Basic Techniques of Programming Part 1: Programming for Data Science with Python

Overview

To write a useful program, the developer needs to use various techniques of programming. Basic Techniques of Programming Part 1 and 2 cover the techniques below.

Some of these techniques are considered as the core of programming to create interactive software applications:

- Input and output
- Selections
- Loops

---

1. Read Data from Console

Scenario

Write a short Python program that reads an integer value from the console and then print out the value.

IMPORTANT NOTES: It is assumed that the user would not make any mistake while entering the value. Therefore, it is not necessary to check the input after reading it.

Syntax

To read data from the console in Python, use the built-in function:

- input(prompt_string)

Where prompt_string is the text used to prompt the user to enter the data.

IMPORTANT NOTES: Need to declare a new variable to store the text read from the console.

Run the following code:

In [3]:
```
anIntValue = input("Enter an interger value:")
print ("The user has entered this value: ", anIntValue)
```

```
Enter an interger value:47
The user has entered this value:  47
```

---

2. Print Data to the Console

- To print data to the console, use the built-in function:

  > print(a_string)

- Where a string can be only one string, or multiple sub-strings and values seperated by commas',':

- To print the values in one line, using the end-of-line character'\n':

  > print (...,'\n')

Run the following code:

In [8]:
```
print ("Examples of using print() function", "\n")

x = 15

print ("This is the value of x:", x, "\n")

y = 25

print ("This is the value of x:", x, "; This is the value of y:", y, ". \n")
```

```
Examples of using print() function

This is the value of x: 15

This is the value of x: 15 ; This is the value of y: 25 .
```

3. Selections

Scenario: A Problem

- Let's review the problem of calculating the diameter and circumference of a circle
- It is assumed that a software developer is asked to write a Python program that can calculate and print out the diameter and the circumference of a circle. The user enters data of the radius and its measurement unit (in, ft, cm, or m) from the console.

Let's imagine this scenario:

The user inadvertently enters a negative value of the radius, which raises the following question:

- Should we let the program ignore this error?
- The answer is definitely "NO."
- So, what should we do?
- We should add selections into our program to check the sign of the input.

Let's write a better pseudo-code:

1. Start
2. Read the input of the radius from the console

   if(radius<0): ←selection

   inform the user about the error

   request to read again
3. Read the measurement unit of the radius (in, ft, cm, m)
4. Calculate the diameter of the circle

   diameter = 2 * radius
5. Calculate the circumference of the circle

   Circumference = diameter * PI (3.14159)
6. Print out the diameter
7. Print out the circumference
8. End

In [17]:
```
radius =int(input('Enter radius of Circle:'))
while radius <0:
    print('Radius value cannot be less than zero')
    radius =int(input('Please re-enter the correct value:'))
unit = input('Enter the unit of measurement in,ft,cm,m')
diameter = 2* radius
circumference = diameter *(3.14159)
print(diameter, unit,'is the diameter of the circle')
print(circumference, unit, 'is the circumference of the circle')
```

```
Enter radius of Circle:-30
Radius value cannot be less than zero
Please re-enter the correct value:-45
Radius value cannot be less than zero
Please re-enter the correct value:10
Enter the unit of measurement in,ft,cm,mm
20 m is the diameter of the circle
62.8318 m is the circumference of the circle
```

4. if Statements

1. Simple if:

if(boolean expression):

> //... ... ... statement(s)

Example:

numCredits = ... # number of credits an undergaduate student completed

if (numCredits >=90):

> studentLevel = "Senior"

2. if Statements: if...else:

if(boolean expression):

> //... ... ... statement(s)

else:

```
                    //... ... ... statement(s)
```

Example:

if(numCredits >= 120):

```
                readyToGraduate = True;
```

else:

```
                readyToGraduate = False;
```

3. if ... elif ... elif ... else

if(boolean expression):

```
            // ... ... ... statement(s)
```

elif (boolean expression):

```
          // ... ... ..... statement(s)
```

elif (boolean expression):

```
          // ... ... ..... statement(s)
```

else:

```
          // ... ... ..... statement(s)
```

Example:

It is assumed that the Registrar Office of a university asks one analyst to provide a solution to the following problem:

Write a Python program that can read input from the console. The user enters a student's name and his/her level (freshmen, ..., senior). The program is expected to assign a numeric code that represents his/her priority to register courses. Students who have higher priority are allowed to register courses before those with lower priority. The code starts from 1 (highest) that is assigned to seniors and increments by 1 for each lower level. Finally, the program prints out the student name, his/her level, and the code of priority to register courses in the same line.

Pseudo-Code with if ... elif ... elif ... else:

1. START
2. ... more code here ...
3. Perform the selection
   - If "senior", priorityToRegister = 1 // highest
   - If "junior", priorityToRegister = 2
   - If "sophmore", priorityToRegister = 3
   - If "freshman" , priorityToRegister = 4
   - If (not any above), print out warning of errors
4. ... more code here ...
5. END

Code:

studentLevel = ... #level: freshman, sophmore, junior, senior

if(studentLevel == "Senior")

```
    priorityToRegister = 1
```

if(studentLevel == "Junior")

```
    priorityToRegister = 2
```

if(studentLevel == "Sophmore")

```
    priorityToRegister = 1
```

Run the following code:

In [10]:

```
studentLevel= "Senior" # level: freshman, sophomore, junior, senior
if(studentLevel =="Senior"):
```

```
    priorityto Register = 1
elif(studentLevel =="Junior"):
    priorityto Register = 2
elif(studentLevel =="Sophomore"):
    priorityto Register = 3
elif(studentLevel =="Freshman"):
    priorityto Register = 4
else:
    print("Invalid studentLevel!!!")

print("studentLevel:", studentLevel, "; Priority to register",priorityto Register, "\
```

```
studentLevel: Senior ; Priority to register 1
```