Python Operators and Conditionals

September 14, 2020

Today's topics

Python Operators

- Arithmetic
- Logical
- Assignment

Conditionals

- If, If-else, and if-elif-else

Homework 1 discussion

Operators

Special symbols that perform defined operations:

- Mathematical
- Comparison/Relational
- Assignment
- Boolean/Logical/Conditional

Mathematical Operators

+ - * / // % **

+ - addition; works as you would expect

- Subtraction; works as you would expect

* multiplication

/ floating point division - results in a float number

// Integer division. Only valid if you have two integers; produces an integer.

% modulo

** exponentiation

Some examples:

5/3

5//3

5%3

Comparison Operators

<   less than

<= less than or equal to

> greater than

>= greater than or equal to

==   equal to

!= not equal to

Be careful that you don't confuse = and ==

= is the assignment operator; it sets the value on the right to the variable on the left

== is a test to see if what's on the left is equal to what's on the right

Assignment operators

=

+=

* =

-=

/=

Some examples

num_candy = num_candy + 1

num_candy += 1

num_candy *= 2

num_candy -= 2

num_candy /= 2

Boolean operators

and

or

not

A boolean is either True or False

Boolean operators take Boolean values, combine them and yield a single Boolean value

9 > 8 and 5 < 9

num_candy != 0 or choice = ‘yes’

Boolean Operators: And, Or, Not

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

x	y	x or y
True	True	True
True	False	True
False	True	True
False	False	False

X	not X
True	False
False	True

Thanks, Dr. Johnson

Order of operations

**	Highest
* / % //	
- +	
<= < > >= != ==	
not	
and	
or	Lowest

Remember, like the noble and majestic honey badger, parentheses don’t care about order of operations and will take precedence over everything.

Some coding examples with operators

Conditionals

Conditionals

Sometimes you want to write code that only gets executed sometimes

- Only print an error message if the user has entered a negative number for "age"
- Only print a congratulatory message if the student has made the Dean's List
- …

So you use conditionals to control whether or not a section of code (one or more statements) gets executed

Conditionals rely on Boolean expressions to determine whether or not code gets executed

Three forms of conditionals:

1. One option: If a condition is true, do something. Otherwise, do nothing. “If” statement
2. Two options: If a condition is true, do something, Otherwise, do something else. “If….else…” statement
3. More than two options: If a condition is true, do something. Otherwise, check to see if another condition is true; do something else. Otherwise, keep checking conditions until we find one that’s true or we just give up. “If...elif...elif...else…” statement

The simplest conditional: the if statement

```
if [boolean]:
   # do something
```

Example:

```
if student == "awesome":
   print("You must be here for CMSC 201")
```

Now you know why we spent so much time talking about True, False, comparison and logical operators! They go after ‘if'!

Two cases: If and Else

Ask a student for her major. If she’s a Physics major, print out “smart choice.” Otherwise print out “there is still time”

```
major = input(“please enter your current major”)

if major == “Physics”:

    print(“smart choice”)

else:

    print(“there is still time”)
```

Remember that input returns a string, so this comparison is valid

Colon after else, as well!

Indent the code that’s part of the “else” block

Python and indentation

The boolean condition is everything between “if” and the colon : A colon terminates the condition. It can be as simple or as complex as you want

Indentation matters!! To python, white space - either tabs or spaces - indicates what’s in the code to be executed. You only have to indent one space, but I’m a believer that you should indent with tabs.

```
If x == 6:
    print (“X is 6”)

    print (“and we’re done”)
```

If x is really equal to 5, this doesn’t print out anything.
This still prints “and we’re done” because that’s not part of the conditional

```
if x == 6:
  print (“X is 6”)
print (“and we’re done”)
```

“if...else…”

- There must always be at least one line of code under the “if” statement
- You don’t have to have an “else” part, but if you do have an “else” there must be at least one line of code under it.

Three or more cases: elif

Input returns a string. This turns it into an integer

Ask a student her age. If it’s less than 18, tell her she’s a minor and faces some restrictions. If it’s between 18 and 21, tell her she’s an adult but still has some limitations. If she’s over 21, tell her she’s legally an adult.

This is ‘pseudocode’

```
age = int(input(“Please enter your current age in years.”))
if age < 18:
        print(“Sorry but you are a minor”)
        print(“there are a lot of things you cannot do on your own”)
elif age < 21:
        print(“you are an adult but there are still some things you can’t do”)
else
        print(“congratulations you are legally an adult”)
```

Some coding examples with conditionals

Homework discussion

Homework #2 is available; it’s due next Monday at midnight.

We’ll talk about it in class on Wednesday

Now let’s talk about Homework #1

Remember that you can get help during office hours on Discord - please take advantage of that