

# Strings

CAC 180

Anthony Winchester

Birmingham-Southern College

# Class Today

- Odds and Ends/Review
- Strings
- Practice

What is a variable?

# Rules for Naming Variables

- One word
- Must start with a letter
- Cannot have special characters other than underscore
- Cannot be a keyword
- Should be meaningful
- Should start with a lowercase letter

# Which ones are valid variable names?

1. value

6. 2x

2. my value

7. input

3. VALUE

8. total sum

4. total!

9. 4quarter

5. my\_total

10. y

# Printing Tricks

- What does the comma do in the print statement?
  - `print("The total is", total, ".")`
- What does the plus sign do?
  - `print("The total is " + str(total) + ".")`

# Printing Tricks

- Assuming total is a float, the following line causes an error, why?
  - `print("The total is " + total + ".")`
- The plus sign is *overloaded*. It is used to add numbers (mathematical addition) and strings (concatenation). How should it handle adding a number to a string?
- How do we correct the line?

# Printing Tricks

- Assuming total is a float, the following line causes an error, why?
  - `print("The total is " + total + ".")`
- The plus sign is *overloaded*. It is used to add numbers (mathematical addition) and strings (concatenation). How should it handle adding a number to a string?
- How do we correct the line? Cast the total as a string.
  - `print("The total is " + str(total) + ".")`



What is a string?

# Example

H	E	L	L	O
0	1	2	3	4

- The word 'HELLO' has how many letters? This is referred to as a string's length.
- Notice that the positions go from 0 to 4.
- The positions always start with 0.
- Because strings have individually named positions, programmers can access each individual letter.

# Finding the Length

- To find the length of a string, use *len()*

```
myString = 'pumpkin'  
print('The length of the string is', len(myString))
```

- Why do we care about length?
  - Password verification
  - Field validity: 10 digit phone number
  - When traversing through a string, it sets the boundaries of the loop

# Accessing Elements

- When accessing the individual elements of a string, use square brackets after the name

```
fName = 'Susan'
```

```
mName = 'Rachel'
```

```
lName = 'Smith'
```

```
initials = fName[0] + mName[0] + lName[0]
```

String  
Concatenation

# Let's Think...

- If I have the integer 15, how can I break this down into the tens value and the ones value?
- In other words, I want to be able to find that I have a 1 followed by a 5.
- If it is an integer, I would have to use math to manipulate it.

# Let's Think...

- Assume I have the following:  
`num = int(input('Enter a two digit number: '))`
- Determine individual numbers using math:  
`tens = num // 10`  
`ones = num % 10`  
`print('Your number has a', tens, 'followed by a', ones)`
- Determine individual numbers using string:  
`num = input('Enter a two digit number: ')`  
`print('Your number has a', num[0], 'followed by a', num[1])`

# Immutable

- Strings are immutable.
- Meaning that you can't change the value of a string once you set it.

# Immutable

name = 'Joe Smith'

name

Joe Smith

name = 'Joseph Smith'

name

Joe Smith

Joseph Smith



# Repetition

- In Python, you can multiply strings.
- What do you think would be the outcome of the following?  
word = 'go'  
newWord = word\*3

# Slicing

- This is often referred to as substring
- You use the string followed by [:]
- Example:  
name = 'Joe Smith'  
fname = name[0:3]
- The first number represents the starting index, and the second number represents one past the upper bound of where we want to stop.

# Let's Code

1. Print the first character of a string
2. Print the last character of a string
3. Ask the user to enter separate words and build a string (first name, middle name, last name)
4. Using string repetition, print 'Mississippi'

# For Next Class

- Read Chapter 3 if you haven't already