More on Lists and Strings

Python

Slices

- We already know two sequence types: lists and strings
  - Sequences can be sliced: A slice is a new object of the same type, consisting of a subsequence
  - Use a bracket cum colon notation to define slices.
    - sequence[a:b] are all elements starting with index a and stoping before index b.

Slices

- String slices
  - Number before colon:
    - Start
  - Number after colon:
    - Stop
  - Default value before colon:
    - Start with first character
  - Default value after colon
    - End with the string

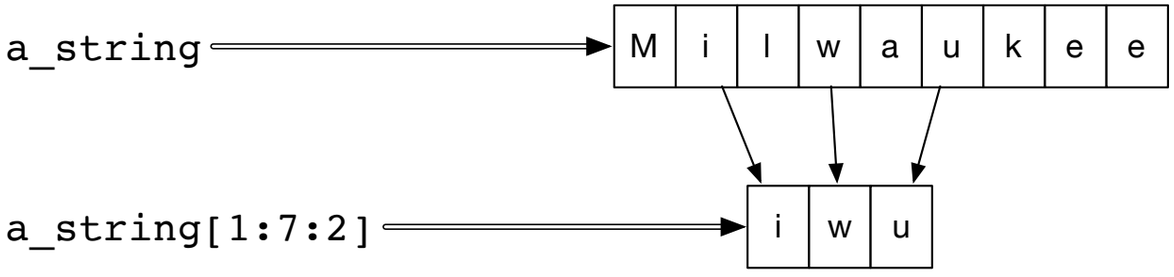
```
>>> a_string = "Milwaukee"
>>> a_string[3:6]
'wau'
>>> a_string[1:5]
'ilwa'
>>> a_string[:6]
'Milwau'
>>> a_string[4:]
'aukee'
```

Slices

- String slices:
  - Optional third parameter is Stride
    - First character is character 1
  - Next one is character 1+2
  - Next one is character 1+2+2
  - Next one would be character 1+2+2+2, but that one is >= the stop value.

```
>>> a_string = "Milwaukee"
>>> a_string[1:7:2]
'iwu'
```

0 1 2 3 4 5 6 7 8



start value is index 1

stop value is index 7

stride is 2

Slices

- Negative strides are allowed.
  - Create a new string that is reversed using default values

```
>>> a_string = "Milwaukee"
>>> b_string = a_string[::-1]
>>> b_string
'eekuawliM'
```

Slices

- Negative strides are allowed

```
>>> a_string = "Ahmedabad, Gujarat, India"
>>> a_string[20:3:-3]
'ItaGda'
```

- Character 20 is “I” of India
- Next character is 17, the “t” in Gujarat
- Stop before character 3 (the fourth character)

Ahmedabad, Gujarat, India

Lists and Strings

- Both lists and strings are sequences
  - Length: `len(a_string), len(a_list)`
  - Concatenation: `a_string + b_string, a_list + b_list`
  - Repetition: `3*a_string, 3*a_list`
  - Membership: `if 'x' in a_string, if a in a_list`
  - Iteration: `for ele in a_string, for ele in a_list`

Lists and Strings

- Strings are immutable

- Lists are mutable