INFO 450 Fall 2020

Week 3

Agenda

- boolean (truthiness)
- conditionals
- if, elif, else

cars.py

Example code and output to talk about equality, conditions, etc.

All car manufacturers should be printed 'title' case, but BMW should be all caps.

```
cars = ['audi', 'bmw', 'subaru', 'toyota']

for car in cars:
  if car == 'bmw':
      print(car.upper())
  else:
      print(car.title()
```

```
Audi
BMW
Subaru
Toyota
```

Note: Using print, as those outputs aren't "logging" type statements.

Boolean Expressions

Whether a statement is True or False

Used to track conditions/conditionals

```
>>> game_active = True
>>> can_edit = False
>>> can_edit = true
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'true' is not defined
```

Conditionals

"At the heart of every *if* statement is an expression that can be evaluated as True or False and is called a 'conditional test'"

Checking for equality:

```
>>> car = "bmw" # Assigns 'bmw' to variable car
>>> car == "bmw" # Compares variable car to value "bmw"
True
```

```
>>> car = "audi"
>>> car == "bmw"
False
```

Testing strings is 'case sensitive'

```
>>> car = "Audi"
>>> car == "audi"
False
>>> car.lower() == "audi".lower()
True
```

Checking for Inequality

```
>>> requested_topping = "mushrooms"
>>> requested_topping != "anchovies"
True
```

! creates the 'inverse' of the resulting comparison

True becomes False

False becomes True

Numerical Comparisons

```
>>> age = 18
>>> age == 18
True
>>> age != 18
False
>>> age < 20
True
>>> age <= 17
False
>>> age > 15
True
>>> age >=18
True
```

```
>>> age = 17
>>> if age < 21:
>>>     print("No drink for you.")

No drink for you.
```

Multiple Conditions

'and' 'or'

```
>>> age_0 = 22
>>> age_1 = 18
>>> age_0 >= 21 and age_1 >=21
False
```

AND

	True	False
True	True	False
False	False	False

OR

	True	False
True	True	True
False	True	False

Checking lists

Is it IN the list?

```
>>> requested_toppings = ['mushrooms', 'onions', 'pineapple']
>>> 'mushrooms' in requested_toppings
True
>>> 'bacon' in requested_toppings
False
```

Is it NOT IN the list?

```
>>> banned_users = ['andrew', 'carolina', 'david']
>>> user  = 'marie'
>>> user not in banned_users
True
```

if Statements

'one test and one action, simplest form'

```
if -conditional test-:
    do something
```

```
age = 19
if age >= 18:
    print("You are old enough to vote!")
```

Code sets the age variable, checks it against 18, and then decides to execute the commant.

```
age = 19
if age >= 18:
    print("You are old enough to vote.")
    print("Have you registered to vote?")
```

if-else Statements

if statements help execute an action when conditions are met, but usually programs need to do something else if they aren't met

```
age = 17
if age >= 18:
    print("You are old enough to vote.")
    print("Have you registered to vote?")
else:
    print("Sorry, you're too young to vote.")
    print("Please register to vote as soon as you turn 18.")
```

if-elif-else chain

If we need more than one or two 'branches' in the code, we can add more conditional checks

```
age = 12
if age < 4:
    print("Your admission cost is $0.")
elif age < 18:
    print("Your admission cost is $25.")
else:
    print("Your admission cost is $40.")
```

Supports 0 or more 'elif' statements

Once one scenario is reached as True, the rest are ignored.

Else block can be omitted

Testing Multiple Conditions

Sometimes you want more than one action to happen based on events.

```
requested_toppings = ['mushrooms', 'extra cheese']

if 'mushrooms' in requested_toppings:
    print("Adding mushrooms.")
if 'pepperoni' in requested_toppings:
    print("Oh yes, all the pepperoni.")
if 'extra cheese' in requested_toppings:
    print("Adding extra cheese.")

print("\nFinished making your pizza.")
```

Checking if a list is empty or not

```
requested_toppings = []
if requested_toppings:
    for requested_topping in requested_toppings:
        print(f"Adding {requested_toppings}.")
else:
    print("Are you sure you want a plain pizza?")
```

Using multiple lists

Check our pizza order against an available list of toppings.

```
available_toppings = ['mushrooms', 'extra cheese', 'olives',
                      'pineapple', 'pepperoni', 'green peppers']
requested_toppings = ['mushrooms', 'french fries', 'extra cheese']

for requested_topping in requested_toppings:
    if requested_topping in available_toppings:
        print(f"Adding {requested_topping}")
    else:
        print("Sorry, we don't have {requested_topping}")
```

Styles

```
if age < 4: #looks good with whitespace
    pass # do nothing

if age<4: # looks ugly, add whitespace
    pass
```

Code practice from book:

Page 84 - check_users.py

Homework - Due Wednesday night, midnight easter

Page 89 - check_users.py

In your github repo, [username]/[repo]/check_users.py

Complete 5-10 on page 89 of your book by completing the following program:

```
def check_users(current_users, new_users):
    pass
    # YOUR CODE HERE

if __name__ == "__main__":
    current_us = ['chris','haritha', 'sally', 'darnell', 'superman']
    new_us = ['george', 'ringo', 'superman', 'hannibal']
    check_users(current_us, new_us)
```

Assignment will be listed in Canvas. When complete, link to your .py file in github.