**Python Lab 4**

**Decisions:** **If statement**

As always I believe I should start each lab with a warm-up typing exercise, so here is a short program to compute the absolute value of an integer:

n = int(input("Number? "))

**if** n < 0:

 **print**("The absolute value of", n, "is", -n)

**else**:

 **print**("The absolute value of", n, "is", n)

Here is the output from the two times that I ran this program:

Number? **-34**

The absolute value of -34 is 34

Number? **1**

The absolute value of 1 is 1

So what does the computer do when it sees this piece of code? First it prompts the user for a number with the statement "n = int(input("Number? "))". Next it reads the line "if n < 0:". If n is less than zero Python runs the line "print("The absolute value of", n, "is", -n)". Otherwise it runs the line "print("The absolute value of", n, "is", n)".

More formally Python looks at whether the *expression* n < 0 is true or false. An if statement is followed by an indented *block* of statements that are run when the expression is true. Optionally after the if statement is an else statement and another indented *block* of statements. This second block of statements is run if the expression is false.

There are a number of different tests that an expression can have. Here is a table of all of them:

|  |  |
| --- | --- |
| **operator** | **function** |
| < | less than |
| <= | less than or equal to |
| > | greater than |
| >= | greater than or equal to |
| == | equal |
| != | not equal |

Another feature of the if command is the elif statement. It stands for else if and means if the original if statement is false but the elif part is true, then do the elifpart. And if neither the if or elif expressions are true, then do what's in the else block. Here's an example:

a = 0

**while** a < 10:

 a = a + 1

 **if** a > 5:

 **print**(a, ">", 5)

 **elif** a <= 3:

 **print**(a, "<=", 3)

 **else**:

 **print**("Neither test was true")

and the output:

1 <= 3

2 <= 3

3 <= 3

Neither test was true

Neither test was true

6 > 5

7 > 5

8 > 5

9 > 5

10 > 5

Notice how the elif a <= 3 is only tested when the if statement fails to be true. There can be more than one elif expression, allowing multiple tests to be done in a singleif statement.

**Examples**

*# This Program Demonstrates the use of the == operator*

*# using numbers*

**print**(5 == 6)

*# Using variables*

x = 5

y = 8

**print**(x == y)

And the output

False

False

**high\_low.py**

*# Plays the guessing game higher or lower*

*# This should actually be something that is semi random like the*

*# last digits of the time or something else, but that will have to*

*# wait till a later lab. (Extra Credit, modify it to be random*

*# after the Modules lab)*

number = 7

guess = -1

**print**("Guess the number!")

**while** guess != number:

 guess = int(input("Is it... "))

 **if** guess == number:

 **print**("Hooray! You guessed it right!")

 **elif** guess < number:

 **print**("It's bigger...")

 **elif** guess > number:

 **print**("It's not so big.")

Sample run:

Guess the number!

Is it... **2**

It's bigger...

Is it... **5**

It's bigger...

Is it... **10**

It's not so big.

Is it... **7**

Hooray! You guessed it right!

**even.py**

*# Asks for a number.*

*# Prints if it is even or odd*

number = float(input("Tell me a number: "))

**if** number % 2 == 0:

 **print**(int(number), "is even.")

**elif** number % 2 == 1:

 **print**(int(number), "is odd.")

**else**:

 **print**(number, "is very strange.")

Sample runs:

Tell me a number: **3**

3 is odd.

Tell me a number: **2**

2 is even.

Tell me a number: **3.4895**

3.4895 is very strange.

**average1.py**

*# keeps asking for numbers until 0 is entered.*

*# Prints the average value.*

count = 0

sum = 0.0

number = 1 *# set to something that will not exit the while loop immediately.*

**print**("Enter 0 to exit the loop")

**while** number != 0:

 number = float(input("Enter a number: "))

 **if** number != 0:

 count = count + 1

 sum = sum + number

 **if** number == 0:

 **print**("The average was:", sum / count)

Sample runs

**average2.py**

*# keeps asking for numbers until count numbers have been entered.*

*# Prints the average value.*

*#Notice that we use an integer to keep track of how many numbers,*

*# but floating point numbers for the input of each number*

sum = 0.0

**print**("This program will take several numbers then average them")

count = int(input("How many numbers would you like to average: "))

current\_count = 0

**while** current\_count < count:

 current\_count = current\_count + 1

 **print**("Number", current\_count)

 number = float(input("Enter a number: "))

 sum = sum + number

**print**("The average was:", sum / count)

Sample runs:

This program will take several numbers then average them

How many numbers would you like to average: **2**

Number 1

Enter a number: **3**

Number 2

Enter a number: **5**

The average was: 4.0

This program will take several numbers then average them

How many numbers would you like to average: **3**

Number 1

Enter a number: **1**

Number 2

Enter a number: **4**

Number 3

Enter a number: **3**

The average was: 2.66666666667

**Exercises**

1. Write a program that asks the user their name, if they enter your name say "That is a nice name", if they enter "John Cleese" or "Michael Palin", tell them how you feel about them ;), otherwise tell them "You have a nice name."
2. Modify the higher or lower program from this section to keep track of how many times the user has entered the wrong number. If it is more than 3 times, print "That must have been complicated." at the end, otherwise print "Good job!"
3. Write a program that asks for two numbers. If the sum of the numbers is greater than 100, print "That is a big number."