Student

Online

Python

Programming

Workbook

## This Workbook

This workbook is very important and you must complete each challenge to progress. You will need it along with your blog to keep track of your work and your progress.

Read carefully and follow all instruction. The workbook is in parts which will help you achieve the following

* Python Achievement Level 3 : First Steps
  + Challenge 1 - 5
* Python Achievement Level 4 : Apprentice
  + Challenge 6 - 8
* Python Achievement Level 5 : OWL
  + Challenges 9 - 15
* Python Achievement Level 6 : Advance Skills
  + Challenges 16 - 26
* Python Achievement Level 7 : Professor
  + Challenge 27

## Help

**Ask** for help from **your team**

Complete them in **sequence** and do not move on to the next challenge until you have successfully got the program running.

If possible try the extension tasks.

Make sure that you save your solutions to your Python Practice folder using the challenge title/topic as the file name.

Look at the programs that you have completed in previous lessons for help if you struggle.

When you have completed an achievement level, let your teacher know so you can receive to achievement certificate.

## Using Python and VirtualBox

To use Python you must first open the Virtual Machine.

All work you want to save needs to be saved into your VirtualBoxFile. Failure to save here will mean you will lose your work.

## Organised

Life will be much harder if you don’t save your work carefully.

Create the following folder:

My Work Folder

VirtualBoxFiles

Python Practice

## Using Your Blog

You need to continue to use your blog.

**@start** of the lesson, you must create the POST for this lesson with the following details

1. POST Title : Programming in Python
2. Add the lesson objectives to your blog – “To complete as many challenges as possible”

**@end** of lesson, you must complete your post with the following information before PUBLISHING your post

1. List all challenges completed
2. Write more than 2 sentences on what you have achieved in the lesson and what you have learnt

In addition, while you hare working there will be places where you are asked to update details or attempt challenges which need to be added to your blog.

Good luck in your Python challenges.

## C:\Users\jwardell\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\92R9WSNQ\MC900331536[1].wmfInputting information and strings

print() function

The print function prints information to the screen. Type this in:

print (“Hello World”)

Input() function

The input function allows the user to enter information into the computer. Type this in:

name = input("Hi. What's your name? ")

print ("Hello, ", name)

Variables

A variable is used in a program to store information. In the program above the variable is **name.** Because the value is in quotes “” or ‘’, the variable type is called a **string**.

String

A string can contain letters, characters and numbers.

### Challenge 1 - Write a program that….

Use variable names:

firstName

surnameName

1. Asks for the first name and then a surname.
2. Print the First name and then the second name
3. On the next line print the second name and then the first name

Extension: The name may not have a space between first and second names,  
 can you work out how to add one?

Save as: Challenge 1 - Names

UPDATE BLOG…add “Completed Challenge 1” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

Strings can be joined together:

### Challenge 2 - Write a program that…

Use a new variable **fullName.** Add to your program:

1. fullName = firstName + surnameName
2. print fullName

Save as :Challenge 2 - Fullnames

UPDATE BLOG …add “Completed Challenge 2” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

## Working with INT and FLOAT to handle numbers in Python

We have used the input command to ask the user to enter text. We’re going to use this again but with numbers. What do you think this code will do?

### Challenge 3 - Write a program that….

Type this in:

numberOne = input(“Enter a number between 0 and 10”)

numberTwo = input(“Enter a number between 0 and 10”)

print (numberOne + numberTwo)

Save as : Challenge3 - numbers

UPDATE BLOG : Write the following questions and your answers to these questions in your blog.

What happened? Why do you think this happened?

UPDATE BLOG …add “Completed Challenge 3” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog .

### Challenge 4 – Change the Program

Type this in:

numberOne = int(input(“Enter a number between 0 and 10”))

numberTwo = int(input(“Enter a number between 0 and 10”))

print (numberOne + numberTwo)

Save as : Challenge 4 - intnumbers

Type in 5 and 5 as your numbers to test your program

Type in 5.5 and 4.5 to test your program again. You should receive an error message.

UPDATE BLOG : Write the following questions and your answers to these questions in your blog.

What happened? Why do you think this happened?

UPDATE BLOG …add “Completed Challenge 4” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 5 – Change the Program

Type this in:

numberOne = float(input(“Enter a number between 0 and 10”))

numberTwo = float(input(“Enter a number between 0 and 10”))

print (numberOne + numberTwo)

Save as : Challenge 5 - floatnumbers

Type in 5 and 5 as your numbers to test your program

Type in 5.5 and 4.5 to test your program again. Did you get an error message?

UPDATE BLOG : Write the following questions and your answers to these questions in your blog

What happened? Why do you think this happened?

UPDATE BLOG …add “Completed Challenge 5” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

### Recap :

You have started Python programming. You have completed 5 challenges. You have created string, integer and float variables. You have explore what is different between all of these any put your observations into your blog.

**Well done, you have completed your first steps. Make sure all your challenges (1-5) are saved.**

**You need claim your first steps certificate from teacher before proceeding.**

## Apprentice Level Challenges

## Combining string and number handling in Python

### Challenge 6 – Calorie Counter

Now let’s build a calorie counter. The NHS recommends that an adult male takes on board 2,500 calories per-day and an adult woman takes on 2,000 calories per-day. Build your program for a woman or a man. A sample of code is provided here for you to modify to meet this challenge.

Create a new folder called Challenge 6. Save your program as **CalorieCounter** into this new folder

print("YOUR NAME’s calorie counter")

calories = int(input("How many calories have you eaten today?"))

s=2000-calories

print("You can eat", s, "calories today")

1. Line 3 of the code subtracts how many calories they’ve eaten and stores it as the **variable** s

Line 4 then prints outs in between 2 ’strings’ of text

2. This asks the user to enter how many calories they have eaten as the **variable** s. It stores it as an integer (whole number)

UPDATE BLOG …add “Completed Challenge 6 CalorieCounter” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

**Challenge**

Write a program that asks the user to enter how much they have spent on their school dinner. It should subtract how much they’ve spent from the amount they had at the start of the day. Display the result on the screen. As an extension – do this for 5 days – if the total goes negative give a warning message that they do not have enough money on their account.

Save your work as **SchoolDinner** in your Challenge 6 folder

UPDATE BLOG …add “Completed Challenge 6 SchoolDinner” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

Now try these challenges. Think carefully – what type of variable will you need to use?

Remember that by ***default*** variables are strings.

### Challenge 7 – Area Calculator

Jennifer wants to carpet her new room with pink carpet. Create a program that will work out the area of any sized room, (length x width).

*Save as Challenge7 - Square*

UPDATE BLOG …add “Completed Challenge 7 – Area Calculator” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

### Challenge 8 – Days Alive Calculator

Write a program to work out how many days you have been alive for (to the nearest year for the moment - there are 365 days in a year)

Get the program to ask for the person’s name and age.

Develop this to work out how many hours this is – 24 hours per day

Develop it further to work out how many minutes and seconds you have lived for – 60 minutes per hour / 60 seconds per minute.

Make sure the information is clearly displayed on the screen.

*Save as Challenge 8 – Age*

UPDATE BLOG …add “Completed Challenge 8 – Age” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

### Recap :

You have designed and created your own Python programs.

**Well done, you have completed your Apprentice challenges. Make sure all your challenges (6-8) are saved.**

**You need claim your Apprentice certificate from teacher before proceeding.**

## 

## Some more about Strings

Strings are variables that contain letters, numbers and symbols. There are lots of things that can be done with strings.

### Challenge 9 – I am an apprentice programmer

Type this in:

Quote = “i am an apprentice programmer”

Print (Quote)

What is shown on the screen should be the same as you typed in.

Now try changing your programme to this:

Type this in:

Quote = “i am apprentice programmer”

Print (Quote.lower())

This is known as a string **method**

|  |  |
| --- | --- |
| **String method** | **Description of what it does** |
| lower() |  |
| upper() |  |
| title() |  |
| swapcase() |  |
| capitalize() |  |

Copy this table into your blog. Then instead of lower use these methods and write down what happens:

UPDATE BLOG …add “Completed Challenge 9 – String Methods” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

### Challenge 10 – Concatenation

Strings can be joined together – which is called concatenation.

Write a programme that asks for 3 gifts that someone would like for Christmas.

Add the variables (something like this) : giftTotal = gift1 + gift2 + gift3

Then print the variable giftTotal (make sure there are spaces between each word)

gift1 = input(“What gift would you like most”)

gift2 =

gift3 =

giftTotal = gift1+gift2+gift3

Save as **Challenge 10 -** **Christmas Gifts**

UPDATE BLOG …add “Completed Challenge 10 – Concatenation” in your blog.

Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog

## How to use Python for basic math

Let’s look at how python calculates using numbers.

### Challenge 11 – Multiplication

Let’s see how much a student spends on food at school. Do you remember why we have to use int or float?

food = float(input(“How much do you spend on lunchtime food in an average day?”)

food = food \*5

print (“This means that in a week you will spend “, food)

Save as **Challenge 11 - Food spending**

UPDATE BLOG …add “Completed Challenge 11 – Multiplication” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 12 – Dividing

An Aunt wins on the lottery. She gives £1000 to you. You are thinking of sharing it with some of your family. How much would each person get (try different numbers of people to share it with). To get you started:

shares = int(input(“How many people do you want to share the money with?”)

money = 1000 / *????*

*finish this off*

If you want an answer to be a decimal, instead of typing 1000, try entering 1000.0. Try it to see what happens.

Save as: **Challenge 12 - sharing my money out**

UPDATE BLOG …add “Completed Challenge 12 – dividing” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 13 – Modulus

If we divide two numbers and they don’t divide evenly we get a remainder. Modulus gives us the remainder of a sum.

For example 7/2 = 3 with remainder 1. Try this:

Print 7 % 2

You should get the answer 1. Try some other numbers to see what you get.

UPDATE BLOG …add “Completed Challenge 13 – Modulus” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 14 – Addition

Addition is easy. Try this:

length = 10

length = length + 20

What’s the value of the variable length:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Try it out to test it.

length = 10

length += 20

Another way of writing this in python is:

Test it out. It’s just quicker to write!

## Operands

The +, -, /, \* symbols are called operators. That’s because they “operate on” the numbers we put around them.

Write in what these are equivalent to

Copy and complete this table into your blog.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Is Equivalent to** |
| \*= | K\*=5 | K = K \* 5 |
| /= | k/=5 |  |
| %= | K%=5 |  |
| += | K+=5 |  |
| -= | K-=5 |  |

UPDATE BLOG …add “Completed Challenge 14 – Operands” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 15 - Restaurant Tip

Write a program.

Two people eat dinner at a restaurant and want to split the bill.

The total comes to £100 and they want to leave a 15% tip. How much should each person pay?

Harder: make the program ask how many people there are, what percentage the tip should be and how much the bill comes to.

Save this as **Challenge 15 - Restaurant tip**.

### Recap :

You have designed and created your own advanced Python programs. You have learnt and used string manipulation, mathematical operators and advanced operands.

**Well done, you have completed your OWL challenges. Make sure all your challenges (9-15) are saved.**

**You need claim your OWL certificate from teacher before proceeding.**

## Advance Skills Challenges

## Making decisions in programs

Python can make decisions based on the input. To make decisions, programs check to see **if** a condition is true or not.

Python has a few ways to test something, and there are only two possible answers for each test: ***true* or *false***

### Challenge 16 – Magic 8 Ball

The code below will use Python’s **random number generator** to make a magic 8 ball game.

 Make this game and save it as **Challenge 16 -** **Magic8Ball.**

import random

answer1=(“Absolutely!”)

answer2=(“No way Pedro!”)

answer3=(“Go for it tiger.”)

print(“Welcome to the Magic 8 Ball game—use it to

answer your questions...”)

question = input(“Ask me for any advice and I’ll help you out. Type in your question and then press Enter for an answer.”)

print(“shaking.... \n” \* 4)

choice=random.randint(1,3)

if choice == 1:

answer=answer1

elif choice == 2:

answer=answer2

else:

answer=answer3

print(answer)

Can you modify this so that it chooses from 5 possible answers?

UPDATE BLOG …add “Completed Challenge 16 – Magic8ball” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

Copy this table into your blog and complete. Write down what these lines of code are doing:

|  |  |
| --- | --- |
| **Programme** | **What it does…** |
| import random | Imports the built in module “Random” |
| answer1=(“Absolutely!”) | Defines three variables as possible answers |
| answer2=(“No way Pedro!”) |
| answer3=(“Go for it tiger.”) |
|  |  |
| print(“Welcome to the Magic 8 Ball game—use it to answer your questions...”) | Welcomes the user by displaying a friendly message |
|  |  |
| question = input(“Ask me for any advice and I’ll help you out. Type in your question and then press Enter for an answer.”) | Prompts the user to input their question, which is stored as a string to the variable “question” |
|  |  |
| print(“shaking.... \n” \* 4) | Displays the text “shaking” four times, under each other |
|  |  |
| choice=random.randint(1,3) | Saves a random number with a value between 1 and 3 to choice |
|  |  |
| if choice == 1: | If the number is 1, it displays the first answer |
| answer=answer1 |
| elif choice == 2: | If the number is 2, it displays the second answer |
| answer=answer2 |
| else: | If the number is 3, it displays the third answer |
| answer=answer3 |
|  |  |
| print(answer) | Displays the answer to the user’s question |
|  |  |

Indenting is very important in Python, it’s a necessary part of how you write the code. Indenting tells Python where blocks of code start and where they end. The Python IDE will do this for you – just be aware that it is doing it.

For testing whether two things are equal Python uses a double equal sign (==) .

Examples: if myAnswer == correctAnswer:

if numberWrong == 3:

if name == “Sharon”:

Copy this table into your blog.

|  |  |  |
| --- | --- | --- |
| **Comparison Operators** | **What the symbols mean** | **Example** |
| == | are two things equal? | firstName == “Gary” |
| != | are two things not equal? | firstName != “Gary” |
| < | less than | income < 300 |
| > | greater than | income > 300 |
| >= | greater than OR equal to | income >=300 |
| <= | less than OR equal to | income <= 300 |

### Challenge 16 – If

Let’s write a programme that compares two numbers.

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

if num1 > num2:

print (num1, " is great than ", num2)

if num1 < num2:

print (num2, " is greater than", num1)

if num1 == num2:

print (num1, "is equal to", num2)

Save this as: Challenge 16 - compare 2 numbers

This tests each combination and does something (prints a line out) if the test is **TRUE**

UPDATE BLOG …add “Completed Challenge 16 – IF” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

Python can do other things if the test is **FALSE**:

* *Do another test* – If the test is False we can get Python to another test: we use **elif** (short for “else if”)
* *Do something else* – If all the other tests come out false do something else: we use **else**

We can re-write the code to use these features:

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

if num1 > num2:

print (num1, " is great than ", num2)

elif num1 < num2:

print (num2, " is greater than", num1)

else:

print (num1, "is equal to", num2)

Why do we **not** need to test if num1==num2 but can print out :

print (num1, "is equal to", num2)

knowing it to be TRUE?

### Challenge 17 – Mega Sale

A local shop is having a promotion. If you spend over £10 you will get a £1 voucher to spend next time you come in the store. If you spend over £20 you get a £3 voucher.

Write a programme to tell the sales assistant which voucher to give the customer.

Save this as: challenge 17 - megasale

UPDATE BLOG …add “Completed Challenge 17 – Mega Sale” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 18 – Happy Message

Write a programme that gives a message depending upon how happy they say they are.

You could get the user to rate how happy they feel on a scale between 1 and 10. If the reply is 3 or less it gives one message.

Between 4 and 7 (including these numbers) they get another message.

8 and above they get a different message.

Try to make the messages ones to make them happy all day long!

Save this as: challenge 18 - Happy message

UPDATE BLOG …add “Completed Challenge 18 – Happy Message” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 19 – Mobile Phone Costs

You want to see how much a mobile phone will cost. There are charges for sending pictures (£0.35), for texts (£0.10) and for data (£2.50 for 500MB).

1. Write a program that asks the user for how many pictures, texts and data they would use each month. It should then calculate a total bill for the month.
2. If the total comes to more than £10 they would be better on a contract. Get the programme to give them this advice.

Save this as : Challenge 19 – Mobile Phone Costs

UPDATE BLOG …add “Completed Challenge 19 – Mobile Phone Costs” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 20 – Secret password

Make a program where the user has to enter a secret password to use the programme.

The programme could be one you’ve already written, or make it display a message when they have got the password correct.

Save this as : Challenge 20 – Secret password

UPDATE BLOG …add “Completed Challenge 20 – Secret Password” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

## Boolean or logical expression

Boolean is a type of arithmetic that only uses two values: true or false, yes or no, 1 or 0.

It was invented by an English mathematician George Boole.

We use Boolean expressions and can combine them with **and**, **or** and **not** to make decisions in our programmes.

### Challenge 21 – For loops

Sometimes we want to do something a number of times.

We may know how many times we want to do it – and we use a **counting loop** OR

We may want to do it until something happens – and then we can use a **conditional loop.**

## Conditional Loops

This is the simplest loop. Try these out and make a note in your blog of what they do:

This code:

for a in range (10):

print(a)

This code:

for a in range (1, 11,2):

print(a)

for a in range (1, 11):

print(a)

for a in range (1, 10):

print(a)

This code:

This code:

for a in range (1, 11,2):

The 1 tells the loop:

The 1 tells the loop:

The 1 tells the loop:

UPDATE BLOG …add “Completed Challenge 21 – For loops” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 22 – For loops

Write a loop that displays numbers 10 to 100 in steps of 5.

Save as : challenge 22 – forloops

UPDATE BLOG …add “Completed Challenge 22 – for loops” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 23 – For loops

Write a loop that displays the 5 times table

Save as Challenge 23 - 5 timestable.

UPDATE BLOG …add “Completed Challenge 23 – fivetimestable” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 24 – While loops

All of this programme should be familiar to you apart from the while loop. Think about what it does before typing it in. Copy and complete this table in your blog.

|  |  |
| --- | --- |
| **Code** | **Describe what the code does** |
| import random | Imports the built in module “Random” |
| guess ="" | Sets a variable “guess” to equal an empty string |
|  |  |
| print ("I've thought of a number between 1 and 10. Try to guess it.") | Displays the text in the brackets |
|  |  |
| randomNumber = random.randrange (1,10) | Saves a random number between 1 and 10 to randomNumber |
|  |  |
| while guess != randomNumber:  guess = int(input("Guess the number"))  print("You got it wrong") | Repeats this loop until the guess is the same as the randomNumber, because the conditional becomes True. This code is particularly bad because you would be told you were wrong when you get it right |
|  |  |
| input ("Well done. Press any key to exit.") | Waits until the user presses enter to finish the script |

Save as: Challenge 24 - guess a number

**Extra Challenge**.

Can you make the programme show how many guesses have been taken?

**Why is the int necessary?**

guess = int(input("Guess the number"))

The **while** loop repeats whist the conditions is **TRUE**.

In this case this is when guess is **NOT** equal to randomNumber.

UPDATE BLOG …add “Completed Challenge 24 – Guess a number” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

## Comments

Comments are added to a program to explain what the program does and what the section of code is doing. Programmers do it so that if the code is looked at in the future (either by themselves or others) they can work out what the code is doing.

Comments can easily be added using # at the start of the line of code. Python ignores anything on a line that begins with a #.

Load your last program: guess a number 1 to 10.

Add your own comments to the programme to explain what the parts of the code are doing:

|  |
| --- |
| *# this programme creates a random number between 1 and 10*  *# the user guesses the number. It ends when the user guesses the number correctly*  import random |
| guess ="" |
|  |
| print ("I've thought of a number between 1 and 10. Try to guess it.") |
|  |
| randomNumber = random.randrange (1,10) |
| *# this loop runs until the number guessed equals the randomNumber* |
| while guess != randomNumber: |
| guess = int(input("Guess the number")) |
| print("You got it wrong") |
|  |
| input ("Well done. Press any key to exit.") |

We can also add comments to many lines at a time by using three quotes together “”” at the beginning and end of our descriptions:

“” This programme calculates how much savings you have after a number of months.

Written by: May Dup

Date: Created 23/11/12

“””

Some further challenges for you to have a go at. Use this workbook and the programs you have already used to help you solve the problems.

### Challenge 25 – Rock, Paper, Scissors Game

We’ve all played the rock, paper, scissors game. The computer will play against you. Get the computer to ask for the player’s name.

The game rules are simple: rock beats scissors, scissors beat paper, paper beats rock. Two of the same gives a draw.

You can start with pseudo code to help you identify the steps needed.

Add **comments** to the game so it’s clear what the code in the game is going.

Extension: Ask the user how many rounds they want to play, between 3 and 10.

Keep score and show this at the end of the game.

Further extension: Check to make sure the user can only enter a number between 3 and 10 and give them an error message.

Save as : Challenge 25 – Rock, paper, scissors

UPDATE BLOG …add “Completed Challenge 25 – Rock Paper Scissors” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Challenge 26 – Making a times table (using nested if)

With a times table we start with a number and then multiply this by a series of numbers.

For the 2 times table we multiply it by 1 to 12, before moving on to do the 3 times table, etc.

Computers are very good at this sort of task and we can do it using loops within loops to consider every combination – these are called **NESTED LOOPS.**

Type this code in (try to predict what you will get before you do it).

**for i in range(1,13):** # i is the first number we are going to multiply by

# print a title at the top of the times table to show which times table

# we are working on

**print (i, "Times table\n")**

**for j in range (1,13):** # loop through the numbers we are multiplying

# i by and then print the results

**print (i, "times", j, " = ", i\*j)**

## Extension:

Maths Teachers will be impressed if you’ve worked out the 1345 x 1 to 12 times table - change the program to work out your own different times tables.

Save as : challenge 26 – times tables

UPDATE BLOG …add “Completed Challenge 26 – Times Tables” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

### Recap :

You have designed and created and used advanced skills Python programs. You have used structures IF, IF-ELSIF, FOR, WHILE.

**Well done, you have completed your Advanced Skills challenges. Make sure all your challenges (16-26) are saved.**

**You need claim your Advanced Skills certificate from teacher before proceeding.**

## Professor Level Challenges

### http://www.oldelpaso.co.uk/~/media/Images/OEP_UK/Recipes/Original-smoky-bbq-fajitas2.ashxChallenge 27 – Chicken Wraps (using nested if)

Students can choose to have any combination of ingredients in their chicken wrap.

Write a programme to calculate the price of each combination of Chicken Wraps.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tortilla**  **http://0.tqn.com/d/mexicanfood/1/0/l/1/tortilla.jpg** | **Chicken**  http://t1.gstatic.com/images?q=tbn:ANd9GcQIbU1LLt2Qw1fLxQd_nbg9Gj8qS6X1hMK0-SzT1yV8T-TLylVFjCuHKkuv | **Salad**  http://t2.gstatic.com/images?q=tbn:ANd9GcSisgWZnXEfBwFIxrLai4mnGoqvB7xwENcoUohyBqP3By_OCVnNz0Xd4Rx-fw | **Chilli Sauce**  http://t1.gstatic.com/images?q=tbn:ANd9GcTS66-BN_GijYualEBhJgEf-UgEd5cc8rOri7b9Kd4rz9gt4-JXvhI-Bq67bQ |
| **£0.25** | **£0.55** | **£0.12** | **£0.24** |

Here are the combinations that can be selected:

start

Tortilla choice

Chicken choice

Salad choice

Chilli choice

yes

no

y

y

n

y

n

n

n

n

n

n

y

y

y

y

y

y

n

y

n

n

n

n

n

n

y

y

y

y

This is a decision tree. It shows that for each option there is only a Y / N answer (Boolean).

The option chosen above is Yes a Tortilla, No Chicken, Yes Salad, Yes for Chilli.

To start with see what this code does. It uses loops within loops to consider every combination – these are called **NESTED LOOPS.**

print("\tTortilla \tChicken \tSalad \tChilli Sauce")

count = 1

/t tabs to align the columns and data rather than spaces

for tortilla in (0,1):

for chicken in (0,1):

print("#", count)

print ("\t\t",tortilla, "\t\t", chicken)

count =count+1

Note: ***The \t is an instruction to Python to tab the data – that is put it into columns***

Develop this to that it creates all combinations for Salad and Chilli.

Test the programme to see if the results are as you would expect. You will have to do some manual calculations and work out the costs ar.

## Extension

Add another column labelled cost. Calculate the total cost of the ingredients in each option.

Think carefully before you do this – there is a very simple way, and much longer ways.

Test it. You will have to do some manual calculations and work out what the costs should be.

Save as : challenge 27 – chicken wraps

UPDATE BLOG …add “Completed Challenge 27 – Chicken wraps” in your blog. Indicate whether the challenge was EASY/OKAY/DIFFICULT in your blog.

## Functions

Programming languages have pre-made functions for us to use. Python has some such as print() or random.

But we can make our own functions, these are called user-defined functions:

A function is a block of organised, reusable code that is used to perform a single action.

By using functions we can make our programs modular (made up of separate parts) and simpler, because we don’t have to write the same instructions many time but reuse the same code.

### Challenge 28 – Area or Perimeter

Let’s make a programme to calculate the area **OR** the perimeter of a rectangle.

**Area = length \* width**

**Perimeter = length \*2 + width \*2**

width = 100

length = 200

We can define a function to calculate area when we need it and one to calculate the perimeter when we need it:

To define a function type in def and the function name:

def area():

def area():

shapeArea = length \* width

print("Area = ",shapeArea)

def perimeter():

shapePerimeter = length\*2 + width\*2

print ("Perimeter = ", shapePerimeter)

Look at the code and think what it is doing.

Then type it into Python and run the code. Copy this table into your blog and write your own explanation.

|  |  |
| --- | --- |
| **Code** | **Explanation of the code** |
| length = 200 |  |
| width = 100 |  |
|  |  |
| response = None |  |
|  |  |
| # function to calculate area |  |
| def area():  shapeArea = length \* width  print("Area = ",shapeArea) |  |
|  |  |
| def perimeter():  shapePerimeter = length\*2 + width\*2  print ("Perimeter = ", shapePerimeter) |  |
|  |  |
| while response not in ("a","p"):  response = input("Do you want to calculate area or perimeter? Enter a or p") |  |
| if response == "a" or "A":  area()  elif response == "p" or "P":  perimeter() |  |

Add your own comments to the program.

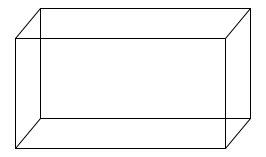
Add these questions to your blog and type up your answers.

Why do you think the functions are defined first?

What would happen if the user input were first?

## Extension

Have the programme ask the user to input their own measurements.

[](http://upload.wikimedia.org/wikipedia/commons/3/32/Cuboid_simple.svg)Add a function to calculate the volume.

To work out the volume of a regular cuboid shape is:

length \* width \* height

Add an option to the programme to calculate the volume defining a new function called volume()

### Recap :

You have designed and created and used Professor level Python programs. You have used functions to structure a business style professional programs.

**Well done, you have completed your Professor challenges.**

**You need claim your Professor certificate from teacher before proceeding.**