

assignment one

by David Railton

Submission date: 09-Feb-2023 09:03AM (UTC+0000)

Submission ID: 196947352

File name: 41753_David_Railton_assignment_one_466019_1726753636.docx (23.27K)

Word count: 758

Character count: 3679

David Railton assignment for
programming 2022/2023
ACCA4020 Gaynor Thomas
level 4

Variables

In Python, a Variable is a named storage area that contains a value that can be modified at a later time in the programme. Data is stored there and utilised as a reference.

In Python there are different types of Variable each with a function individual to its purpose.

An Integer represents all whole numbers from minus numbers up to infinite numbers providing they are whole.

A Float represents decimal numbers, again whether minus or whole.

Strings show sequence of characters, numbers or letters eg: "1234" or "bonkers".

Lists are ordered collections of items eg: [1, 2, 3]

Boolean shows True or False values.

If/else statements

If/else statements are used for executing code only if a certain condition is fulfilled for example:

```
print('what is your name?')
myName = input('Please enter your name: ')

if myName == 'Moriati':
    print('A 1970s classic name')
else:
    print('get a grip you are not an evil genius')
```

So, if myName = Moriati then the code will simply print out “A 1970’s classic name”.

The “else’ statement here means that if the name is anything but Moriati you will be told to “Get a grip you are not an evil genius”

An if/else statement can be expanded into a loop that will keep repeating the process until the correct response is entered or all possibilities are used up then print out an end message, end the program or even, and more annoyingly, start the process over again.

Loops in Python

Loops in Python are used to allow a repetitive execution of a block of code.

In Python there are two types of loops, 'for' loops and 'while' loops.

A 'for' loop repeats a function a certain number of times. Loops can save time as it means the same code does not need to be re-written over and over to achieve the same result.

A simple loop repeating the same function to print a sequence of numbers is:

```
for i in range(21):  
    print(i)
```

This will print out the numbers 0-20 as the upper limit of any list is not included as the count starts from zero.

```
for i in range(250, 0, -7):  
    print(i)
```

This will print the numbers from 250-0 in descending order in increments of 7 until the final number before zero is reached.

With words loops can be used through a string letter by letter.

For example:

```
word = "Chariots"  
for char in word:  
    print(char)
```

A 'while' loop in Python will repeatedly execute a section of code 'while' a set condition remains 'true'. If the condition becomes 'false' the loop will terminate and the program will move onto the next statement.

For example:

National Insurance Data

I had some simple issues with this code and I cannot understand what they are. I ran the same code on numerous occasions and sometimes it would limit it to two decimal places and on others it would run to the actual end point of the “equation”

This is a simple code to establish a persons name, national insurance number and to work out their monthly wage. Each of these, myAge, myWage and NationalInsuranceNumber have variables. The code can be refined to having it only accepting of one user’s details by inserting a “while loop” having the program return each time a false entry is made on the name assuming the name is a set parameter.

It could be further refined to recognise the two letter, six number, one letter combo of a National Insurance number.

I put a friendly touch to the code.

```

name = input('please enter your name')

NationalInsuranceNumber = input('enter your national insurance number')

myAge = int(input('please enter your age'))
while myAge <= 18:
    print('you are way too young')
    myAge = int(input('please enter your age'))
if myAge >= 129:
    print('wow you look great for a mummy')
myWage = int(input('please enter your wage'))
if myWage >= 40500:
    print('that seems very high for your age')
    print('then your monthly wage would be')
    monthlyWage = str(input(myWage/12))
    """print('your monthly wage is ' + int(monthlyWage))"""
if myWage < 40500:
    print('that seems acceptable')
    print('then your monthly wage would be')
    monthlyWage = int(input(myWage/12))
    """print(monthlyWage)"""
    print('your monthly wage is ' + int(monthlyWage))

```

Square circle

With this code I will use the math function of Python to work out the remaining are of a circle after a square of variable parameters (length of side) is removed. I will also try to find a formula to cause a loop if the side of the square is too large to accommodate the length of the square.

assignment one

GRADEMARK REPORT

FINAL GRADE

22/100

GENERAL COMMENTS

Instructor

PART A

A1 Variables - Generally good, need to mention dynamic typing 4/5

A2 If/Else - A reasonable statement could have also talked about multiple conditions and if/elif/else 2.5/5

A3 Loops - Good, but no code example of while loop 4/5

PART B

B1 Variables

Logical Stages/Testing - 2.5/5 (No testing shown)

Code - Generally OK - some of the validation needs to be while instead of if (in case the user enters the wrong value on the second try) 8/10

B2 Shape

Logical Stages/Testing - 1 /5

Code - /15 None presented

B3 Lists

Logical Stages/Testing - /10 None presented

Code - /20 None presented

B4 Turtle

Code - /20 None presented

General Comments

The type is very large throughout. 11pt is usually sufficient.

OVERALL TOTAL FOR MODULE: 22% (Many parts of assignment are missing)

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
