**Python Revision Book**

|  |
| --- |
| **Name:** Mohammed Mahin Ibnay Mamun |

### Tracker – How am I doing?

|  |  |  |
| --- | --- | --- |
| **Python Topics** | **Red, Amber, or Green?** | **How can I improve?** |
| **Variables** |  |  |
| **Data Type** |  |  |
| If Statements |  |  |
| **For Loops** |  |  |
| **While Loops & Counters** |  |  |
| **Arrays** |  |  |
| **Lists** |  |  |
| **Sub-programs** |  |  |
| **Object oriented programming** |  | Do more revision and do practice questions |
| **Procedural programming** |  | Do more revision and do practice questions |
| **Event driven programming** |  | Do more revision and do practice questions |

How to do well in Python

To learn Python online visit one of these websites: -

[www.learnpython.org](http://www.learnpython.org/)

[schoolcoders.com/wiki/Python](http://schoolcoders.com/wiki/Python)

w3schools

or try the **Solo Learn** app on your phone or tablet

You can also use lesson slides/ resources and your own notes.

# Knowledge Organizer

|  |  |
| --- | --- |
| **Sequence** | Two or more lines of code that are executed in order, top to bottom. |
| **Selection** | When the code makes a choice, with an "if" statement |
| **Iteration** | Code that repeats, also called a loop. We use "for" and "while" to iterate. |
| **Function** | A piece of code you write once using **def**, then can call over and over. |
| **if** | Keyword that selects, e.g., **if age > 18:** |
| **Elif** | The Python statement that follows "if" to create another condition e.g., **Elif age > 16:** |
| **else** | A Python keyword used with "if" to say what to do otherwise, i.e., when the "if" condition is not true. |
| **for** | Python code that loops a set number of times, e.g.  for num in range (10): |
| **while** | This Python keyword starts a loop or iteration, which runs while a condition is true, e.g.  **while credit < 100:** |
| **variable** | A named storage location that stores data in a program |
| **Casting Function** | function that converts from one type to another. int (), str () and float () are all examples. |
| **int ()** | A casting function that converts a string to an integer, e.g., 14 or 999. |
| **float ()** | A casting function that converts strings to "floating point" or decimal numbers. |
| **str ()** | A casting function that converts numbers to strings e.g., "14". |
| **data type** | The type of data stored in a variable. **Integer**, **float**, **string**, **boolean** and **list** are all Python data types. |
| **Pseudo-code** | A precise way of planning a program in words. It is "mock code", easier to write than Python but still precise. |
| **comment** | A note in a program beginning with # which does not run but explains what the code does. |
| **logic error** | When the program runs but does something unexpected because your code is wrong e.g., subtracting 1 from the score instead of adding 1. |
| **syntax error** | The code does not follow the rules of the language, e.g., missing punctuation in **print (“hello")** |
| **list** | A Python data structure that stores a set of values e.g., **players = ["Kane", "Maguire", "Dier", "Pickford"]** |
| **index** | The number that represents the position of an item in a list., e.g., the number 1 in **"players [1]"**. |
| **random** | A **library module** that contains **functions** including the random integer generator **randint().** |
| **append ()** | A function that adds a value to the end of a list: **players.append("Tripper")** |

**Procedural Programming:** Instructions are given in a sequence. So, Selection decides ways a program will run, and iteration is how many times code repeats.

**Object Orientated Programming:** solutions consist of objects that interact with each other. Objects defined by classes.

**Event Driven Programming:** Program is driven by responding to events.

|  |
| --- |
| 1. What is **integer, string,** and **Boolean** examples of?   Integer – any whole  String – A sequence of character  Boolean – true/false yes/no zero/one |
| 1. What is a piece of code that you write once using the “def” keyword, then call over and over?   subprogram |
| 1. What is a **casting function** that converts strings to floating points?     Converting one data type to another |
| 1. What is the number that represents the position of an item in a list., e.g., the number 1 in "players [1]"? |
| 1. What Python keywords select? |
| 1. What do we call a Python function that converts data from one type to another? int (), str () and float () are all examples?   casting |
| 1. A note in a program beginning with # which does not run but explains what the code does. → comment. **True/False.**     True |
| 1. Look at this flowchart.  Write the Python code for this algorithm here: |
| 1. Look at this **list**:  players = ["Kane", "Maguire", "Dier", "Pickford"]      * 1. Write the Python code to display "Dier"   2 because code starts at 0 and Dier is the 3rd value but in code it is the second due to 0     * 1. Write some code to add "Lallana" to the list.       Players.append(Lallana) |
| 1. Look at this code. Explain it below. Label the **function definition, function calls, iteration, selection, inputs, and outputs.**     u  Defines a subroutine called message.  Variable called message is created which iterates in range 10 times. Prints out the variable message.  Defines a subroutine called timetable. Creates a variable which is called num and is ran 10 times in a for loop. Output will print out the timetable therefore if you type 4 for the variable, the code will run this. 4 x 1, 4 x 2, 4 x 3. |

# Vocabulary Boost

|  |  |
| --- | --- |
| Step 1: Complete the definitions | |
| Sequence | Two or more lines of code that are executed in order, top to bottom. |
| Selection | When the code makes a choice with an “if” statement. |
| iteration | A piece of code that repeats, also called a loop. We make code loop with “for” and “while” |
| Function | A piece of code you write once using the “def” keyword, then can call over and over. |
| If | A Python keyword that makes a selection or choice |
| Elif | The Python statement that follows “if” to create another condition e.g., Elif age > 16: |
| Else | A Python keyword used with “if” to say what to do otherwise |
| For | Python code that makes a loop repeat a set number of times |
| while | This Python keyword starts a loop or iteration, which runs while a condition is true |
| Logic error | When the program runs but does something unexpected because your code is wrong |
| Syntax  error | The code does not follow the rules of the language |
| Variable | A named storage location that stores data in a program |
| list | A Python data structure that stores a set of values |
| str () | A casting function that converts numbers to strings e.g., "14". |
| Iteration | Code that repeats also called a loop |
| pseudocode | A precise way of planning a program with English words.    It is “mock code,” easier to write than Python but still precise. |
| Step 2: Powerful Knowledge | |
| Read the following paragraph. | |
| A test plan is designed to ensure that the coded solution works as expected. The test plan will include all the inputs the program will be tested with and expected outputs.    It is important to test all **branches** of the program, so your test data must include many inputs.    Test plans are used to find **logic errors** where the code is **syntactically** correct but does not behave as designed, and **runtime errors**. *Division by zero* is a runtime error that may not be obvious when coding.    Test plans might be written by one programmer but given to another programmer who will then test the code. For this reason, the code should be well-written and readable, with indentation, whitespace, meaningful variable names, and comments. | |
| Now answer these questions: | |
| 1. Why do we write a test plan?   So, the coded solution works as expected | |
| 1. What needs to be included in a test plan?   All the inputs and outputs | |
| 1. Why can the IDE not find logic errors?   Test plans are used to find logic errors where the code is syntactically correct but does not behave as designed | |
| 1. Why is it important to make your code readable?   Because the code will be sent to other programmers to check and test so they should be able to read it. It is good to add comments using # so the user can clearly understand. | |
| 1. Explain why these features make your code more readable? 2. Indentation   Shows which part of the code belongs to which area. For example, loops use indent to show the code is within the loop     1. Comments   Makes it easier because you can type anything without errors   1. Whitespace   Just makes the code clearer   1. Meaningful variable names   Easier to interpret and you can refer to it knows what it is. | |

# Python Revision: Python Programing

These programs use different operators. Remember the operators are as follows:

== is equal to

! = is not equal to   
 > is greater than

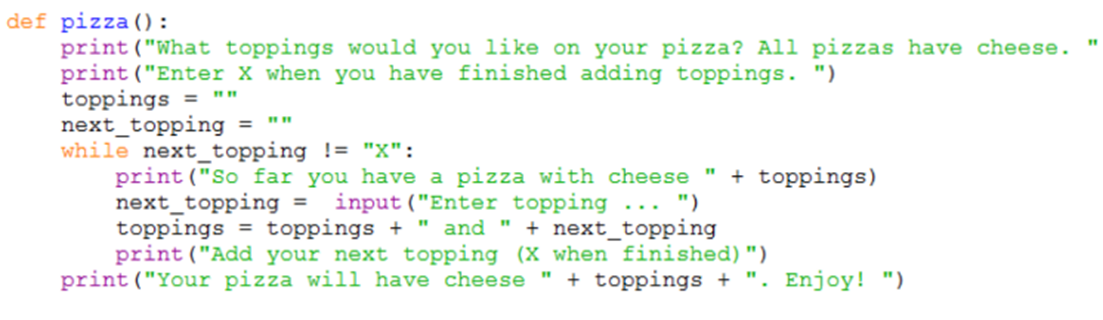
< is less than   
>= is greater than or equal to   
<= is less than or equal to

|  |  |
| --- | --- |
| 1. Write a program that writes “Today is Monday” one hundred times using a while loop. Compare this to the way that we might have done this last week with a for loop. | …… and so on …. |
| 1. Write a program that asks the user for the name of the recent storm and keeps asking until they put in the name “Doris.” To the right is some code for the input statement to get you started |  |
| 1. Write a program that asks the user for a number until they put in 99. When they put in 99 the program stops. |  |
| 1. Write a function bored () using a while loop that has the output shown to the right.   Start with |  |
| 1. Write a program that prints 1 to 10. You can do this with a while loop or a for loop. Try to do it both ways. |  |
| 1. Now change your program so that it prints the numbers from 10-1. |  |

**Key concepts to remember**

|  |  |  |
| --- | --- | --- |
| Term | What it means | How to do it in Python |
| Iteration | A loop. There are two loops – for loops (a set number of repetitions) and conditional loops or while loops. |  |
| Variable | Something you can give value to and then change it at other times in the program |  |
| Selection | Where there is a choice point in the program design and an if statement is used to create more than one pathway. |  |
| Input/Output | Getting input from the keyboard or outputting something to the screen. |  |
| Assignment | Where a variable is given a value |  |

Stretch question: Explain what this code does.



Line 1) Defines a subprogram called pizza.

Line 2) outputs statement “What toppings would you like on your pizza? All pizza have cheese.”

Line 3) outputs statement “Enter X when you have finished adding toppings.”

Line 4) Creates an empty variable, list called toppings

Line 5) Creates an empty variable, list called next\_topping

Line 6) creates a while loop which will run until variable next\_topping is not equal to “x”

Line 7) prints out code “So far you have a pizza with cheese” and then it adds all the other options which the user inputted in line 4, values stored in varibale toppings

Line 8) variable next\_topping allows an input for the user to type.

Line 9) variable toppings get reassigned by topping “and” next\_topping

Line 10) outputs message “Add your next topping (x when done”

Line 11) code prints out outside of the while loop. The message outputted will say “Your pizza will have cheese” and then all the toppings which was inputted then it finishes with “Enjoy! ”