

# Lab 5 GRS

While loops, strings, constants

# Housekeeping

- Exam 1 grades are out
  - Please check your exam to ensure none of the grading is wrong
  - Submit regrade requests on gradescope if the grading is wrong

# Magic Numbers

- From this point on, do **not** use magic numbers
- Magic numbers are numbers that are **not** 0, 1, or (generally) 2

```
circumference = 3.14 * diameter
```




Magic  
number

# Constants

- Variables that **don't** change in your program
- Make your code easier to read
- Constants are declared at the top of your program before main

```
PI = 3.14  
NUM_LETTERS_IN_ALPHABET = 26  
NUM_ITEMS = 14  
MAX_STUDENTS = 100000000
```

```
circumference = PI * diameter
```



constant

# Strings

- Strings are **immutable!!!**
  - Meaning if you use a string operation, the original string doesn't get changed
  - A copy of the string will get returned
  - Should assign modified string to a variable if you want to use it

```
my_string = "!!!!!!hello, world!!!!!!"  
print(my_string)  
my_string.strip("!")  
print(my_string)
```

```
!!!!!!hello, world!!!!!!  
!!!!!!hello, world!!!!!!
```

# Strings

- You can treat strings like lists
  - They're kinda like lists of characters
- You can loop through them like a list and do (most) list functions on them

# String functions

- `split()` - separates string by spaces, returns a list
- `split(delimiter)` - separates string by a string delimiter, returns a list
- `strip()` - takes away leading and trailing whitespace from a string, returns a string
- `strip(delimiter)` - takes away leading and trailing delimiter, returns a string

# Concatenation

- “Joining” 2+ strings together with plus signs
- Remember that strings are immutable, you must assign the concatenated string to a variable



```
string_1 = "Hello"  
string_2 = "World"  
  
string_3 = string_1 + ", " + string_2 + "!"  
  
print(string_1)  
print(string_2)  
print(string_3)
```

```
[agatha3@linux5 lab5] python3 fib.py  
Hello  
World  
Hello, World!
```

# While loops

- Run a chunk of code until a condition is false
- Used when we don't generally know how many times that piece of code will run
- Unlike for-loops, it's your responsibility to make sure you don't get in an infinite loop
  - If you get in an infinite loop, use the command **CTRL + C** to quit your running program

# Major differences between while and for loops

- For loops are guaranteed to run for a certain amount of times
  - Usually the length of a list
- While loops generally aren't guaranteed to run a certain amount of times
  - Input validation, inputting items in a list
- For-loops ensure you don't go out of bounds or have an infinite loop
  - You're responsible for everything in while loops

# Activity

- Get file from my public directory
  - `cp /afs/umbc.edu/users/a/g/agatha3/pub/201_grs/lab5/initials.py .`
  - **Don't forget the period at the end!**