

# **Basic Python WORKBOOK**



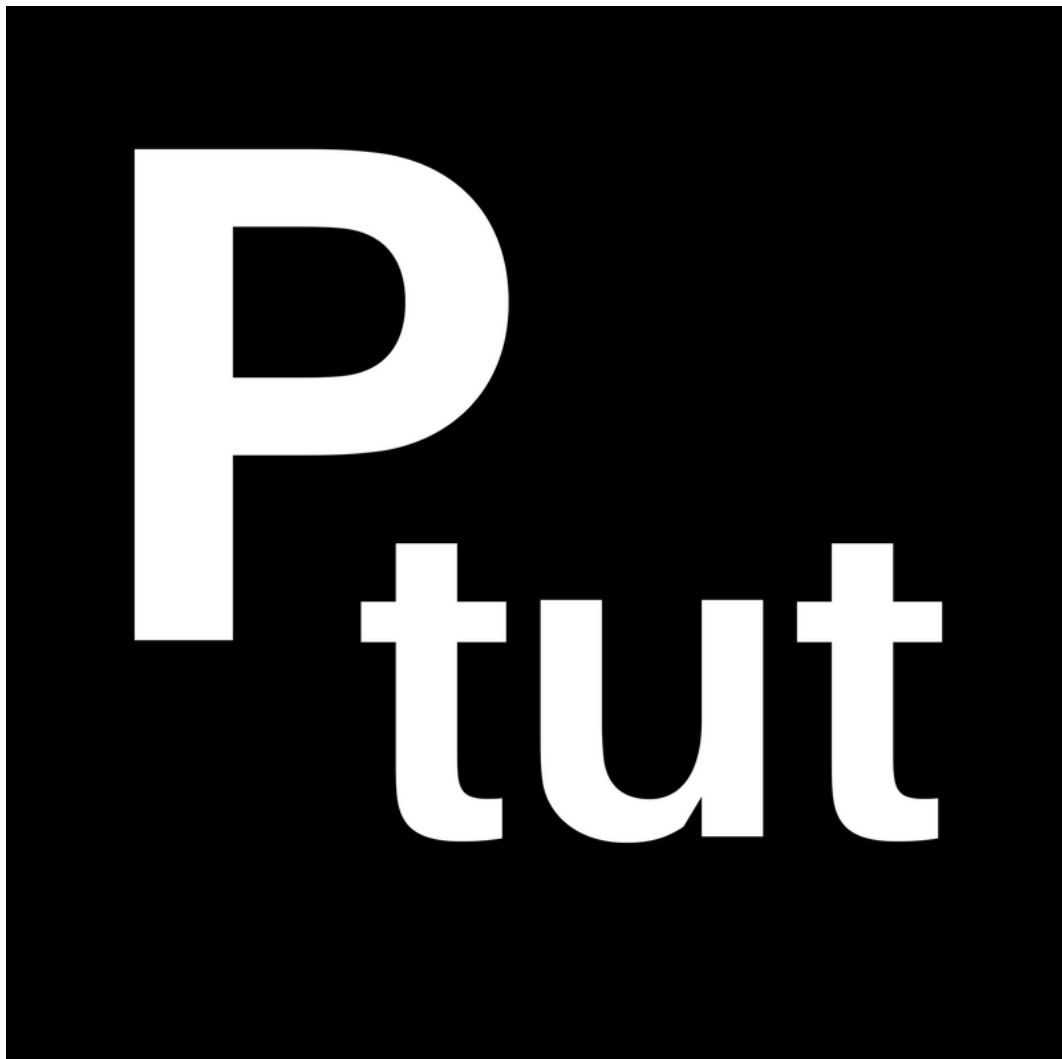
**Matthew Dewey**

# Introduction;

Welcome to your basic Python workbook! First of all, let me thank you for using this resource as you are no doubt enrolled in Programming Tut's free Python Code Breakdown Course. By becoming a Programming Tut student you unlock many benefits and workbooks to help you with your programming. As well as receive 25% off all courses on our website, [www.programmingtut.com](http://www.programmingtut.com).

My name is Matthew Dewey. I am lead programming instructor at Programming Tut. Having studied many programming languages over the years I can say that Python is an extraordinary language that offers you so many benefits. Python today is so useful with the expansion of the World Wide Web.

By taking this course we shall of course begin with the basics of Python programming, moving straight on to 'What is Python?'



# Chapter 1 - What is Python?;

Python is an Object Orientated programming language.

Full stop, let's break this done:

**Object Orientated – When a programming language is ‘object orientated’ it means the language makes use of objects. Objects are sections of programming containing methods, methods being miniature programs that can do small tasks. By being object orientated, Python creates programs that take advantage of these methods, utilizing them fully.**

Python was first introduced in 1990. During this time it grew in popularity and took the world by storm. Python became the most common tool next to JavaScript in web development. Many programs and website based services make use of Python. As you can imagine this is a huge triumph for any programming language, but what takes it further is that large companies such as Google and YouTube make use of Python.

Guido van Rossum, the developer of Python, soon made millions of dollars earning himself a permanent position with Google because of his constant work and development of Python. This year, 2018, a stable release of Python lead to a spike in Python programmers, making it a truly outstanding development in programming.

Today, Python is the second-most used programming language, second only to JavaScript. However, the gap between the two is only closing as Python grows more in popularity. What we can say for sure is that Python will always be a needed language and its programmers the most highly paid.

## **Chapter 1 – Quiz**

1. Who developed the Python programming language?

- A) Steve Jobs
- B) Guido van Rossum
- C) Kim Knapp
- D) Bill Gates

2. Which year was Python released?

- A) 2018
- B) 1995
- C) 2000
- D) 1990

3. “Python is the \_\_\_\_ -most used language.”

- A) Third
- B) Tenth
- C) Second
- D) Fifth

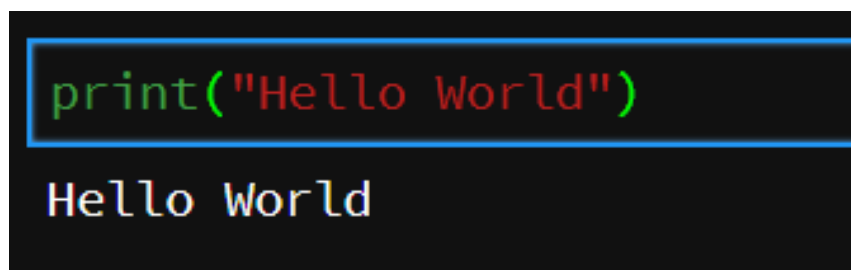
ANSWERS:

## Chapter 2 – Simplest of Code

Let us take a look at the most basic of Python code. A simple output line that prints a line of text saying, 'Hello World.'

**Hello World is the most common output any beginner creates. It signifies the programmer's entry into the programming world and thus it is known throughout the programming community that covers the globe.**

The line of code looks as follows:

A screenshot of a code editor with a dark background. The first line of code is `print("Hello World")`, where `print` is green, the opening and closing quotes are red, and the text inside is white. Below the code, the output `Hello World` is displayed in white text.

By typing in **`print("Hello World")`** you have begun your journey into programming. Let us break down this simple line of code further.

`Print()` is what we call a method. Methods as discussed before are small programs that perform a task. In this case, the `print` method outputs a line of text.

How we identify common methods is by the set of `()` that follow. Not all methods have these, but the most common ones do. The `print()` method takes what is inside and outputs it once the code it run.

What it outputs has to either be text or a variable. In this case we outputted some text. Text is kept in a set of `""` or `'''`. In Python, this is how we can tell a text from a variable name. A variable name does not have a set of `""` or `'''`

around it. We will discuss more on variables in the next chapter, but for now, we have learnt a very important piece of code to your programming career.

Before we close this chapter let's take a look at these segments of code:

```
print("Hello")  
print("World")
```

Hello  
World

and

```
print("Hello\nWorld")
```

Hello  
World

What you see here are two ways in which you can print on two separate lines. In the first image you see a program with two lines of code to print the two lines of text. In the second we use a shortcut to print the same result

**\n is a tag. The \n tag is used to start a new line when written in text. It is a tag which is used commonly by data processing programmers. Knowledge of these tags always proves useful in a pinch.**

**Another common tag is the \t tag, which creates a tab space (four spaces)**

## Chapter 2 – Quiz

1. What kind of method is the print() method?

- A) Input method
- B) Data method
- C) Output method
- D) Void method

2. Which of these is NOT text?

- A) "1234"
- B) 'Hi'
- C) True
- D) '1'

3. What is the result of this code: `print("Hello\t\nMy name is\n\t John")`

- A. Hello  
My name is  
John
- B) Hello My name is John
- C) Hello My name is John
- D) Hello John

Answers:

1) C 2) C 3) A

# **Chapter 3 – Data Types and Variables**

When you studied mathematics you no doubt learned there are different number types. Real numbers, rational, irrational and so on. Well, like number types there are also different data types. There are four data types we will be discussing. Strings, integers, floats and booleans.

## **Strings**

You have already encountered strings. Strings are lines of text, lines of text being one or more characters encapsulated in "" or more commonly ' '.

Eg: "John grew up on a farm"

"123 & 124"

## **Integers**

Integers are whole numbers ranging from negative infinity to positive infinity.

Eg: -72983

93747

33



## **Floats**

Floats have a similar, but far larger range than that of integers. Floats range from the negative infinity to positive infinity as well, but include all decimal point numbers as well.

Eg: 56.898

129730750237507.0232414

1.0

## **Boolean**

Boolean is a special data type. It has two values. True and False, or in machine code, 1 and 0.

## **Variables**

Variables in programming are the same as variables in mathematics. Variables are containers for values of any data type. To create a variable you simply type what you want to call it. Before we do that, let's go over some naming conventions.

1. You do not use special characters (#\$%^&\*etc) or numbers (23456etc)
2. You do not use spaces, but rather, underscores ( \_ )
3. Use lower case letters
4. Be smart naming, descriptive

Eg: name

first\_name

this\_is\_a\_very\_specific\_variable

Here is an example of assigning a value to a variable:

```
name = "John"

name

'John'
```

Notice, once I gave the name variable a value, I ran another line of code where I just typed the variable name. What this did is return the value and from the " we can see that it is a String data type.

```
name = "John"

name

'John'

name = 5

name

5
```

Notice how all variables can be overwritten, so be careful in your variable creation when working with larger numbers. You don't want to reuse a variable thinking it is the first time you created it. In this code you see me overwrite a variable containing a string with an integer.

## Chapter 3 – Quiz

1. Which is a valid variable name?

- A. print
- B. hello
- C. I\_am\_1
- D. this\_is\_a\_great name

2. Which is not a correct way to assign a variable a value?

- A. name = 56
- B. num = "Hello"
- C. num is 10
- D. bool = True

3. Which data type will this variable be: num = 5.67

- A. Float
- B. Integer
- C. String
- D. Boolean

ANSWERS:

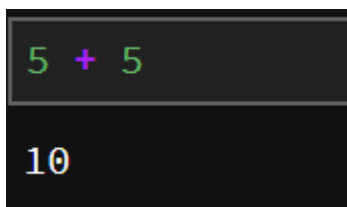
1) C 2) C 3) A

# Chapter 4– Programming Mathematics

Mathematics in programming has small changes compared to real mathematics done on a piece of paper or in a calculator. Here are the basic math functions you need to know. BODMAS applies in programming.

## **Addition**

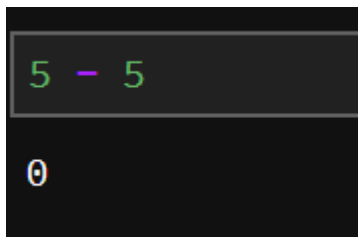
Addition is done with the + symbol



```
5 + 5  
10
```

## **Subtraction**

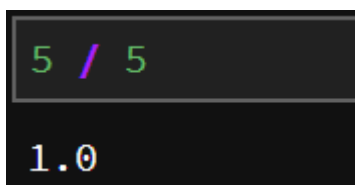
Subtraction is done with the – symbol



```
5 - 5  
0
```

## **Division**

Division is done through the / symbol (notice integer is no float)



```
5 / 5  
1.0
```

## Multiplication

Multiplication is done with the \* symbol

```
5 * 5
25
```

## Modulus

Modulus is used to tell you the remainder of divisible numbers. Eg: 5 goes into 9 once, the remainder is 4. ( $9 - 5 = 4$ )

OR

5 goes into 18 three times, remainder is 3. ( $18 - 15 = 3$ )

Modulus is done with the % symbol.

```
18 % 5
3
```

## Division with integral result (discard remainder)

This is the inverse of modulus, it discards the remainder and returns an integer based on how many times a number can fit into another number. This is done with the // symbols (two forward slashes). Eg:  $9 // 2 = 4$

```
9 // 2
4
```

## Chapter 4 – Quiz

1. What is the result of:  $5 // 9$

- A. 1
- B. 4
- C. 0
- D. 3

2. What is the result of:  $25 \% 6$

- A. 1
- B. 2
- C. 5
- D. 4

3. What is the result of:  $(5*5-6) + 5$

- A. 25
- B. 89
- C. -6
- D. 24

ANSWERS:

1) C 2) A 3) D

# Chapter 5 - If Statements

If statements are used to check data before running code. If statements make use of operators and clauses to see if values prove true before running code. For example, say you want to print 'hello', but only if another value is equal to 'hello'. Your if statement would look like:

```
greet = "hello"

if greet == 'hello':
    print("Hello")

Hello
```

An if statement is structured as follows.

If (clause is true):  
    (run following code)

Clauses are the section that follow the if and end at the :

Clauses make use of operators which are characters or sets of characters used to compare to values. In the clip above we compared greet's value to 'hello'. The clause proved true and the code ran. However, if it was different even slightly, like let's say greet used an uppercase H instead of lowercase, the clause would be false. Hello does not equal hello.

Here is a list of operators:

- >      The greater than, used usually to compare numbers. Eg:  $5 > 4 = \text{TRUE}$
  
- <      The less than.      Eg:     $4 < 5 = \text{TRUE}$
  
- >=    The greater than or equal to      Eg:     $4 >= 5 = \text{FALSE}$
  
- <=    The less than or equal to      Eg:     $3 <= 5 = \text{TRUE}$
  
- ==    The equal to, we don't use =, because that is for assigning values. == is for comparing values.      Eg:     $5 == 10 = \text{FALSE}$
  
- !=    The not equal to      Eg:     $5 != 10 = \text{TRUE}$

And there you have the operators. You should also know that operators return boolean values, which if statements are based on. This means that you could use a boolean as a clause instead of an operator.

```
bool_test = True

if bool_test:
    print("It works!")

It works!
```

Another useful reason to use boolean variables.



## Chapter 5 – Quiz

1. \_\_\_\_ contain \_\_\_\_
  - A. Operators, clauses
  - B. Clauses, if statements
  - C. If statements, if statements
  - D. Clauses, operators
2. If statements are used to \_\_\_\_
  - A. Create conditional based code
  - B. Repeat code
  - C. Output data
  - D. Ask questions
3.  $56 \geq 55$ 
  - A. True
  - B. False

ANSWERS:

1) D 2) A 3) A

# Chapter 6 – The While Loop

Like an if statement a while loop is used to contain code based on a clause. As long as the clause is true, the code will be repeated till the clause proves false. As such it is always smart to create a while loop that will eventually prove false, otherwise a programmers must deal with an infinite loop.

A basic way to do this is to base the clause on a number value, increasing that number value within the loop till the number value reaches a certain point. It would look as follows:

```
num = 0

while num < 5:
    print(num)
    num = num + 1

0
1
2
3
4
```

NOTE: LIKE IF STATEMENTS YOU MUST INDENT YOUR CODE IN THE LOOP

Of course, since the while loop is based on a boolean value like an if statement you can use a boolean variable or base your loop on a users input. As such you can create a loop that can run an uncertain amount of times.

These kind of loops are used to perform special tasks that could save you the programmer plenty of time coding. Loops come in many forms, while loops being the most common as they have an array of uses compared to other loops.

## Chapter 6 – Quiz

1. While loops can run \_\_\_\_ times

- A) 5
- B) 10
- C) 9,000,000
- D) Infinite

2. While loops are \_\_\_\_

- A) Obsolete
- B) Common
- C) Often encountering errors
- D) Unstable

3. While loops are \_\_\_\_ if statements

- A) Similar to
- B) Unlike
- C) The same as
- D) The opposite of

ANSWERS:

1) D 2) B 3) A

# **Chapter 7 – Errors**

You will encounter three different types of errors in Python programming. Syntax, logical and exceptions.

## **Syntax Errors**

Syntax errors are common errors that will arise from a character out of place or perhaps from misspellings. Syntax errors only occur in this form and not through your actual code structure. As such these errors are the easiest to solve.

## **Logic Errors**

Logic errors come from a structure in your code. Unlike syntax errors these errors are hard to find, making them one of the worst errors that you can encounter in your Python programming. These errors are often solved using debuggers.

## **Exceptions**

Exceptions come from Python being able to decipher the code, but unable to run it due to more hidden reasons beyond the programming. For example, trying to access a file that isn't there or to access the internet with no internet connection. These are the common forms of exception error.

## Chapter 7 – Quiz

1. Identify the error: Attempting to divide a variable, but it contains no value.

- A) Syntax
- B) Logical
- C) Exception
- D) No error

2. Identify the error: Reading a text file, but misspelling in file name

- A) Syntax
- B) Logical
- C) Exception
- D) No Error

3. Identify the error: `print("Hello World")`

- A) Syntax
- B) Logical
- C) Exception
- D) No error

ANSWERS:

1) B 2) C 3) A

# **Conclusion**

Congratulations! By completing this work book you can count yourself amongst the novice Python programmers and are ready to take on your basic practical studies with a head start. Programming isn't difficult with these pieces of knowledge in mind, and if you made it this far with barely any struggle I can say with certainty that you have what it takes to become a programmer.

Learning the basics may seem tedious, even boring to some, but in the end what you learn can help construct the most interesting and useful programs imagined. Keep this all in mind and you will soar into the future with your programming know-how.

If you are curious where to go from here I recommend taking my basic Python course for beginners. In this course we escape theory, install some software and learn real programming from the ground up. By the end of the course you will have a firm foundation and can count yourself as an average programmer, ready to tackle the advances in the Python language.

If this sounds good to you visit our site, [www.programmingtut.com](http://www.programmingtut.com), and receive the course at 25% off, saving you \$10!

I hope you found this free course enjoyable and informative and feel free to use this workbook as a handy cheat-sheet in your future studies.

Kind regards

Matthew Dewey, lead programming instructor at Programming Tut