Python

Lists

Data Types

Data Types

- Primitive
- Collection

Data Types

- 29.5
- 'a'
- 23954
- 'Four and twenty black birds'
- [3, “Smoot”, 'Kilgore']

List Data Type

- [3, “Smoot”, 'Kilgore']

Lists

- Consist of Elements
  - ordered - like strings
  - values IDd by Index – like strings
  - different types – unlike strings
- can be Nested
  - called Sublist

Accessing Elements

- blah = [1, '1', 'one']
- len(blah)
- blah[1]

You Try It

```
blah = [1, '1', 'one']
print( len(blah) )
print( blah[1] )
print( blah[2] )
```

How would you refer to the n in one?

Accessing Elements

- blahT = [1, '1', [2, '2', 'two], 3]
- len(blahT)
- blahT[3]
- blahT[2][2]

You Try It

- blahT = [1, '1', [2, '2', 'two'], 3]
- len(blahT)
- blahT[3]
- blahT[2][2]

How would you refer to the w in two?

Lists

- List Membership – in, not in

```
blahT = [1, '1', [2, '2', 'two'], 3]
for item in blahT:
    print(item)
```

Lists

- List Mutability – myList[2] = 22
  - unlike strings individual items can be modified

Objects and References

- Aliasing
  - same list with two names
- a = [1, 2, 3]
- b = a
- b[2] = 67
- print(a)

- Cloning
  - copy of original
- a = [1, 2, 3]
- b = a[:]
- b[2] = 67
- print(a)

Repetition and References

```
origlist = [45, 76, 34, 55]
print(origlist * 3)
newlist = [origlist] * 3
print(newlist)
nothernewlist = origlist * 3
origlist[1] = 99
print(newlist)
print(nothernewlist)
```

List Methods

append	item	mutator	Adds a new item to the end of a list
insert	position, item	mutator	Inserts a new item at the given position
pop	none	hybrid	Removes and returns the last item
pop	position	hybrid	Removes and returns the item at position
sort	none	mutator	Modifies a list to be sorted
reverse	none	mutator	Modifies a list to be

List Methods

append	item	mutator	Adds a new item to the end of a list
insert	position, item	mutator	Inserts a new item at the given position
pop	none	hybrid	Removes and returns the last item
pop	position	hybrid	Removes and returns the item at position
sort	none	mutator	Modifies a list to be sorted
reverse	none	mutator	Modifies a list to be

```
mylist = mylist.sort()   #probably an error
print(mylist)
```

You Try It

- woof = [22, 3, 1, 33, 21, 24]
- print(woof)
- woof.sort()
- print(woof)
- woof = [22, 3, 1, 33, 21, 24]
- woof = woof.sort()
- print(woof)

Append v Concatenate

- list = [45, 22, 78]
- list.append(“cat”)

- list = list + [“cat”]
- list = list + [999]

Append v Concatenate

- list = [45, 22, 78]
- list.append(“cat”)

- list = list + [“cat”]
- list = list + [999]

ONLY
lists can be
concatenated
to lists

Lists and For Loops

```
for blah in list:
        print(blah)

for count in range(len(list)):
        print(list[count])
```

Nested Lists

```
nested = ["hello", 2.0, 5, [10, 20]]
innerlist = nested[3]
print(innerlist)
item = innerlist[1]
print(item)

print(nested[3][1])
```

split & join

```
quote = “fly you fools”
words = quote.split()
print(type(words))
words = quote.split('y')
print(words)

glue = ' '
new = glue.join(words)
print(type(new))
print(“”.join(words))
```

list Type Conversion

```
sc = list(“Crunchy Frog”)
print(sc)
x = 123
print( list(x) )
```

list Type Conversion

```
sc = list(“Crunchy Frog”)
print(sc)
x = 123
print( list(x) )
print( list(str(x)) )
```


That's All Folks