

Box A—Key Words

Annotation: Commenting your code to explain what parts do.

Assignment: Storing a value (numerical or otherwise) to a variable.

Data Structure: A way of storing multiple bits of information at once.

Index: A position reference in a data structure.

Iteration: Repeating sections of codes using loops, usually with changing values each time.

List: Also known as an array. A data structure.

Module: A collection of linked functions. External modules can be imported in to your code.

Selection: Using code to choose what happens in a program. Also called a conditional.

User Input: Allowing the person using the code to provide data.

Variable: A value that can change. Used to store information for use in a code.

Basic Python Programming

Box B—Key Skills

Understanding Python Code:

```
name = input("Enter Name") #A
age = 14 #B
users = ["John", "Jane"] #C
length = len(users) #D
valid = False #E

for i in range(length): #F
    if name == users[i]: #G
        valid = True #H

if valid == True: #I
    print("Valid user") #J
else: #K
    print("Invalid user")
```

A: Stores user inputted text to a variable called name.

B: Stores an integer value 14 to a variable called age.

C: Creates a list which contains 2 string values. Stores to users.

D: Calculates the length (how many items are in) of users.

E: Stores the Boolean value False to variable valid.

F: Use of white space to make code clearer.

G: Creates a loop that will iterate for every element in users.

H: Use of selection to determine if the entered value match with any values from the list. users[i] looks up the current value from the list based on the given index.

I: If the name is in the list users, valid is changed to True.

J: Selection based on whether valid was changed to True.

K: Runs alternative code if the criteria from J is not met.

Box C—Key Knowledge

Key Syntax:

= used for assignment	== used for comparison
-----------------------	------------------------

Python functions (such as print or len) must have brackets after them, which may contain information. E.g. print("Hello") or exit()

Keywords in lower-case only e.g. if, while	# used to start a comment
--	---------------------------

: must be used at the end of selection and iteration

Handling Data Types:

Convert to integer `int(x)`

Convert to float `float(x)`

Convert to string `str(x)`

Convert to Boolean `bool(x)`

User input in python is `str` string by default.

Variable Name Conventions:

Should not be overly long but should be easy to identify purpose. Must start with a letter.

Must not contain any symbols or spaces. Instead of a space, use an underscore or camelCase, e.g.:

user name ❌ user_name ✅ userName ✅

Box D—Key Words

Append: Add or attach.

Callback: Running a function.

Function: A defined section of code to carry out a certain process. Useful where code may be reused and to improve structure.

GUI: Graphical user interface. The visual elements of a program that allow the user to interact with the program.

Import: Bringing in an external resource. In python, this means loading external functions ready to be used.

Parameter: The information required for a function to run.

Scope: (of variables). The domain which a variable is valid. For example, a variable defined in a function will be local, only usable by that function.

Widget: A component of a user interface.

Advanced Programming

Box E—Key Skills

Using Functions

```
def exampleFunction(a, b):  
    if a > b:  
        print(a, "is bigger")  
        return a  
    elif a < b:  
        print(b, "is bigger")  
        return b  
    else:  
        print("They are the same")  
        return a
```

Short for define → `def`
Name of function → `exampleFunction`
Parameters → `(a, b)`

```
num = 9  
exampleFunction(5, num)
```

return turns the function in to the returned value. E.g. `exampleFunction(5,9)` will become the value 9.

Calls the function, sending the values 5 and 9 as the parameters.

Using External Files:

```
openFile = open("name.txt", "r")  
readFile = openFile.readlines()  
print(readFile)  
  
writeFile = open("new.txt", "w")  
writeFile.write("Example")  
writeFile.close()
```

Opens a file called `name.txt` and reads it. `readlines` stores each line of the file in a list.

`writeFile.write("Example")` will store the contents of the brackets to the file.

Must close the file when done or it won't update.

The "w" opens the file in write mode, which will create a new file of the given name.

Box F—Key Knowledge

Random Module:

The random module contains functions allowing random features within your code.

Key Functions:

`randint(a, b)` #generates a random integer between a and b.

`choice(LIST)` #chooses a random element from LIST.

`LIST.shuffle()` #randomises the elements in LIST.

`sample(LIST, n)` #returns n random items from LIST

Tkinter:

Tkinter is a GUI for python, allowing professional looking menus and interactivity. It contains many widgets for different functionality. Widgets need to be placed using either grid or pack geometry manager. E.g. `widget.grid()`

Key Functions:

`Tk()` #loads tkinter to the program and creates a root window.

`Label()` #Widget used to store text.

`Button()` #Interactive widget which can call a function when clicked.

`Entry()` #Textbox to allow user input.