Short review of if-elif-else statements

What is the difference between

```
if condition:
    statement
    statement
if condition:
    statement
    statement
```

```
if condition:
    statement
    statement
else:
    statement
    statement
```

Short review of if-elif-else statements

Why does order matter here?:

```
aqi = int(input(‘Enter the air quality index (aqi): ’))
if aqi >300:
        print (‘Air quality is hazardous for everyone’)
elif score >200:
        print (‘Air quality is very unhealthy for everyone’)
elif score >150:
        print (‘Air quality is unhealthy for everyone’)
elif score >100:
        print (Air quality is unhealthy for sensitive people.’)
else:
        print(‘Air quality is moderate or good.’)
```

Setting booleans as flags

A *flag* is a variable that signals when some condition exists in the program. When the flag variable is set to True, it means the condition does exist.

```
if weather== ‘rainy’:
   umbrella = True
else:
   umbrella = False

if umbrella:
   print(“Don’t forget your umbrella!”)
```

You can also set your flag at the start such as:

```
umbrella = True
```

Logical Operators: and, or, not

Logical Operators: and, or, not

- a *and* b: True only if *both* a *and* b are True
- a *or* b: True if *either* a *or* b are True
- *not* a: True only if a is False

Examples of *and* in real life

If hungry ***and*** cafeteria is open, get food.

```
if hungry:
    if cafeteriaOpen:
        getFood

if hungry and cafeteriaOpen:
    getFood
```

Truth Tables for *and, or*

S1 *and* S2	**S2 == True**	**S2 == False**
S1 == True	True	False
S1 == False	False	False

S1 *or* S2	**S2 == True**	**S2 == False**
S1 == True	True	True
S1 == False	True	False

Examples of Using Logical Operators: and, or, not

- Test a number to see if it is in a range:
  - E.g., Is number between 1 and 10?

```
if number >= 1 and number <= 10:
  print(“Number is in range!”)
else:
  print(“Number is out of range!”)
```

Note on logical operators - try these!

```
n=2

if n==1 or n==2:
    print('True')
```

```
n=2

if n==1 or 2:
    print('True')
```

```
n=2

if n==1 or 3:
    print('True')
```

How is that happening??

This is due to boolean value of integers in Python. 0 would be false.

Examples of Using Logical Operators: and, or, not

Applicant qualifies for loan from bank if:

- salary > $20000.00
- ***and*** years on job >= 3

Two ways to code it:

```
if salary > 20000:
    if years_on_job >= 3:
        answer = “Yes”
    else:
        answer = “No”
else:
    answer = “No”
print(answer)
```

```
if salary > 20000 and years_on_job >= 3:
    answer = “Yes”
else:
    answer = “No”
print(answer)
```

Examples of Using Logical Operators: and, or, not

- Student can register for a class if:
  - Took AP calculus in high school (hsCalc = True)
  - ***or*** is taking calculus now (calcNow = True)
- Two ways to code it:

```
if hsCalc:
    answer = “Yes”
else:
    if calcNow:
        answer = “Yes”
    else:
        answer = “No”
print(answer)
```

```
if hsCalc or calcNow:
    answer = “Yes”
else:
    answer = “No”
print(answer)
```

Short-Circuit Evaluation

- Logical operators evaluate only as much of a combined expression as necessary
  - For efficient operation of program

- **and**: keeps going until one part is **False**
  - a == 5 **and** b < 10 **and** age < 47 **and** height >= 2.5

- **or**: keeps going until one part is **True**
  - a == 5 **or** b < 10 **or** age < 47 **or** height >= 2.5

Be careful with combining logical operators

```
if b > 10 or c < 20 and d == 5:
    print('first one works')

if ((b > 10) or (c < 20)) and (d == 5):
    print('second one works')

if b > 10 or (c < 20 and d == 5):
    print('third one works')
```

Incrementing a variable (needed for lab 1)

```
>>>year = 2018
>>>year = year+1
>>>year
2019
```

You can also increment with:

```
>>>year+=1
>>>year
2020
```

Same for -= *= /=