Strings: the Basics

Strings: What We Already Know

- Strings are sequences: order is important
  - indexing, slicing
  - looping over characters in a string
  - concatenation, len(), etc.
- Strings are immutable: they do not change
  - no use of append, etc.
  - cannot change values via index re-assignment, etc.
- Strings can occur in boolean statements
  - in test
  - alphabetic checks (<, >, etc.)

String Formatting

Basic Placeholders

- Strings can have placeholders, the values in placed in .format():

```python
s = "you're eating {:0}".format("crazy_cheese")
# s = "you're eating crazy cheese"
```

- We can do that with variables, too:

```python
location = 'paris'
s = "you would think I'm from {}".format(location)
# s = "you would think I'm from paris"
```

- And with more than one value:

```python
lyric = "you know I'm getting {}, you think I'm getting {}".
adj = ("fly", "high")
print(lyric.format(adj))
```

(See: https://docs.python.org/3.1/library/string.html, sec. 7.1.3.2)

Conversion Types

```python
s = 'we have {:4d} MCs and {:4d} DJ'.format(3,1)
# s = 'we have 3 MCs and 1 DJ'
```

```python
r = 'we have {:4.1f} MCs and {:4.2f} DJ'.format(3.1, 1.0)
# r = 'we have 3.0 MCs and 1.00 DJ'
```

```python
d = 'we have {} MCs and {} DJ'.format(3, 1)
# d = 'we have 3 MCs and 1 DJ'
```

```python
f = 'we have {} MCs and {} DJ'.format(3.1, 1.0)
# f = 'we have 3.0 MCs and 1.00 DJ'
```

This is the real "win" in string formatting: easy integration of different kinds of information

```python
s = 'we have {:4d} MCs and {:4.2f} DJ'.format(3, 1)
```
### find

**Example**

- Find where a string starts (cf. `index()` for lists)

```python
phrase="the reason that you're smilin'
phrase.find('son')  # 7
phrase.find('smile')  # -1
if phrase.find('you')>=0:
    print("me!")
```

- not: `find` does NOT return a Boolean value: if it does not find the substring, it returns `-1`

### Join and Split

1. Split the haystack phrase into multiple words

```python
words=phrase.split()
```

- `.split()` can take an argument, namely the thing you want to split on (default=whitespace)

2. Reverse the order of the words

```python
words.reverse()
```

3. Join the words back together with commas

```python
', '.join(words)
```

### Changing Case

1. Make a string all lowercase

```
'SMILIN'.lower()
```

2. Make a string all uppercase

```
'wildin'.upper()
```

3. Make all but the first letter of a string lowercase

```
'LISTEN'.title()
```

### replace

1. Replace `smilin` with `frownin` in the phrase

```python
phrase.replace('smilin', 'frownin')
```

2. Replace `e` with `o` in the phrase

```python
phrase=phrase.replace('e', 'o')
```

### strip

1. Strip off newline characters from end of the phrase

```python
phrase=phrase.strip('\r\n')
```

2. Strip off any leading or trailing whitespace from the phrase, and convert to upper case

```python
phrase=phrase.strip().upper()
```

3. Strip off any leading or trailing whitespace from the haystack phrase, replace `smilin` with `frownin` and convert to upper case

```python
phrase=phrase.strip().replace('smilin', 'frownin').upper()
```