**Introduction to Python – Practice Exercises**

**Producing output to the screen**

1. Using two print statements, output the following to the screen:

Introduction to

Python Programming

Solution

print(**"Introduction to"**print(**"Python Programming"**)

1. Produce the same output but only using one print statement:

Solution

print(**"Introduction to\nPython Programming"**)

**Using variables and formatting output**

1. Assign a variable called my\_age the value equal to your own age and produce output in the following format:

My age is 21

Solution

my\_age = 21

print(**"My age is",my\_age**)

1. Assign two variables called my\_name and my\_age containing your name and age, add one to your age and produce output in the following format:

Next year John will be 22 years old

Solution

my\_name = **"John"**my\_age = 21  
my\_age = my\_age + 1  
print(**"Next year"**,my\_name,**"will be"**,my\_age,**"years old"**)

**Accepting input from the user**

1. Accept input from the user of their name and place of birth and produce output in the following format:

John was born in Salisbury

Solution

name = input(**"Enter your name: "**)  
place\_of\_birth = input(**"Enter your place of birth: "**)  
print(name,**"was born in"**,place\_of\_birth)

1. Prompt the user to enter three integer numbers, calculate the average and produce the following output in the following format (with the average rounded to 2 decimal places):

The average of 10, 20 and 8 is 12.67

Solution

num1 = int(input(**"Enter first number: "**))  
num2 = int(input(**"Enter second number: "**))  
num3 = int(input(**"Enter third number: "**))  
avg = (num1 + num2 + num3) / 3  
print(**"The average of {0}, {1} and {2} is {3:.2f}"**.format(num1,num2,num3,avg))

**Decisions and conditions**

1. Prompt the user to enter an integer number and calculate if it is an odd or an even number. Produce output in the following format:

3 is an odd number

Solution

*# Check if the input number is odd or even.  
# A number is even if division by 2 gives a remainder of 0.  
# If the remainder is 1, it is an odd number.*num = int(input(**"Enter a number: "**))  
**if** (num % 2) == 0:  
 print(**"{0} is an even number"**.format(num))  
**else**:  
 print(**"{0} is an odd number"**.format(num))

1. Prompt the user to enter a % in-course assessment and a % exam mark, calculate and print out the final grade according to the following grading scheme:

Final Grade

Under 40% in either the in-course assessment or exam Fail

Between 40-79% in both in-course assessment and exam Pass

Over 80% in either the in-course assessment or exam Distinction

If the user enters a value greater than 100 for either mark, output an error message.

Solution

incourse\_mark = int(input(**"Enter in-course assessment mark between 0 and 100: "**))  
exam\_mark = int(input(**"Enter exam mark between 0 and 100: "**))  
**if** incourse\_mark < 40 **or** exam\_mark < 40:  
 print(**"You have failed"**)  
**if** incourse\_mark>=40 **and** incourse\_mark<80 **and** exam\_mark>=40 **and** exam\_mark<80:  
 print(**"You have passed"**)  
**if** (incourse\_mark>=80 **and** incourse\_mark<=100) **or** (exam\_mark>=80 **and** exam\_mark<= 100):  
 print(**"You have passed with distinction"**)  
**if** incourse\_mark > 100:  
 print(**"In-course assessment mark casnnot be greater than 100"**)  
**if** exam\_mark > 100:  
 print(**"Exam mark cannot be greater than 100"**)

**Loops**

1. Prompt the user to enter a word and print the word vertically one letter at a time each on a different line. For instance, if the user enters the word DATABASE, the output will be as follows:

D

A

T

A

B

A

S

E

Solution

word = input(**"Enter a word: "**)  
**for** count **in** range(0, len(word), 1):  
 print(word[count])

1. Calculate the square of a number of each number entered by the user until the user enters a 0.

Solution

*#Calculate the square of a number*num =int(input(**"Enter a number (0 to quit): "**))  
*#loop until the user enters a 0***while** num != 0:  
 square = num \* num  
 print(**"The square of "**,num,**" is "**,square)  
 num =int(input(**"Enter a number (0 to quit): "**))

**Working with lists**

1. Initiate a list containing three menu options (‘Calculate sum’,’Calculate average’ and ‘Exit’) and then output each menu option in the list ordered 1, 2 and 3 on a separate line as follows:

Main menu

---------------

1. Calculate sum
2. Calculate average
3. Exit

Solution

*#initialise the main menu list*

main\_menu = [**"Calculate sum"**,**"Calculate average"**,**"Exit"**]  
print(**"Main menu"**.center(30))  
print(**"---------"**.center(30))  
option\_num = 1  
*#print each menu option with a number*

**for** menu\_option **in** main\_menu:  
 print(**"{0}. {1}"**.format(option\_num,menu\_option))  
 option\_num = option\_num + 1

12. Write a program to accept the input of the first three letters of a month name (e.g. Feb or Jul) and display the number of days in that month. Use lists and compare the month entered against the lists to determine the correct number of days in that month. Output should be in the following format:

MAR has 31 days

Solution

*#initiate lists*months\_28\_days = [**"FEB"**]  
months\_30\_days = [**"APR"**,**"JUN"**,**"SEP"**,**"NOV"**]  
months\_31\_days = [**"JAN"**,**"MAR"**,**"MAY"**,**"JUL"**,**"AUG"**,**"OCT"**,**"DEC"**]  
*#get user to input a month*month = input(**"Enter the first three letters of the month: "**)  
*#convert what they have entered into uppercase*month = month.upper()  
*#check month entered against each list and when found display the corresponding number of days***if** month **in** months\_28\_days:  
 print(month,**"has 28 days"**)  
**else**:  
 **if** month **in** months\_30\_days:  
 print(month,**"has 30 days"**)  
 **else**:  
 **if** month **in** months\_31\_days:  
 print(month,**"has 31 days"**)  
 **else**:  
 print(month,**"is not a valid month"**)

This can be rewritten using elif’s as follows which is easier to read than many nested if’s:

*#initiate lists*months\_28\_days = [**"FEB"**]  
months\_30\_days = [**"APR"**,**"JUN"**,**"SEP"**,**"NOV"**]  
months\_31\_days = [**"JAN"**,**"MAR"**,**"MAY"**,**"JUL"**,**"AUG"**,**"OCT"**,**"DEC"**]  
*#get user to input a month*month = input(**"Enter the first three letters of the month: "**)  
*#convert what they have entered into uppercase*month = month.upper()  
*#check month entered against each list and when found display the corresponding number of days***if** month **in** months\_28\_days:  
 print(month,**"has 28 days"**)  
**elif** month **in** months\_30\_days:  
 print(month,**"has 30 days"**)  
**elif** month **in** months\_31\_days:  
 print(month,**"has 31 days"**)  
**else**:  
 print(month,**"is not a valid month"**)

**Importing and using modules**

1. Import the random module and initialise an empty list. Generate five random integer numbers between 1 and 10 and append each one to the list. Produce output in the following format:

The random numbers generated were [2, 8, 7, 2, 5]

Solution

**import** random  
rand\_num\_list = []  
**for** i **in** range(0, 5):  
 rand\_num\_list.append(random.randint(1,10))  
print(**"The random numbers generated were"**,rand\_num\_list)

**Bringing all the elements together**

1. Accept the input of an integer number and determine if that number is a prime number or not.

Solution

*# Program to check if a number is prime or not  
# Get user to enter a number*num = int(input(**"Enter a number: "**))  
prime = **True***# prime numbers are greater than 1***if** num > 1:  
 *# check for factors* **for** i **in** range(2, num):  
 **if** (num % i) == 0:  
 print(num, **"is not a prime number"**)  
 print(i, **"times"**, num // i, **"is"**, num)  
 prime = **False  
 break  
 if** prime:  
 print(num, **"is a prime number"**)  
*# if input number <= 1, it is not prime***else**:  
 print(num, **"is not a prime number"**)