Intro to Coding with Python– For Loops

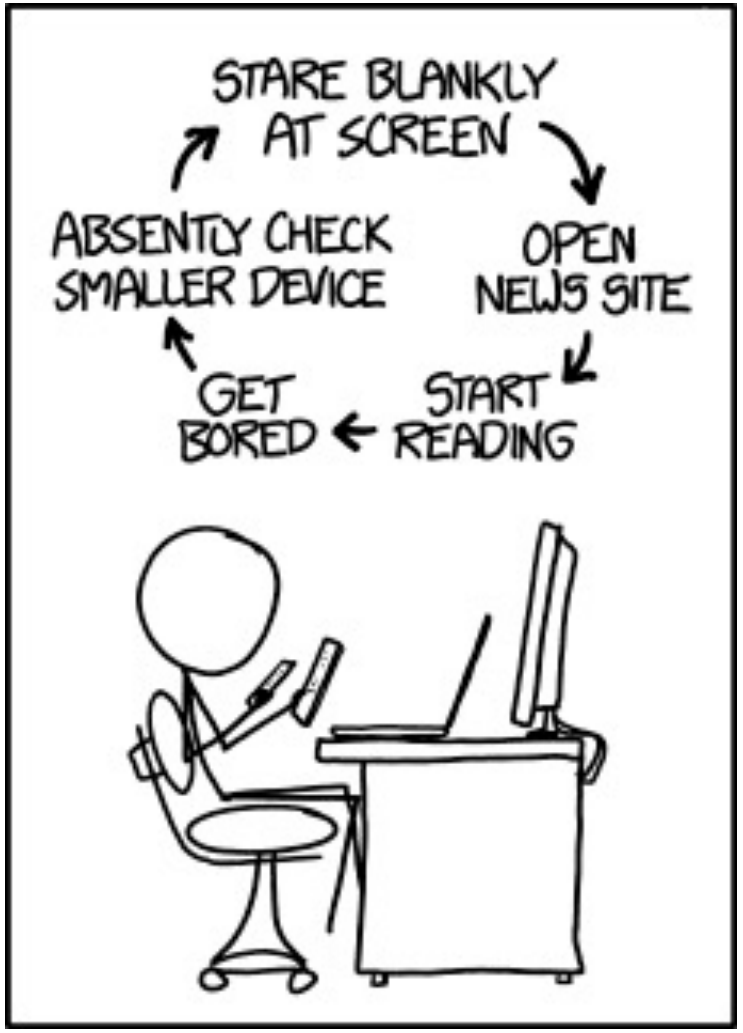
Dr. Ab Mosca (they/them)

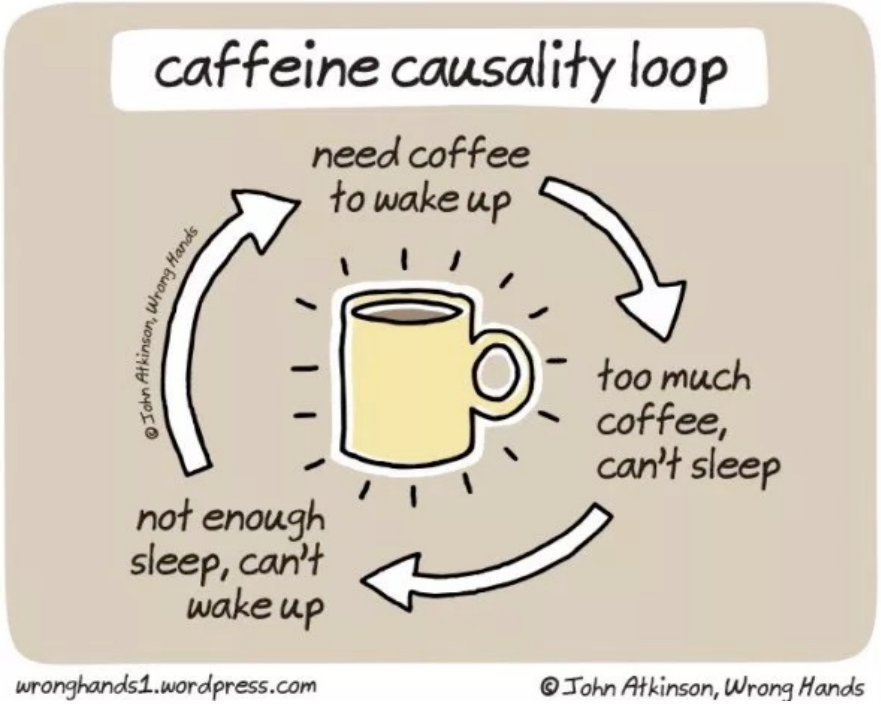
Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

- For loops

Loops: a familiar idea





https://xkcd.com/1411/

Loops in computer programming

- Sometimes, we want to do the exact same thing multiple times
- Ideally, we can write the code we need repeated once, and tell the computer to repeat it as many times as needed

Loops in computer programming

- Sometimes, we want to do the exact same thing multiple times
- Ideally, we can write the code we need repeated once, and tell the computer to repeat it as many times as needed
- Enter: ***Loops***

Loops in computer programming

- Sometimes, we want to do the exact same thing multiple times
- Ideally, we can write the code we need repeated once, and tell the computer to repeat it as many times as needed
- Enter: ***Loops***
- A ***loop*** is a chunk of code that we tell the computer to continuously repeated for a specified time

Loops in computer programming

- Sometimes, we want to do the exact same thing multiple times
- Ideally, we can write the code we need repeated once, and tell the computer to repeat it as many times as needed
- Enter: ***Loops***
- A ***loop*** is a chunk of code that we tell the computer to continuously repeated for a specified time
- There are three main approaches:
  - run **until** some condition is met
  - run for **each item** in a list
  - run a specific **number of times**

Loops in computer programming

- Sometimes, we want to do the exact same thing multiple times
- Ideally, we can write the code we need repeated once, and tell the computer to repeat it as many times as needed
- Enter: ***Loops***
- A ***loop*** is a chunk of code that we tell the computer to continuously repeated for a specified time
- There are three main approaches:
  - run **until** some condition is met
  - run for **each item** in a list
  - run a specific **number of times**

`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list

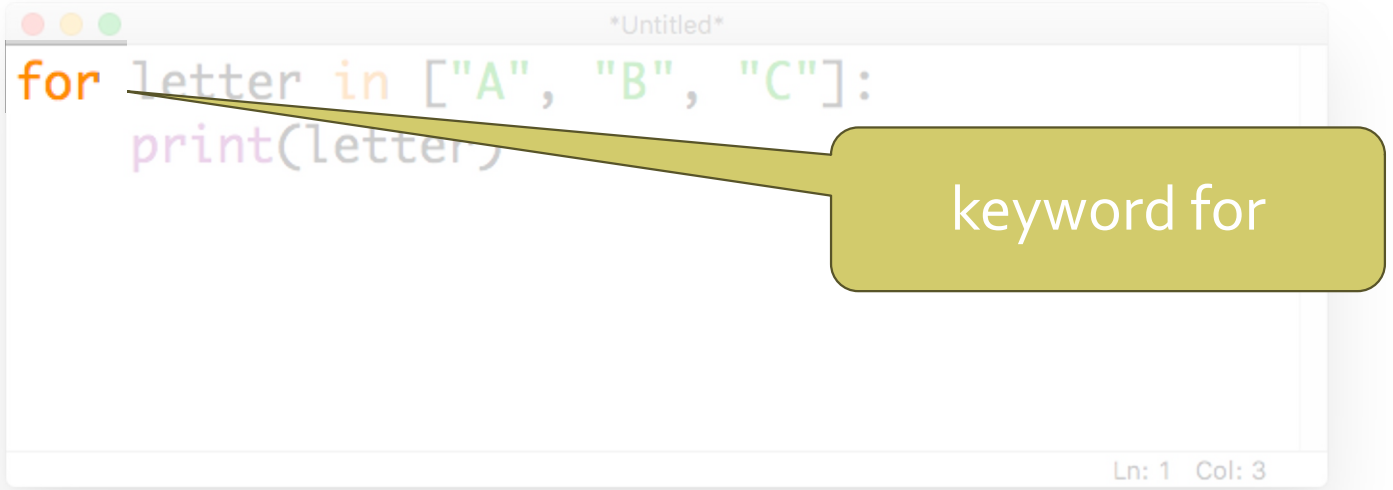
Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Ln: 1 Col: 3

`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list



`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list

```
*Untitled*
for letter in ["A", "B", "C"]:
    print(letter)
Ln: 1   Col: 3
```

variable that will be used in the loop code

`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list

Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

keyword in

Ln: 1 Col: 3

`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list

Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Ln: 1 Col: 3

list of values to use in loop code

for...in loops

- In Python, we use the keywords **for** and **in** to loop through a list

Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

colon

Ln: 1 Col: 3

`for...in` loops

- In Python, we use the keywords **`for`** and **`in`** to loop through a list

Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Ln: 1 Col: 3

indented code to be repeated

`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:

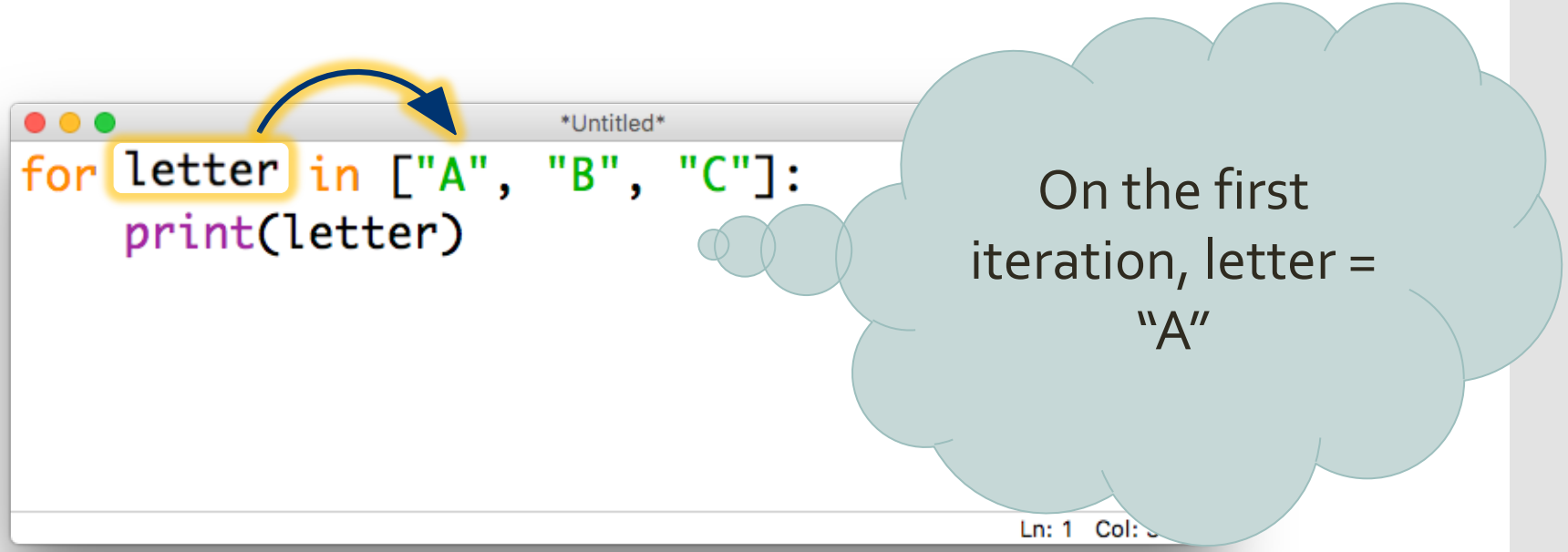
Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Ln: 1 Col: 3

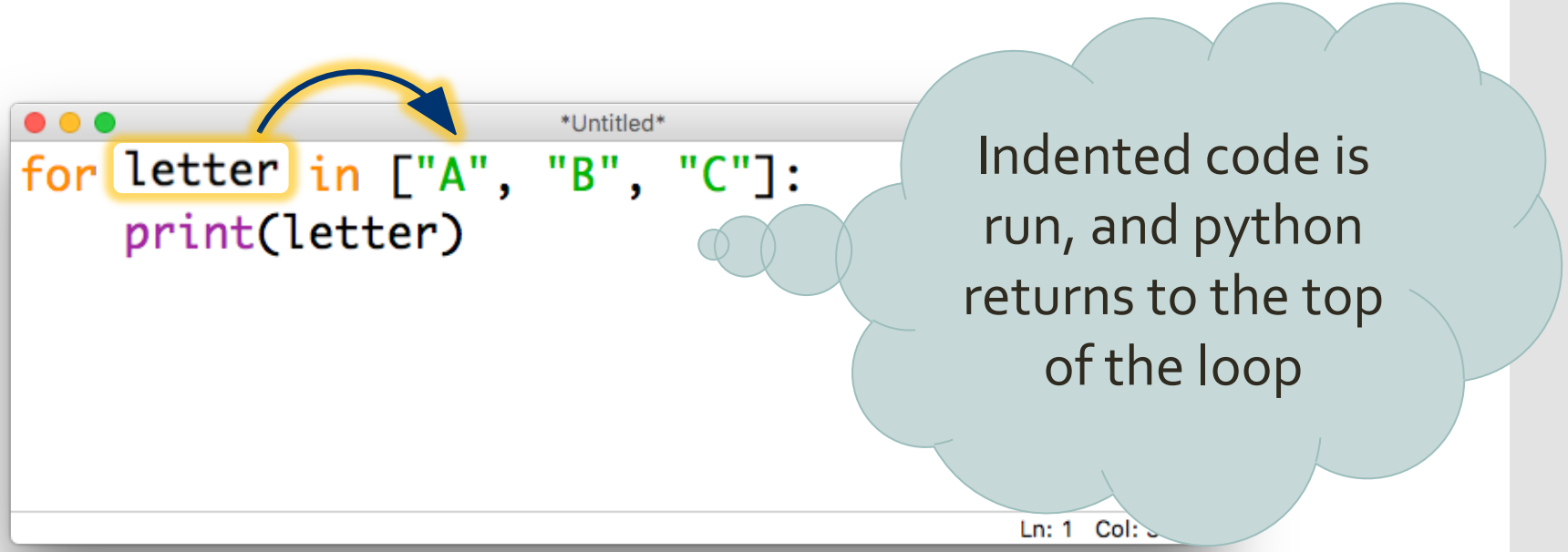
`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:



`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:



`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:

```
for letter in ["A", "B", "C"]:
    print(letter)
```

On the second iteration, letter = "B"

`for...in` loops

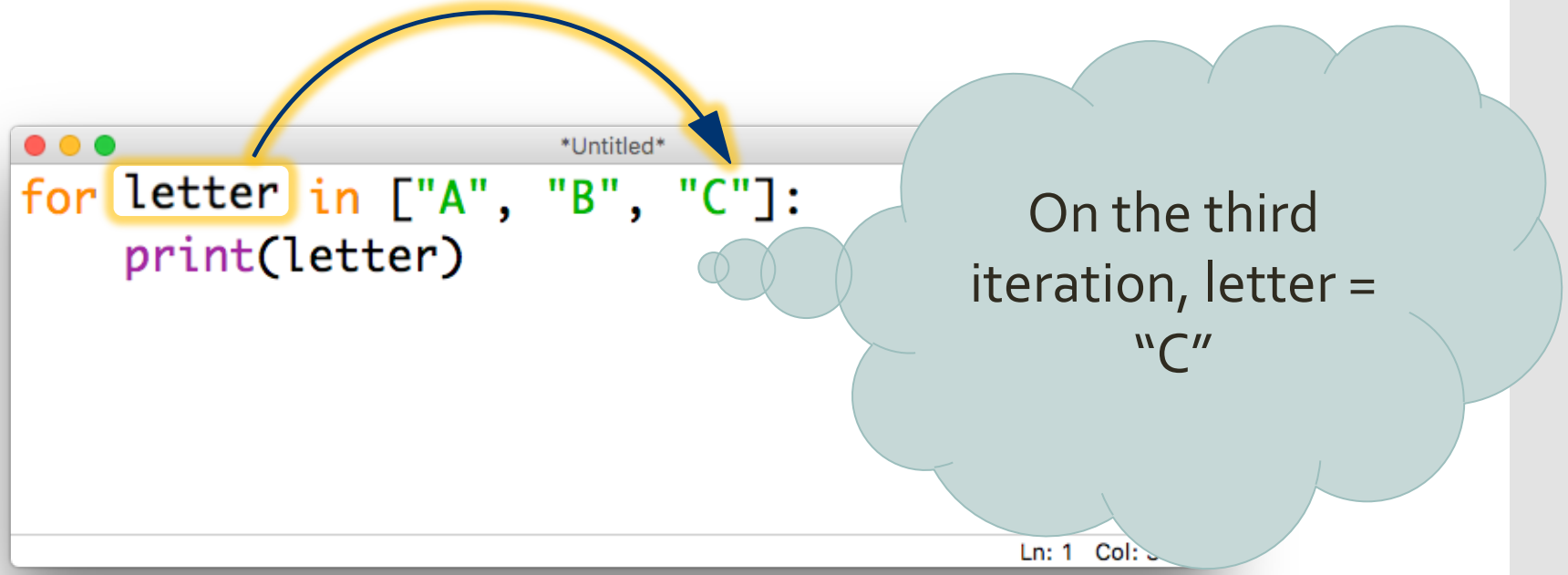
- We can think of this in terms of where the variable **`letter`** is pointing:

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Indented code is run, and python returns to the top of the loop

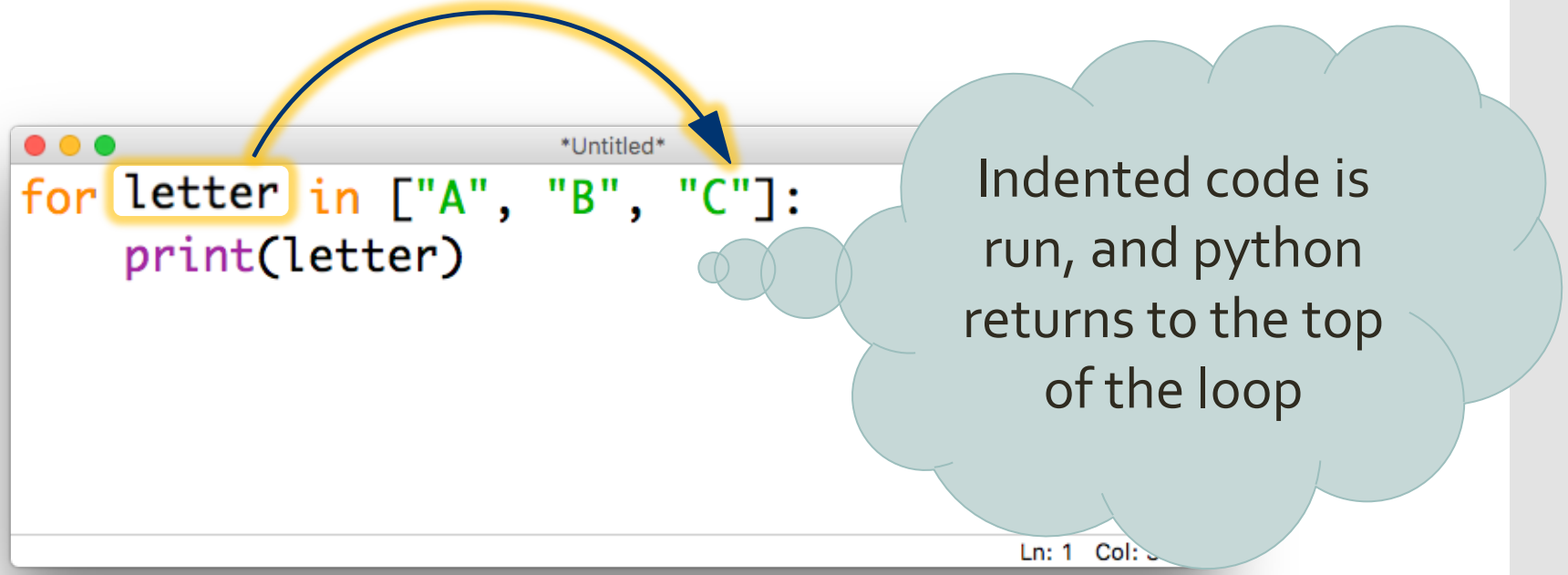
`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:



`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:



`for...in` loops

- We can think of this in terms of where the variable **`letter`** is pointing:

Untitled

```
for letter in ["A", "B", "C"]:
    print(letter)
```

Ln: 1 Col:

All letter has been set to all items in the list, so python moves on to un-indented code

`for...in` loops: unpacked

- We could accomplish the same thing by writing it out as **three separate assignments**:

Untitled

```
letter = "A"
print(letter)
letter = "B"
print(letter)
letter = "C"
print(letter)
```

Ln: 6 Col: 13

`for...in` loops: a common "gotcha"

- Python will allow you to **modify a list** while you're looping through it:

*lecture4-demo.py - /Users/jcrouser/Google Drive/Teac...

```
for letter in ["A", "B", "C"]:
    letter = letter.lower()
    print(letter)
```

Ln: 5 Col: 0

- This is generally a **bad idea** (more on why later)
  - it's fine to format the **values**, etc.
  - just don't **overwrite** the originals!

Demo: compute a sum

Use a `for` **loop** to compute
the **sum** of a list of numbers input by the user

Looping `n` times

- **Bad news**: there isn't a way to say "run this loop `n` times" in Python – we'll have to find a way around that
- If we want a **`for...in`** loop to run a specific # of times, we can "trick" it using a list of numbers that's the right size

Untitled

```
for i in [1,2,3,4,5]:
    print("Hello!")
```

Ln: 2 Col: 19

Looping `n` times

- **Bad news**: there isn't a way to say "run this loop `n` times" in Python – we'll have to find a way around that
- If we want a **`for...in`** loop to run a specific # of times, can "trick" it using a list of numbers that's the right size

common practice: use the letter **`i`** to denote the "index"

Untitled

```
for i in [1,2,3,4,5]:
    print("Hello!")
```

Ln: 2 Col: 19

Looping `n` times

- **Bad news**: there isn't a way to say "run this loop `n` times" in Python – we'll have to find a way around that
- If we want a **`for...in`** loop to run a specific # of times, can "trick" it using a list of numbers that's the right size

common practice: use the letter **`i`** to denote the "index"

Untitled

```
for i in [1,2,3,4,5]:
    print("Hello!")
```

Col: 19

What will this code do?

The `range()` function

- The **`range()`** function lets us generate lists of integers
- Given **one** integer **`a`**, **`range(a)`** will generate a list starting at 0 and going up to (but not including) **`a`**
- For example, if we want a loop to run **5** times:

Untitled

```
for i in range(5):
    # do something
```

Ln: 1 Col: 15

The `range()` function

- Given **two** integers **`a,b`**, **`range(a,b)`** will generate a list starting at **`a`** and going up to (but not including) **`b`**
- E.g., if we want to loop over the integers from 1 to 5:

Untitled

```
for i in range(1,6):
    # do something
```

Ln: 1 Col: 18

The `range()` function

- These values can be **positive** or **negative** (but for now, the second integer should be **larger** than the first)
- E.g., if we want to loop over the integers from -5 to 5:

Untitled

```
for i in range(-5,5):
    # do something
```

Ln: 1 Col: 19

The `range()` function

- Given **three** integers **`a,b,c`**, calling **`range(a,b,c)`** will generate a list starting at **`a`** and going up to (but not including) **`b`** with step size **`c`**
- E.g., if we want the integers from 0 to 9, counting by 3s:

Untitled

```
for i in range(0,10,3):
    # do something
```

Ln: 1 Col: 21

The `range()` function

- If we want to count down instead of up, we can set **`b < a`** and use a negative step size
- E.g., if we want to count down from 10 to 1:

Untitled

```
for i in range(10,0,-1):
    # do something
```

Ln: 1 Col: 20

15-Minute Exercise: convert $°F$ to $°C$

Use a **`for`** **loop** and the **`range()`** function to generate a **conversion table** of temperatures from $°F$ to $°C$ ranging from $100°F$ to $-30°F$ in increments of $10°F$

Tips:

- use the formula $°C = (°F - 32) * 5 / 9$

Discussion

What did you come up with?