



Hello, World!

Hello, World!

EMT1111: Logic and Problem Solving | Spring 2016 | Dr. Mendoza

LESSON 7 (Labs): Complex Boolean Expressions and Loops

Temperature with a function (temperature2.py)

Write a Python program to **ask the user for** the weather, specifically, **the current temperature in Fahrenheit** degrees. The program should define **a function** **getOpinion** that, **depending on the temperature** entered passed as parameter **temp**, the function should display one of the following messages:

- *“Not too bad out there”*, **if** the temperature is 50 or more
- *“You might want to wear a jacket today”*, **otherwise**

Temperature.py

```
weather = int(input("What is the current weather  
temperature? "))  
  
if weather >= 50:  
    print("Not too bad out there.")  
else:  
    print("You might want to wear a jacket today.")
```

Temperature.py

```
def getOpinion(temp):  
    if temp >= 50:  
        print("Not too bad out there.")  
    else:  
        print("You might want to wear a jacket today.")  
  
weather = int(input("What is the current weather  
temperature? "))  
  
getOpinion(weather)
```

Complex Boolean expressions

Complex Conditionals

You can use **and** and **or** to join several logical expressions together as shown in the table below.

Expression	Meaning
$(a < b) \text{ or } (c < d)$	The whole expression is true if a is less than b or c is less than d.
$(a < b) \text{ and } (c < d)$	The whole expression is true if a is less than b AND ALSO c is less than d.
$\text{not } a < b$	Only true if a is actually greater than or equal to b. The logical expression <code>not a < b</code> is the same as <code>a >= b</code> .

or means that if *either* of the expressions is *true*, the whole expression is *true*.

and means that *only if both* expressions are *true*, the whole expression is *true*.

not negates the logical value that follows it

Lab Assignment 1 (price.py)

Write a Python program that define the **function** **rightPrice** that receives a **parameter** **cost**. Depending on the **cost**, the function should display one of the following messages:

- *“Must buy!”*, if the price is less than 50
- *“On sale. Good price”*, if the price is between 50 and 59
- *“Regular price.”*, if the price is between 60 and 75
- *“Expensive!”*, if the price is greater than 75

Your program should **ask the user** for the price of a sweater and store it in the **variable** **price**. Then your program should call the **function** **rightPrice** and pass price as argument.

Lab Assignment 2 (tri_sides.py)

The sum of the lengths of any two sides of a triangle is greater than the length of the third side.

Write the function `validTriangle(side1, side2, side3)` that determine if the three sides measures passed as parameters can be the sides of a triangle. *The function should print out a message indicating whether the values entered by the user can form the sides of a triangle or not.*

Your Python program should ask the user to enter 3 numeric values to validate as a triangle, and call the `validTriangle` function.

Check the following example

Lab Assignment 2 (tri_sides.py)

```
Python 3.5.1 (default, Dec 2015, 13:05:11)
```

```
[GCC 4.8.2] on linux
```

```
>
```

```
Give me three numbers.
```

```
Value 1: 5
```

```
Value 2: 3
```

```
Value 3: 7
```

```
Good! The values you entered can form the sides of  
an triangle.
```

```
Python 3.5.1 (default, Dec 2015, 13:05:11)
```

```
[GCC 4.8.2] on linux
```

```
>
```

```
Give me three numbers.
```

```
Value 1: 3
```

```
Value 2: 5
```

```
Value 3: 10
```

```
Unfortunately, the values you entered cannot form  
the sides of a triangle.
```

The **while** Statement: Example 2

```
day = input("What days do you have EMT 1111? ")
while day != "Friday":
    print("Nope. That is not it.")
    print("Try again!")
    day = input("What days do you have EMT1111? ")
print("Yeap!")
print("Goodbye")
```

The **while** Statement: Example 2

```
answer = "no"
while answer != "yes":
    print("hi!!!")
    answer = input("Do you want to stop? ")
```

Lab Assignment 3: *Sign In 2* (signin2.py)

Write a Python program to **emulate** the signing in functionality (You should use a if-else statement).

- Create a variable named **psswd** to store a password (any password of your choice).
 - The password should be a string value (text)
- The program should ask the user to enter the password, assign it to the variable **mypswd (this is a different variable)**
- **If** the user enters the **correct password**, the program will simply say "**Access Granted**"
- **If** the user **does not enter the correct password**, the program will simply say "**Access Denied. Try Again**"
- The **program should keep asking** the user to enter the password as long as the user keeps giving an **incorrect password**

Lab Assignment 4 (joke.py)

```
print("Me: Knock! Knock!")
input("You: ")
print("Me: Ho-ho.")
input("You: ")
print("Me: You know, your Santa impression could use a
little work.")
```

Copy and run the above program. Then convert it in the function `theJock()`.

Then modify the program so that it will ask the user if he/she wants to hear the joke again. If the answer is "yes" should call the function `theJock()`. The program should keep on going as long as the user keeps answering yes and it should stop once the user answers no.