Dictionaries!

- Purpose
- Initialization
- Using lists
- keys() and values()
What is a dictionary?

• Lists only store a sequence of data
• Often, we want to associate “keys” with “values”
• How would we actually store a class' test scores?
  – johnny = 85
  – yifan = 79
  – gloria = 90
  – ...

The empty dictionary

```python
>>> nothing = {}
```

```python
>>> nothing
{}
```
What is a dictionary?

```python
>>> # Also called “association lists”, “hashes”, “maps”, etc

>>> grades = {
    'johnny': 85,
    'yifan': 79,
    'gloria': 90,  # trailing comma,

}  # trailing comma

>>> grades
{'yifan': 79, 'gloria': 90, 'johnny': 85}

>>> grades['yifan']
79
```
Using dictionaries

>>> grades = {
    'johnny': 85,
    'yifan': 79,
    'gloria': 90,
}

>>> # Add an item to a dictionary
>>> grades['salem'] = 53
>>> grades
{'yifan': 79, 'salem': 53, 'gloria': 90, 'johnny': 85}
Using dictionaries

>>> # Be careful with KeyValue errors

>>> grades['elana']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'elana'

>>> # Safer:
>>> grades.get('elana')

>>>
Using dictionaries

```python
>>> grades = {
    'johnny': 85,
    'yifan': 79,
    'gloria': 90,
    'salem': 53,
}

>>> # Change a value in the dictionary
>>> grades['salem']
53
>>> grades['salem'] = 100
>>> grades['salem']
100
```
keys() and values()

```python
>>> grades = {
    'johnny': 85,
    'yifan': 79,
    'gloria': 90,
    'salem': 100,
}

>>> # Get a list of the keys in the dict
>>> grades.keys()
['yifan', 'salem', 'gloria', 'johnny']

>>> # Get a list of the values in the dict
>>> grades.values()
[79, 100, 90, 85]
```
keys() and values()

```python
>>> # Class average
>>> grade_list = grades.values()
>>> sum = 0

>>> for grade in grade_list:
...   sum += grade

>>> sum
354
>>> print(sum / len(grade_list))
88
```
Modules

- Purpose
- Built-ins
- Importing
- The random library
What is a module?

- A unified body of code someone else wrote
- We don't want to have to write everything ourselves!
  - Philosophy behind free software
- Libraries: contain many modules
- Python has many built-in modules
Built-in Modules and More

• Built-in
  - random: random numbers
  - urllib: utilities for working with URLs
    >>> import urllib
    >>> urllib.quote('url.with.a.space/here yay')
    'url.with.a.space/here%20yay'

• Other
  - matplotlib: draw graphs and plots!
  - List: https://wiki.python.org/moin/UsefulModules
Using a module

>>> random.randint(0,5)
Traceback (most recent call last):
  File "<stdin>" , line 1, in <module>
NameError: name 'random' is not defined

>>> # Make sure we have the code available!
>>> import random
>>> random.randint(0,5)
5
>>> random.randint(0,5)
4
Using a module

```python
>>> provinces = ['AB', 'BC', 'MB', 'NB', 'NL', 'NS',
    'NT', 'NU', 'ON', 'PE', 'QC', 'SK', 'YT', 'AK']

>>> # Documentation for modules online!
>>> # https://docs.python.org/2/library/random.html
>>> import random
>>> random.choice(provinces)
'NL'
>>> random.randint(0,5)
'ON'
```
Putting it all together

- Demo of `provincial_capitals.py`
  - Follow along on your own computer
  - Make sure to ask questions!

- Break for lunch afterwards!

- Return from lunch for 12:45 PM