**The Python Library**

The Python library contains built-in modules that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well modules provide standardised solutions for many problems that occur in everyday programming.

<http://docs.python.org/2/library/>

Python calls libraries modules, to use commands from a module, the module needs to be imported using the keyword import.

The **datetime** library is used for working with dates and times.

|  |
| --- |
| **import datetime**  **#Find the current date and time.**  **print(datetime.datetime.now())# Formatted YYY-MM-DD-HH-MM-SS-Microseconds**  **today = datetime.date.today()**  **print(today) # Formatted YYY-MM-DD** |

The **strftime** method is used with the **datetime** library to format the date time. The **strftime** method has a number of directives; a few are illustrated below.

|  |
| --- |
| **print(datetime.date.today().strftime("%Y")) # Year with century as decimal.**  **print(datetime.date.today().strftime("%y"))# Year no century as decimal.**  **print(datetime.date.today().strftime("%b")) # Abbreviated month name.**  **print(datetime.date.today().strftime("%B")) # Full month name.**  **print(datetime.date.today().strftime("%w")) # Weekday as decimal.**  **print(datetime.date.today().strftime("%W"))# Week year number as decimal.**  **print(datetime.date.today().strftime("%j"))# Day of year as decimal.**  **print(datetime.date.today().strftime("%d"))# Day of month as decimal.**  **print(datetime.date.today().strftime("%a")) # Abbreviated weekday name.**  **print (datetime.date.today().strftime("%A")) # Full weekday name.**  **print(datetime.datetime.now().strftime("%H")) # Hour as decimal (24 hour).**  **print(datetime.datetime.now().strftime("%I")) # Hour as decimal (12 hour).**  **print(datetime.datetime.now().strftime("%M")) # Minute as decimal.**  **print(datetime.datetime.now().strftime("%p")) # AM or PM.** |

*Exercise:*

|  |  |
| --- | --- |
| Format today’s date and time as:  DD-MM-YYYY HH:MM AM/PM | Format today’s date as:  DD-MMM-YYYY |
| **import datetime**  **dateToday = datetime.datetime.now()**  **print (dateToday.strftime("%d-%m-%Y %I:%M %p "))** | **import datetime**  **dateToday = datetime.date.today()**  **print (dateToday.strftime("%d-%b-%Y"))** |

*Determine the output of the following statements:*

|  |
| --- |
| **#Statement 1**  **import datetime**  **myDate = datetime.datetime(2014, 12, 31, 23, 59, 59)**  **print (myDate)** |
| **#Statement 2**  **import datetime**  **print (datetime.datetime.now().strftime("Today is %A the %dth of %b %Y"))** |

*Calculate the difference between 2 dates:*

|  |
| --- |
| **import datetime**  **print (datetime.date(2014, 5, 18) - datetime.date(2014, 4, 18))** |

The **random** library can be used to generate a random number, or select a random element from a sequence.

|  |
| --- |
| The **random()**methodreturn a random floating point number between the range of >=0.0 and <1.0.  *Try it:*  **import random**  **print (random.random())**  The **randrange(start, stop, step)**methodreturn a randomly selected element from range ().  **print (random.randrange(5)) # Returns an integer <5**  **print (random.randrange(5,10)) # Returns an integer between >=5 and <10**  **print (random.randrange(100, 200, 2)) # Returns an even integer between >=100 and <200**  The **randint(start, stop)** methodreturn a random integer between and inclusive start value and stop value.  **print (random.randint(0, 10)) # Returns an integer between >=0 and <=10**  The **choice(sequence)** methodreturns a random choice from a sequence (string, list).  **print(random.choice('Python Programming'))**  **print(random.choice([1, 3, 8, 9, 20]))**  The **sample(sequence, amount)** methodreturns a new list with a given amount of elements from a given sequence.  **print(random.sample([1, 2, 3, 4, 5], 2) ) # Returns a new list containing 2 elements from the given sequence.**  **print(random.sample('Python Programming', 3)) # Returns a new list containing 3 elements from the given sequence.**  The **shuffle(list)** methodshuffles the elements in list.  **list = [1, 2, 3, 4, 5]**  **random.shuffle(list) # Shuffle the given list.**  **print (list)**  The **uniform(start, stop)** method returns a floating point number between two values.  **print (random.uniform(5, 10))** |

The Python import statement also allows you to import your own module, see the example below.

*Create the following modules, and attempt to explain what is happening.*

|  |  |
| --- | --- |
| *Create a folder on your usb drive name it “pythonWeek03”.*  *Within this folder create yet another folder and name it “myLibrary”.*  *Create the following program and name it “loopExamples”.*  *Place this program into the “myLibrary” folder.* | *Create the following program and name it “mainProgram”.*  *Place “mainProgram” into the “pythonWeek03” folder.* |
| **# loopExamples.py**  **def example01():**  **for x in range(5):**  **print ('Number', x)**  **def example02():**  **num = 0**    **while (num < 3):**  **num = num + 1**  **print('loop',num)**    **print('Out of the loop!')** | **# mainProgram.py**  **def main():**  **import sys**  **sys.path.append('myLibrary')**  **import loopExamples**  **loopExamples.example01()**  **loopExamples.example02()**  **if \_\_name\_\_ == "\_\_main\_\_":**  **main()** |

*Explanation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

**EasyGUI**

EasyGUI is a simple Python GUI module commonly used to obtain information from the user using dialog boxes. Before we can use the EasyGUI module we need to import it. To invoke the EasyGUI module simply type **import easygui**.

EasyGUI Functions

**msgbox(message, title)** – This function is used to display a message.

|  |  |
| --- | --- |
| **import easygui**  **easygui.msgbox(msg="Hello World!", title="My GUI", ok\_button="OK")**  ***alternative:***  **easygui.msgbox("Hello World!","My GUI")** |  |

The EasyGUI module can use an alias as demonstrated below:

|  |
| --- |
| **import easygui as eg**  **eg.msgbox("Hello World!","My GUI")** |

**ccbox(message, title, choices)** – This function presents the user with a choice to “Continue” or “Cancel”, the default is continue.

|  |  |
| --- | --- |
| **msg="Continue?"**  **title="My GUI"**  **eg.ccbox(msg, title)**  ***alternative:***  **eg.ccbox(msg="Continue?", title="My GUI", choices=("Continue", "Cancel"), image=None)**  ***alternative:***  **eg.ccbox("Continue?", "Michael's GUI")** |  |

The return value is a 1 or 0, if the first choice (“Continue”) is selected or the dialog box is cancelled it returns 1, else it returns 0. Try it:

|  |
| --- |
| **decision=eg.ccbox("Continue?", "Michael's GUI")**  **print(decision)** |

**ynbox(message, title, choices)** – This function present the user with a “Yes” and “No” option. It returns either 1 (yes) or 0 (no).

|  |  |
| --- | --- |
| **msg="Are you a programmer?"**  **title="Question 1"**  **choices=("Yes","No")**  **eg.ynbox(msg, title,** **choices)**  ***alternative:***  **eg.ynbox(msg='Are you a programmer?', title='Question 1')** |  |

**boolbox(message, title, choices, image)** This function is similar to the **ynbox()** function.

|  |  |
| --- | --- |
| **eg.boolbox(msg='Are you a programmer?', title='Question 1')** |  |

**buttonbox(message, title, choices)** – The function presents the user with a list of choices. It returns the option selected.

|  |  |
| --- | --- |
| **msg="What is the square root of 64?"**  **title="Question 2",**  **choices=("6", "8", "10")**  **ans=eg.buttonbox(msg, title, choices)**  **print(ans)**  ***alternative:***  **ans=eg.buttonbox(msg="What is the square root of 64?",**  **title="Question 2",**  **choices=("6", "8", "10"))**  ***alternative:***  **ans=eg.buttonbox("What is the square root of 64?", "Question 2",("6", "8", "10"))** |  |

**enterbox(message, title, default)** – The function returns the entered string. If the cancel button presses **none** is returned.

|  |  |
| --- | --- |
| **text = eg.enterbox(msg="Enter a string", title="My GUI", default="")**  **print(text)** |  |

**indexbox(message, title, choices)**– this function returns the index of the selected choice. In the example below 10000000 has index 0, 10000001 has index 2.

|  |  |
| --- | --- |
| **Msg="129 in binary is?",**  **title="Question 3",**  **choices=("10000000", "10000101", "10000001")**  **ans=eg.indexbox(msg,title,choices)**  ***alternative:***  **ans=eg.indexbox("129 in binary is?", "Question 3", ("10000000", "10000101", "10000001"))**  **print(ans)** |  |

**choicebox(message, title, choices)** – This function provides a way to select an item from a list of choices, if the cancel button presses **none** is returned.

|  |  |
| --- | --- |
| **msg = "Select correct answer?"**  **title = "Question 3"**  **choices = ("abc", "def", "ghi")**  **answer = eg.choicebox(msg, title, choices)**  **print(answer)** |  |

**multchoicebox(message, title, choices)** – This function provides a way to select multiple items from a list of choices, if the cancel button presses **none** is returned.

|  |  |
| --- | --- |
| **msg = "Select all correct answers"**  **title = "Question 4"**  **choices = ["abc", "def", "ghi"]**  **answer = eg.multchoicebox(msg, title, choices)**  **print(answer)** |  |

**integerbox(message, title, default, lowerbound, upperbound)**– This function returns a selected integer within specified boundaries, if the cancel button presses **none** is returned. If user input is invalid, the integerbox function warns the user with an error message, and asks for input again.

|  |  |
| --- | --- |
| **enterNum=eg.integerbox( msg='Enter a number between 0 and 100',**  **title='My Game',**  **default='',**  **lowerbound=0, upperbound=100)**  **print(enterNum)** |  |

**diropenbox(message, title, default)** – This function returns the path of a selected directory, or **none** if user presses cancel.

|  |  |
| --- | --- |
| **dr=eg.diropenbox(msg="Choose a directory", title="Directory Selector", default="C:/")**  **print(dr)** |  |

**fileopenbox(message, title, default)** – This function returns the path of a selected file, or **none** if the user presses cancel.

|  |  |
| --- | --- |
| **file=eg.fileopenbox(msg="Choose a file", title="File Selector", default="C:/")**  **print(file)**  ***alternative:***  **file=eg.fileopenbox('Select a file')**  **print(file)** |  |

**eg.filesavebox(message. Title, default)** – This function returns the name of a file to save (user entered file name) or **none** if the user presses cancel.

|  |  |
| --- | --- |
| **eg.filesavebox(msg=None, title=None, default="C:/")** |  |

**passwordbox(message, title, default)** – This function returns a password value, or **none** if user presses cancel. When the password is entered, it is masked as asterisks (\*\*\*).

|  |  |
| --- | --- |
| **psw = eg.passwordbox(msg='Enter your password.', title='Login', default='')** |  |

**multenterbox(message, title, fields, values)** – This function returns a list of multiple values or **none** if user presses cancel.

|  |  |
| --- | --- |
| **msg = "Enter your Details"**  **title = "Online Registration"**  **fields = ["Name","Address","Suburb","State","Postcode"]**  **values = []**  **values = eg.multenterbox(msg,title, fields)**  **print(values)** |  |

**multpasswordbox(message, title, fields, values)** – This function is similar to the **multenterbox()**function, with the exception that the last field is treated as a password.

|  |  |
| --- | --- |
| **msg = "Enter logon details"**  **title = "Login"**  **fields = ["Department ID", "Employee ID", "Password"]**  **values = []**  **values =**  **eg.multpasswordbox(msg,title, fields)** |  |

**textbox(message, title, text)** – This function creates a textbox used to accept user input. The text can either be a string or a list of strings (i.e., the result of file.read()).

|  |  |
| --- | --- |
| **# Example 1**  **answer = eg.textbox(msg="Please enter your name",**  **title="my Gui",**  **text="")**  **print(answer)**  **# Example 2**  **answer = eg.textbox(msg="Please enter your name",**  **title="my Gui",**  **text="Your name is ")**  **print(answer)**  **# Example 3**  **file = open ("TenGreenBottles.txt", "r")**  **text = file.read()**  **file.close()**  **eg.textbox(msg="The words to Ten Green Bottles", title="My GUI", text=text)**  **# Example 4**  **def readFile(filename):**  **file = open (filename, 'r')**  **text = file.read()**  **file.close()**  **return text**  **text = readFile("TenGreenBottles.txt")**  **eg.textbox("The words to Ten Green Bottles", "My GUI",text)**  **# Example 5**  **myList= ['Ww', 'Aa', 'Bb', 'Cc']**  **myList = ' '.join(myList)**  **eg.textbox('','',myList)** |  |

**Import**

The Python import statement allows the code in one module to access the code of another module.

*Create the following programs, and attempt to explain what is happening.*

|  |  |
| --- | --- |
| *Create a folder on your usb drive name it “pythonWeek03”.*  *Within this folder create yet another folder and name it “myLibrary”.*  *Create the following program and name it “loopExamples”.*  *Place this program into the “myLibrary” folder.* | *Create the following program and name it “mainProgram”.*  *Place “mainProgram” into the “pythonWeek03” folder.* |
| **# loopExamples.py**  **def example01():**  **for x in range(5):**  **print ('Number', x)**  **def example02():**  **num = 0**    **while (num < 3):**  **num = num + 1**  **print('loop',num)**    **print('Out of the loop!')** | **# mainProgram.py**  **def main():**  **import sys**  **sys.path.append('myLibrary')**  **import loopExamples**  **loopExamples.example01()**  **loopExamples.example02()**  **if \_\_name\_\_ == "\_\_main\_\_":**  **main()** |

*Explanation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*