**The IF Statement**

**The Boolean Data Type**

There are several different **data types** that Python will recognize. The main ones are:

| Data Type | Example |
| --- | --- |
| integer | x = 3 |
| floating point number | x = 3.43453 |
| string | x = “This is string #49!” |
| Boolean | x = 5 > 3 |

We have seen the first three, but the last one is new: the Boolean data type.

A ***Boolean*** value can either be **True** or **False**. Boolean values are important to test the truth of a statement. For example, try this:

print (5 > 3) # is 5 greater than 3 ?

What happens? Now try this:

print (7 < 2) # is 7 less than 2 ?

A computer doesn’t know what True and False means. In the computer’s memory, a true statement is recorded as a “1”, and a false statement is recorded as a “0”. (Actually a “0” is False and any other value is considered “True” ). A ***Boolean variable*** stores a True or False value. This is useful to keep track of a state during a program. Here are some Boolean variables:

x = True

playerAlive = True

gameOver = False

ILikeLunch = True # most of the time

**The IF Statement**

Boolean values are most often seen in connection with ***conditional statements***. For example:

age = input("Enter your age:")

if age > 18: # this is a conditional statement

print ("Too old for school!")

Run this program. and enter the value 25 when prompted. When you do, it will cause an error. Read the error. The problem is that the computer assumes all input from the user is string input (words). If we want to input numbers only, we add one word:

age = int(input("Enter your age:"))

if age > 18:

print ("Too old for school!")

Now you can enter your age and it will reply with the sentence “Too old for school!”. If you try again with a low value such as 8, you will not get the sentence.

The syntax of an if statement is:

**if** (*condition)***:**

*statement1*

*statement2*

etc.

The ***condition*** in our example is that the age be above 18. If that condition is TRUE, the statements after it are executed.

Notice the statements in the if block are ***indented*** (shifted over). This is by design. In Python, ***the statements belonging to an if statement must be indented.*** This is extremely important. If you don’t indent, python will refuse to run. For example:

age = int(input("Enter your age:"))

if age > 18:

print ("Too old for school!") # this line should be indented!

Don’t worry if you aren’t sure when to indent. As you write Python code the IDE will automatically indent for you. The colon (:) is the sign that tells the editor to indent (try writing the if statement without the colon and then hitting enter – what happens?). You can manually indent using the “tab” key. Some programmers prefer to use 4 spaces instead of a tab, the choice is yours, however it is not a good idea to mix spaces and tabs. Choose one or the other for all your programs.

**ELIF and ELSE**

Let’s add to our program:

age = int(input("Enter your age:"))

if age > 18: # if you are over 18

print ("Too old for high school!")

else: # if you are NOT over 18

print ("Too young for high school!")

The last else statement means “...or else (if your age is less than or equal to 18) ...”

So now our program takes care of two possibilities. Is there any other possibility left? Lets add a final option:

age = int(input("Enter your age:"))

**if** age > 18: # if you are over 18...

print ("Too old for school!" )

**elif** age < 14: # or under 14

print ("You must be in elementary!")

**else:**  # anything else

print ("Finally someone my age! ")

Now there are three stages - if you are more than 18, if you are NOT more than 18 but you are less than 14, and finally everything else. The ***else*** statement means the age must be between 14 and 18.

**Indents**

To understand why indents are important, consider this code. Try entering a number more than 5, then try again with a number less than 5:

x = int(input("Enter a number from 1 to 10: "))

if x < 5:

print ("Your number is less than 5") #this is indented

print ( "Thank you") # so is this

The last line is indented, so it is part of the IF statement. But what if I want to say “Thank you” regardless? Then we unindent the last line:

x = int(input("Enter a number from 1 to 10: "))

if x < 5:

print ("Your number is less than 5") #this is indented

print ( "Thank you") # this is not

This way is better, because it says “Thank you” regardless of what number you type in. Politeness is greatness.

We use indenting in a similar way in English. Consider this to do list:

To do

Have breakfast.

**If** it is nice out :

* Go shopping for a bathing suit
* Go to the beach

Brush teeth.

Go to bed.

Can you see what happens if it isn’t nice out? The indents make it easy to see.

**Relational Operators**

These are the operators used with logical statements:

| **Operator** | **Meaning** | **Example** |
| --- | --- | --- |
| **<** | Less than | 4 < 7 |
| **>** | More than | 8 > 2 |
| **==** | Equal to | 3 == 3 |
| **>=** | More than or equal to | 5 >= 3, 4 >= 4 |
| **<=** | Less than or equal to | 2 <= 100 |
| **!=** | Not equal to | 2 != 3 |

**Assignment statement vs Relational statement**

In English, we use the equals sign (=) to mean two things:

1. Make this equal that: myage = 18
2. Check if this equals that: is myage = 18?

These are two very different meanings, so in programming we must have two separate symbols.

The first symbol is an **assignment operator (=)**:

x = 4 # make x equal to 4

The second is called a **relational operator (==)**:

x == 4: # is x equal to 4?

Notice in this version, we use two equals signs together.

Here is another example:

age = int(input("Enter your age:"))

if age **==** 17:

print ("Finally someone my age! ")

On the other hand, this example will not work:

age = int(input("Enter your age:"))

if age **=** 17: # error!

print ("Finally someone my age! ")

**Using the Modulus with IF**

Remember the modulus operator? No?

The modulus determines the *remainder* of a division. It can be very useful to determine if a number is divisible by another number. Recall that when a number is divisible, there is no remainder. For example:

a = 10

b = 2

print (a % b) # will print zero (there is no remainder)

c = 8

d = 3

print (c % d) # will not print zero (c is not divisible by d)

Here is how it looks inside an if statement:

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

if number % 3 == 0: # does it divide by 3?

print ("A multiple of 3! ")

**Exercises**

From now on, please add comments and a header to your code.

* Comments are brief explanations of the major ideas. Not every line needs to be commented, but just major sections. Your comments should be easy to understand, and should act like a translation of your code.
* The header is four lines containing: The name of your code, a brief description of what it does, your name, the date of submission.

A header looks like this:

########################################

# Program Title

# What the program does on this line

# Your Name Here

# v.1. Feb 22 2045

#########################################

A comment looks like this:

x = 32 # start with 32 dollars

1. Bank Machine

Beginning with the line:

balance = 830

Write a program that asks the user to enter an amount (of money) to withdraw from the bank. Inform the user if :

* The user enters a negative value (hint: less than zero).
* The user enters a value more than they have in the bank (their balance).
* The user enters a value that is not a multiple of $20. (**hint: use the modulus**)

If the user enters a valid amount (passes all the criteria above) then thank them and display their new balance in proper sentence form.

1. Add indents to make the following code functional:

team = str(input("Name your favourite hockey city:"))

if team == "Ottawa":

print ("Go Sens Go!")

elif team == "Toronto":

print ("I am afraid we can no longer be friends... Goodbye!")

else:

print ("At least you didn't pick Toronto...")

print ("Go Sens Go! ")

To get it right, make sure that the final statement ( “Go Sens Go!”) is only executed if the user enters something other than Ottawa or Toronto.

Notice that you have to spell the cities exactly as shown, including the case. This will not work. Can you see why?

Name your favourite hockey city: < toronto>

At least you didn’t pick Toronto...

Go Sens Go!

1. Write the following program from scratch. The program should ask the user to enter a number. The program should identify if the number is even or odd. The program should also identify if the number is positive, negative or zero.

Enter a number: <-7>

The number -7 is odd and negative

Hint #1: What are positive numbers? Think this operator : **>**

Hint #2: What are odd numbers? Think this operator : **%**

1. You are ordering pizza for a party. Each pizza has 8 pieces. Write a program that:

* asks the user how many people are attending.
* asks the user how many pieces each person will likely eat.
* determines how many pizzas need to be ordered.
* determines how many pieces will be left over. (hint - this actually doesn’t necessarily need the modulus).

\*\* Peter’s Pizza Program \*\* You are not Peter! Name it something else!

How many people are attending: <7>

How many pieces will each person eat: <2>

You will need to order 2 pizzas and there will be 2 pieces left over.

Keywords: ***Boolean variable, conditional statement, IF statement, ELIF statement, ELSE statement, indenting***