

Dictionaries

CAC 180

Amber Wagner

Birmingham-Southern College

Class Today

- Dictionaries
- Practice

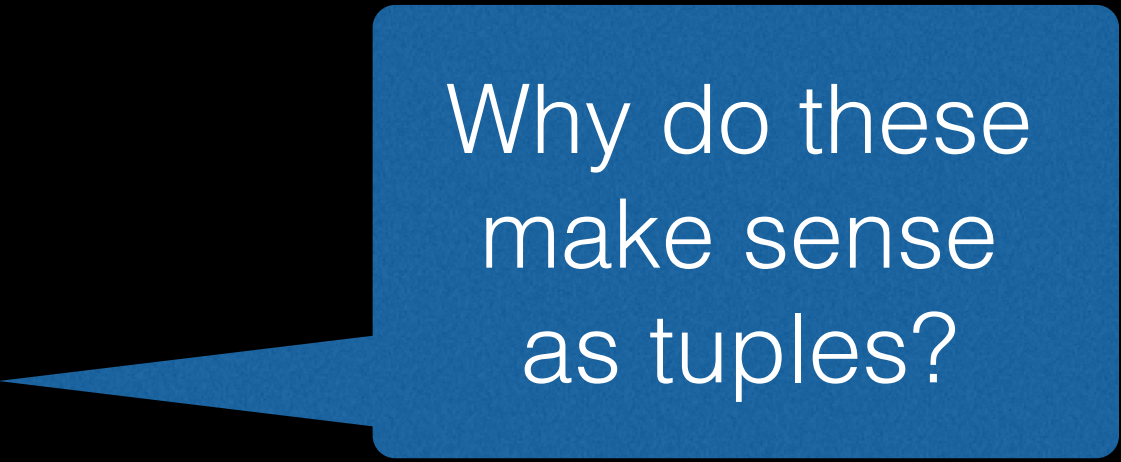
Remind me...



- What is a string?
- How do I reference the third character in a string?
- What is a list?
- How do I know how many elements are in a list?

Tuples

- Similar to a list but immutable (what does that mean again?)
- *variable_name = (val, val, val)*
- Examples:
 - Coordinates
 - RGB values of pixels



Why do these
make sense
as tuples?

Dictionaries

- How does the Oxford dictionary work?
- A dictionary in programming works the same way, there is a key and a value
- *dictionary_name = {key:value, key:value...}*


Example

- I have a list of contacts: names and numbers
- Why would it be a bad idea to store the contacts in two lists...one for names and one for numbers?
- What should the key be for a list of contacts?
- What should the value be?

Names		Numbers
Suzy	→	2055551234
Jack	→	2055552345
Kim	→	2055553456
John	→	2055554567
David	→	2055555678

Example Answers

- I have a list of contacts: names and numbers
- Why would it be a bad idea to store the contacts in two lists...one for names and one for numbers?
 - If you were to delete a name from the name list but forget to delete the associated number, the lists now do not match up.
- What should the key be for a list of contacts?
 - Name
- What should the value be?
 - Number

Names		Numbers
Suzy	→	2055551234
 Mark	↗	2055552345
Kim	↗	2055553456
John	↗	2055554567
David	↗	2055555678

Creating the Dictionary

```
contacts={  
'Suzy':'2055551234',  
'Jack':'2055552345',  
'Kim':'2055553456',  
'John':'2055554567',  
'David':'2055555678'  
}
```

Name	Number
Suzy	2055551234
Jack	2055552345
Kim	2055553456
John	2055554567
David	2055555678

Alternative Method - Helpful for Adding Entries

```
contacts = {}
```

```
contacts['Suzy'] = '2055551234'
```

```
contacts['Jack'] = '2055552345'
```

```
contacts['Kim'] = '2055553456'
```

```
contacts['John'] = '2055554567'
```

```
contacts['David'] = '2055555678'
```

Name	Number
Suzy	2055551234
Jack	2055552345
Kim	2055553456
John	2055554567
David	2055555678

Getting an Item from a Dictionary

- How do we get the phone number for one of our contacts?
- `print(contacts['Suzy'])`
- Dictionaries don't work by position, they work by the key
- Dictionaries are actually hashmaps, can't guarantee order

Updating an Entry

- `contacts['Suzy'] = '2055559876'`

Practice

Will be helpful if you get
the pdf from Moodle

Practice #1

- **Populate a dictionary with the following information (you're welcome to use your own information if you prefer).**
- **Print it to the screen.**
- **Allow the user to edit one player's total yards.**
- **Allow the user to ask for the total yards for one particular player and print as an whole number.**
- **Challenge: For the player above, print the difference between that player's total yards and the average total yards among the five quarterbacks.**
- **Total career passing yards:**
 - **Peyton Manning - 71940**
 - **Brett Favre - 71838**
 - **Drew Brees - 70445**
 - **Tom Brady - 66159**
 - **Dan Marino - 61361**

Practice #2

Morse code is an encoding scheme that uses dashes and dots to represent numbers and letters. In this exercise, you will write a program that uses a dictionary to store the mapping from letters and numbers to Morse code. Use a period to represent a dot, and a hyphen to represent a dash. The mapping from letters and numbers to dashes and dots is shown in the table below.

Your program should read a message from the user. Then it should translate each letter and number in the message to Morse code, leaving a space between each sequence of dashes and dots. Your program should ignore any characters that are not letters or numbers. The Morse code for Hello, World! is shown below:

.... . .-... .-... --- .-- --- .- .-... -..

Letter	Code	Letter	Code	Letter	Code	Number	Code
A	. -	J	. - - -	S	. . .	1	. - - - -
B	- . . .	K	- . -	T	-	2	. . - - -
C	- . - .	L	. - . .	U	. . -	3	. . . - -
D	- . .	M	- -	V	. . . -	4 -
E	.	N	- .	W	. - -	5
F	. . - .	O	- - - -	X	- . . -	6	-
G	- - .	P	. - - .	Y	- . - -	7	- - . . .
H	Q	- - . -	Z	- - . .	8	- - - . .
I	. .	R	. - .	0	- - - - -	9	- - - - .