Junior Knights Python I - Week 5

Review & For Loops

Review

Variable Typing

Input/Output

Mathematical Operations

If, Else, and Elif

Variable Typing

- Integer
- Double/Float
- String
- Character

- Valid naming conventions
  - Cannot start with a number
  - Case-sensitive

- In other languages, you have to declare your variable's type before using it. Python does this for us implicitly

Input/Output

- How do we format a statement that asks for a user inputted integer?
  - A string?
  - A float?

- How do we format a statement we want to appear as output (with no variables)?
  - If we wanted to print a statement with a variable value?

Mathematical Operations

- Basic Operations:
  - Addition (+), subtraction (-), multiplication(*), division(/)

- Python Operations
  - Exponentiation (**)
  - Integer division (//)
  - Modulus (%)

If, If-Else, and Elif

- Relational Operators:
  - ==
  - !=
  - >
  - >=
  - <
  - <=

- How do we format an if statement?
- Do you need to have an else statement?
- How many elif statements can you have?

For Loops

Usage

What is the point of for loops?

- When we want to run multiple similar operations and we know in advance how many times we want them to run

Say we wanted to print a countdown, from 10 to 1.

Usage

What is the point of for loops?

- When we want to run multiple similar operations and we know in advance how many times we want them to run

Say we wanted to print a countdown, from 10 to 1.

One way of doing this is using multiple print statements.
However, as you progress in your coding pursuits, it's important that you keep your code as efficient as you can manage.

We could recreate that same output in two lines with for loops.

Syntax

```
for <variable> in <range>:
    stmt1
    stmt2
    ...
    stmtn
```

Note that like if statements, the for loops will only execute what is under its indentation:

```
 for <variable> in <range>:
    stmt1
    stmt2
Stmtn
```

Would result in statement n being executed **outside** of the for loop.

Ways to Define Range

- The most simple way to specify a range is to provide a single number. For example, range(10), represents the set of integers {0, 1, 2, ...,9}.
  - Therefore, `range(n)` will iterate from 0 to n-1. Note that this is not inclusive of n itself.

- To set a different starting number than 0, we use the syntax `range(a,b)`, where a is the number you begin at, and you iterate until b-1.

- Lastly, if you wanted to iterate by numbers other than 1 (say, print every other number or every 5th number, etc.) we use the syntax `range(a,b,c)`, where a and b hold the same functions as above, but c represent the “step size”. This step size will be added until the value exceeds OR EQUALS b.

Live Coding