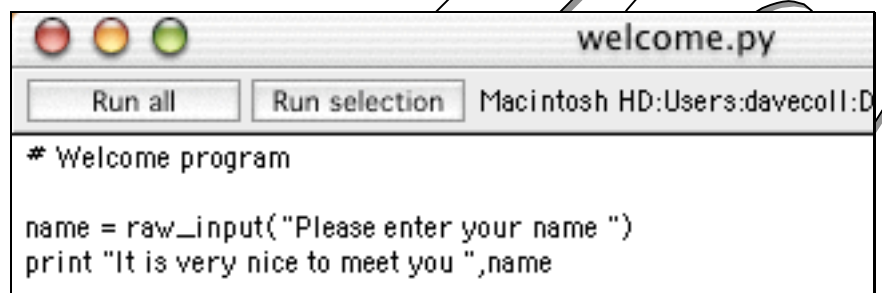
The computer will ask you to enter your name. Type in your name and press return. Now type the following line:

```
Print "It is very nice to meet you ",name
```

It should have displayed something like - **It is very nice to meet you Dave.**

The method you have used so far has been to enter the commands one at a time through the Python Shell (Win) or Python Interactive (Mac). While this can be a handy way to test single commands, it is a lot easier to group the commands together. This can be achieved in Python by selecting **New** from the **File** menu and then entering the commands into a sequence. It is of course important to enter all of the commands in the correct order. Enter the commands from the above program into a new window and save it a **welcome.py**. It should look like this:

Lines that start with a # are comments and are ignored. This is a form of documentation and is a good habit to get into



```
# Welcome program

name = raw_input("Please enter your name ")
print "It is very nice to meet you ",name
```

If you are using a Mac, click on the **Run All** button, or if you are using a windows machine select **Run Script** from the **Edit** menu (Control F5 is a handy short cut). It should run just like the example above.

## Binary Selections (IF)

As mentioned before, computers are dumb machines that need to be told exactly how to perform any task. When they need to make a decision it is crucial that they are told exactly to how to do this. The first method that can be used to do this is called a **binary selection**. Binary Selections allow the computer program to follow one of two paths. One of those paths may be to do nothing. Here is an example of a Python program that can be used to make a decision. Enter it into a new window and save it as **binary.py**

```
#Binary Selection  
# written by Dave Collins, 2002, No Limits IT  
  
name = raw_input("Please enter your first name: ")  
if (name=="Dave"):  
    print "You may have written this book"  
else:  
    print "You definitely didn't write this book"
```

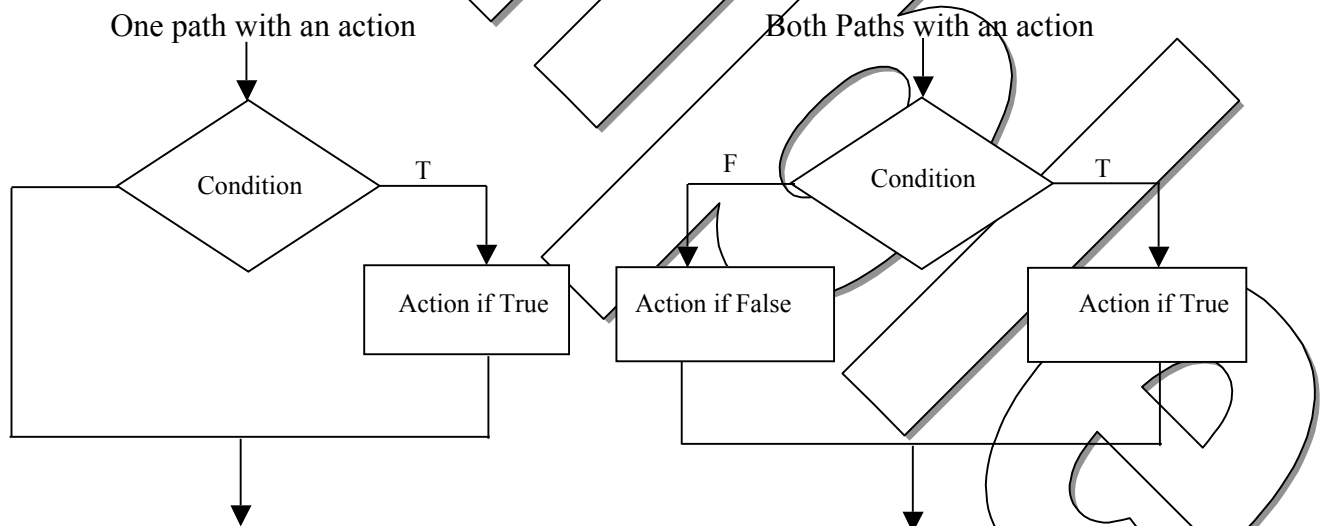
Now run this program and write down below what it does.

---

---

---

The flow chart for a binary selection can take one of two forms



Pseudocode is another method of describing an algorithm (a solution to a problem – your teacher will give you a more detailed definition of an algorithm). Pseudocode is combination of commonly used computer terms and English language. It is quite often preferred over Flowcharts by programmers as it more closely represents the languages they work in. Here is how these two control structures are represented in Pseudocode.

The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)

### One path with an action

```
IF condition is true THEN
    Action 1
END IF
```

### Two paths with actions

```
If condition is true THEN
    Action 1
ELSE
    Action 2
END IF
```

The first example that you entered into Python would look something like this in Pseudocode

```
Ask the person for their First Name
Store the First Name in name
IF Name ="Dave" THEN
    Display "You may have written this book"
ELSE
    Display "You definitely didn't write this book"
END IF
```

Let's find out how a Binary Selection works in Python.

All binary selection statements start with the keyword **IF**. Keywords are words that have a special meaning to the language and therefore can't be used for other purposes such as variable names. They are then followed by a condition. This will tell the computer how to choose which path to follow. It is a good programming habit to surround conditions by brackets. Each condition must have a logical operator and Python has the following available to make decisions.

Operator	Meaning
==	Is equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
AND	Joins two conditions together. Both conditions must be true.
OR	Joins two conditions together. At least one of the conditions must be true.
NOT	Reverses the logic of the condition. True becomes false, false becomes true.

**Important Note:** It is crucial that you understand the difference between "==" and "= ". A single equals sign is used to make to set the value of a variable. If you use "= " as a logical operator it will always return true if it can successfully set the variable. Use must use the double equal signs (==) to compare if two values are equal.

The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)

Here is the binary selection from the first example with explanatory notes on the side

```
if (name=="Dave"): ← Checks if Name is equal to Dave
    print "You may have written this book" ← Action performed if true
else:
    print "You definitely didn't write this book" ← Action performed if false
```

## Why Indent?

In most computer languages the indentation that appears in programming code is simply for the programmer so that they can better understand the flow of the code. Code without indentation is extremely difficult read. Just wait until you have to debug someone else's code that is missing internal documentation (comments) and indenting. Python is partially designed as a teaching language and has chosen to **enforce indentation**. This means that computer uses the indents to tell it which pieces of code it needs to execute for each condition.

### #Binary Selection

# written by Dave Collins, 2002, No Limits IT

```
name = raw_input("Please enter your first name: ")
```

```
if (name=="Dave"):
```

```
    print "You may have written this book"
```

```
    print "You must be a very smart person"
```

```
    print "You also aren't very humble"
```

```
else:
```

```
    print "You definitely didn't write this book"
```

```
    print "You still might be smarter than the author,"
```

```
    print "but you are clever enough to be using this book"
```

```
print "Everyone will see this line"
```

← All these actions are performed if true

← All these actions are performed if false

← This line will be performed regardless of the value of name

## Exercise 2

What do the following programs do. Try answering the question without entering them into python. Once you have answered the questions enter them into python to check your answer

1.

```
mark = input("Please enter the your mark for the examination")
```

```
if (mark>=65):
```

```
    print "You will get a least a Credit for this exam"
```

```
else:
```

```
    print "You will not get a Credit for this examination"
```

```
print "Thank you for using Grade Giver 2002"
```

The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)

2.

```
mark = input("Please enter the your mark for the examination")
average = input("Please enter the class average for the test")
if (mark>=average):
    print "You are better than the average student in the class"
else:
    print "You are not as good as the average student in the class"
print "Thank you for using Grade Giver 2002"
```

3.

```
mark = input("Please enter the your mark for the examination")
average = input("Please enter the class average for the test")
if (mark>=average) and (mark>=75):
    print "You are better than the average student in the class"
    print "You will receive a distinction"
else:
    print "You are not as good as the average student in the class"
    print "Your grade will be less than a distinction"
print "Thank you for using Grade Giver 2002"
```

4. Write a flow chart for question 3

**Design programs to meet the following criteria. Make sure you test each program and save it.**

1. Reads in a number and if the mark is 50 or over it tells the user they passed, otherwise it tells them that they failed.
2. Reads in a number and if the number is within the range of 1 to 100 it displays **Within Range**, otherwise it displays **Out of Range**
3. Asks the user to enter a password and if they enter the password **goaway7** the program displays **Access Granted** otherwise it says **Access Denied**.
4. Asks the user to enter a user name and a password. If they enter a user name of **Ryan** and a password of **blake2** the program displays **Access Granted** otherwise it says **Access Denied**.
5. Asks the user to enter a user name and a password. There are two acceptable users in this system. They are:

**User Name: Ryan Password: blake2**

**User Name: Kate Password: daniel5**

If they enter an acceptable user name and password the program displays **Access Granted** otherwise it says **Access Denied**.

**The full set of blackline masters is available for \$50 Australian (+GST and postage)  
Send me an email if you wish to get a price in American Dollars via [dave@nolimitsit.com](mailto:dave@nolimitsit.com)**

## Assignment 2

Create a program in Python that will allow two people to play the following game taken from Survivor 5 in Thailand (the American Television series). The program will be called Thai21.

- There are 21 flags at the start.
- Each player takes it in turn to take either 1, 2 or 3 flags. They can't take 0 or 4.
- The winner is the person who takes the last flag.

Sample output from the program should look like this:

```
The number of flags left are 21
Next Player is 1
Please enter the number of flags you wish to take: 2
The number of flags left are 19
Next Player is 2
Please enter the number of flags you wish to take: 3
The number of flags left are 16
Next Player is 1
Please enter the number of flags you wish to take: 2
The number of flags left are 14
Next Player is 2
Please enter the number of flags you wish to take: 3
The number of flags left are 11
Next Player is 1
Please enter the number of flags you wish to take: 2
The number of flags left are 9
Next Player is 2
Please enter the number of flags you wish to take: 2
The number of flags left are 7
Next Player is 1
Please enter the number of flags you wish to take: 1
The number of flags left are 6
Next Player is 2
Please enter the number of flags you wish to take: 2
The number of flags left are 4
Next Player is 1
Please enter the number of flags you wish to take: 1
The number of flags left are 3
Next Player is 2
Please enter the number of flags you wish to take: 3
Player 2 wins
```

**NO LIMITS**