Conditions

Fall 2019

Conditions

- Allow for a block of code to be executed only under certain conditions
  - Does so through comparisons
- Signified by if, elif, and else statements

Comparison	Definition
==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Condition Anatomy

- What you're checking (or comparing) should be inside of parentheses followed by a colon (:)
- Any code associated with that condition (so, if we wanted to do certain calculations if a value was equal to three) *NEEDS* to be indented
    - Any code you *don't* want associated with that condition should be unindented

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

# checks a condition
if(number == 3):
    print("Your number is equal to three")

    # you can have more code in here
    # it doesn't just have to be one line

# anything outside of your condition
print("This happens regardless.")
```

Condition Anatomy

- Conditions can also contain an “else”
- This is performed only if the “if” part of the comparison isn’t used
- An “else” has *no conditions* to be checked
  - It is still followed by a colon
  - It still has indentation rules
- There can only be ***one*** else per “if” -- you also *cannot* have an else without an “if”
- There cannot be any code in between the “if” and “else”

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

# checks a condition
if(number == 3):
    print("Your number is equal to three")

    # you can have more code in here
    # it doesn't just have to be one line

# else...
else:
    print("Your number is not three")

# anything outside of your condition
print("This happens regardless.")
```

Condition Anatomy

- Elif's are used to check another condition (combination of "else if")
- Each elif has a condition in the same syntax as an if-statement
- An elif *will not* be checked if an "if" (or "elif") before it is triggered
- There can be multiple "elif"s per if-statement
- You *cannot* have an elif-statement without a preceding if-statement

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

# checks a condition
if(number == 3):
    print("Your number is equal to three")

# if it's three, this is never checked
# because we KNOW it isn't four
elif(number == 4):
    print("Your number is four")

# else...
else:
    print("Your number is not three or four")

# anything outside of your condition
print("This happens regardless.")
```

Compound Conditions

- You can check multiple conditions in one if-statement (or elif-statement)
  - This is done by “and” and “or” conditions
- There can be multiple ands and ors per condition
- This can help identify ranges (with numbers) or if an answer can be multiple things (such as “yeah”, “yes”, “sure”, etc)
- OR - needs only ONE condition to be true to equate to true
- AND - needs EVERY condition to be true to equate to true

Compound Condition Anatomy

- Can be used on if-statements and elif-statements
- Must check the variable EVERY time
  - Ex. if checking whether a number is equal to 3 or 4
  - if(number == 3 or 4) -- *invalid*
  - if(number == 3 or number == 4) -- *valid*

```
answer = input("Enter yes or no: ")

# checks for correct input
if(answer != "yes" and answer != "no"):
    print("That's not a valid answer!")
```

Multiple Ifs VS If/Elif Statements

- There is a time to use multiple if-statements and a time to use a series of if/elif statements
- Technically, if/elif-statements, when usable, are more efficient
  - Since elif-statements are *only* checked if a previous hasn't been met, it prevents excess computing
- Multiple if-statements
  - Used when comparing variables that have nothing to do with one another
- If/Elif-statements
  - Used when comparing the same variable or if triggering one statement should *end* the comparisons

Multiple Ifs VS If/Elif Statements

- In this example, multiple if-statements make sense because the lucky number and the favorite animal have nothing to do with each other
- If we used an elif-statement for the lucky number, it wouldn't be checked if we had the same favorite animal

```
animal = input("What is your favorite animal? ")
number = int(input("What is your lucky number? "))

# checks favorite animal answer
if(animal == "dog"):
    print("Cool, me too.")

# checks lucky number answer
if(number != 8):
    print("Oh. Mine is 8")
```

Condition Notes:

- Make sure you pay attention to data types
  - In this case, you won't always get a type-error because it's comparing
    - == and != won't yield a type error
    - <, >, <=, and >= will yield a type error sometimes
  - It just might not hit a condition when you think it should if the types are incorrect
- Pay attention to when it's best to use multiple if-statements versus an if/elif-statement block
- You can have multiple if/elif-statement blocks in your code -- just make sure you know that if you have multiple if-statements, any following elif or else-statements only coincide with the most recent if-statement