

CS 106
Introduction to Computer Science I

07 / 09 / 2021

Instructor: Michael Eckmann

Today's Topics

- Questions / comments?
- int (no maximum), float (limited precision)
- str methods etc.
- Lists
- Another function in random module --- choice
- For loops

Int and float

- There is no maximum int in python3 --- it is only restricted to available memory space to store it
- e.g.
- `21 ** 1000`
- But float has limits
- e.g.
- `21 ** 1000.0`

str methods/operations

- We already saw how to concatenate strings together using +
- Python has some powerful tools that allow us to easily do things with strings (like search them, make them all uppercase, find their length, etc.)
- Do `help(str)` as well as look at documentation for string methods on the web.
- Let's look at a sampling of the available str methods:

capitalize

endswith

lower

upper

find, rfind

replace

- Note: methods are a lot like functions except we precede them with a variable or a value (in this case a variable or value of type str)

str methods/operations

- Examples:

```
word = 'science'
```

```
word1 = word.upper()
```

```
# returns SCIENCE and assigns it to word1,
```

```
# word remains unchanged
```

```
word2 = word.replace('s','$')
```

```
# returns $cience and assigns it to word2
```

str methods/operations

- in, not in, *, len(), min(), max(), []
- Examples:

word = 'science'

len(word) # returns 7

word[0] # returns 's'

min(word) # returns 'c'

lists

- Another builtin type in python is a list
- Lists store data in a linear fashion
- Lists are mutable (can change values, add or remove them, etc.)
- An example list: [45, 26, 89, 32]
- List values are specified as comma separated values inside square brackets
- Each list element has an associated index. Indices start at 0 and end at length of the list – 1
- The list: [45, 26, 89, 32] has 45 in the 0 index, 26 in the 1 index, 89 in the 2 index and 32 in the 3 index and the length of that list is 4. Notice the valid indices go 0 .. (4-1)=3

lists

- Values of different types may be stored in the same list, but it is usually most useful to store values of the same type within a list.
- e.g. `mylist = ['a', 34, 32.0, False, 45, -2.3, 'hello']`

lists

indices within [] (and negative indices)

slices using :

len function

Some useful list methods:

append

remove

insert

clear

count

index

reverse

try:

help(list.append)

help(list.remove)

help(list.count) etc.

Another random function

- The function `choice` in the `random` module returns a random element from a sequence.
- Lists are sequences
- so are strings and tuples – which we have yet to learn
- e.g. `random.choice([43,23,22,-233,95])` will return a random element from the list
- Because `choice` works on a sequence and a string is a sequence, we can do this:
`random.choice('abcdefg')` to return a random letter from that string

For loop

- Like the while loop we learned recently, a for loop is code that repeats lines of code (iterates over the body of a loop) 0 or more times.
- The syntax is

```
for ____ in ____:  
    # body of the loop  
    # 2nd line in body
```

- The second ____ is a sequence (e.g. a list or a string). The first ____ is a variable name that gets set to an element of the sequence each iteration of the loop.

For loop

- Example:

```
for item in [5,10,15,20]:  
    print(item)
```

- The code above would print each element of the list

For loop

- Example:

```
for char in 'Skidmore':  
    print(char)
```

- The code above would print each character in the string 'Skidmore'

For loop with range

- Range returns an object that produces a sequence of integers.
- Range can be called in 3 ways:
 - `range(stop)`
 - `range(start, stop)`
 - `range(start, stop, step)`
 - Each of these produces a sequence of ints starting at start and ending just before stop (that is, stop is not inclusive)
 - In first call above, start defaults to 0
 - step in third call changes the default of 1 to what is passed in

```
for i in range(10):  
    print(i)
```

- 0,1,2,3,4,5,6,7,8, and 9 will print

For loop with range

```
for i in range(5,10):  
    print(i)
```

5,6,7,8, and 9 will print

```
for i in range(1,20,3):  
    print(i)
```

1, 4, 7, 10, 13, 16, 19 prints