CSCA08

Joshua Concon, Albion Fung
github.com/conanap

Lists

- Literally a list of things, like a todo list
- Index starts at 0, ends at number of elements - 1

```
>>> alst = [1, ‘waterdose’, False]

>>> a[2]

False
```

```
>>> bst = [] // empty list

>>> bst.append(1)

>>> print(bst)

[1]

>>>bst[0] = ‘hi’

>>> print(bst)

[‘hi’]
```

List addition

- You can add list to lists and it becomes 1 big list

```
>>> a = [1, 2, 3]

>>> b = [4, 5, 6]

>>> c = a + b

>>> print(c)

[1, 2, 3, 4, 5, 6]
```

Mutability

- Lists are mutable - you can change the values in the list
- Anywhere you pass a list and you modify it, the original list is modified too

```
def mutate(a):
  a[0] = 1

>>> b = [9, 8, 7]
>>> mutate(b)
>>> print(b)
[0, 8, 7]
```

Tuples

- Ordered elements
- Not mutable

>>> tups = (1, ‘hi’, True) // can be as long as you want

>>> tups[0]

1

>>> tups[0] = 1 # crashes

- Used very often to swap values

(a, b) = (b, a)

Sets

- An unordered list - you can’t tell what will come next
- Automatically removes repeated elements
- Useful when every element has to be unique and used in a loop
- Mutable

```
>>> a = {} # empty dict not set!

>>>a = set() # this is an empty set

>>> a = {1, 2, 3}

>>> a.append(4)

>>> print(a)

{1, 2, 4, 3} # notice orders are randomized

>>> print(a)
{2, 3, 1, 4}
```

```
def notOddUnder10(num):
    acceptable = {1, 3, 5, 7, 9}
    for i in acceptable:
        if i == num:
            return False

return True
```

Dictionaries

- Like lists, except unordered and we use whatever we want as the key

- In list, suppose we have list a. The keys are integers. a[0], a[100], a[420]

- Use whatever you want in dicts!

```
>>> a = {} # empty dict

>>> a = dict() # also an empty dict

>>> a = {‘Gender’ : ‘Attack helicopter’, ‘isSmart’ : True, 13 : ‘What you doin’}

>>> a[‘name’] = ‘Albion’

>>> print(a)

>>> {‘Gender’ : ‘Attack helicopter’, ‘name’ : ‘Albion’, ’isSmart’ : True, 13 : ‘What you doin’}

>>> a[‘Gender’]

‘Attack helicopter’

>>> a[13]

‘What you doin'
```

Files

- You can open, read, write or append to a file
- append means to add to the end of the file
- write means erase the original file and write whatever you want to put in there

File modes

- r means read
- w means write
- a means append

```
# do it like this: file = open(pathOfFile, mode)

# opening ./example/readme.txt for reading

file = open('./example/readme.txt', 'r')

for i in file: … # returns empty string at EOF

line = file.readline() ….

lines = file.readlines()…
```

```
# open file for writing
file = open('./example/readme.txt', 'w')

file.write("A rare pepe appeared")
```

appending

```
file = open('./example/readme.txt', 'a')

file.write('Look! Another rare pepe!')
```

Tracing