CPE101
List

Winter 2019
@ Cal Poly SLO
By
Toshi

Learning Objectives

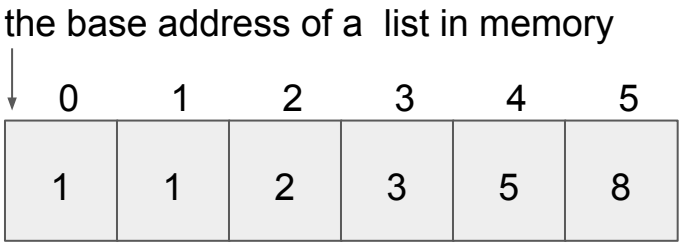
1. Lists

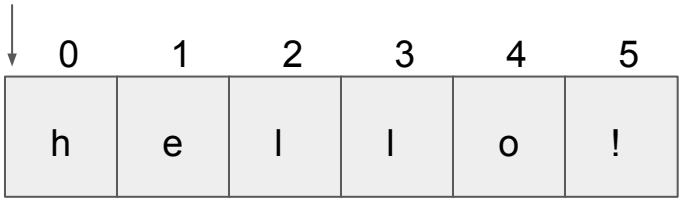
2. List Comprehensions (sadly not covered in the linked book)

Lists (Arrays)

```
fib = [1,1,2,3,5,8]
print len(fib) # 6
print fib[0] # 1
print fib[1] # 1
print fib[2] # 2
print fib[3] # 3
print fib[4] # 5
print fib[5] # 8

greeting = [‘h’, ‘e’, ‘l’, ‘l’, ‘o’, ‘!’]
Print len(greeting) # 6
print greeting[0] # h
print greeting[1] # e
print greeting[2] # l
print greeting[3] # l
print greeting[4] # o
print greeting[5] # !
```





Accessing elements in list

```
fib = [1,1,2,3,5,8]

# get the element at index 1
n = fib[1] # 1

# get the first element
n = fib[0] # 1

#get the last element
n = fib[-1] # 8
```

Adding Elements to list

```
mylist = [‘a’, ‘b’, ‘c’, ‘d’, ‘e’]

mylist.append(‘f’) # adding ‘f’ at the end
print(mylist) # [‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’]
mylist.insert(0, ‘$’) # adding ‘$’ to the front

print(mylist) # [‘$’, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’]
mylist.insert(2, ‘$’) # adding ‘$’ at index 2

print(mylist) # [‘$’, ‘a’, ‘$’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’]
c = mylist.count(‘$’) # returns the count of ‘$’
print(c) # 2
```

Assigning values to elements in list

```
mylist = [1,2,3,4,5]

# change the value at index 0
mylist[0] = 0
print(mylist) # [0,2,3,4,5]

# In Python all the elements in a list does not need to be values of the same type
mylist[4] = “end”
print(mylist)  # [0,2,3,4,’end’]

# you can assign an object
mylist[1] = Point(1,1) # [0,(1,1),3,4,’end’]

# you can assign a list
mylist[2] = [1,2,3]  #  [0,(1,1),[1,2,3],4,’end’]
```

Removing

```
mylist = [‘a’, ‘b’, ‘c’, ‘d’, ‘e’]

popped = mylist.pop() # pop an item from the end
print(popped) # ‘e’

print(mylist) # [‘a’, ‘b’, ‘c’, ‘d’]

popped = mylist.pop(1) # specify the index to be removed

print(popped) # ‘b’
print(mylist) # [‘a’, ‘c’, ‘d’]
mylist.remove(‘a’) #remove the first item from the list whose value is ‘a’
print(mylist) # [‘c’, ‘d’]
```

Slicing

```
Mylist = [‘a’, ‘b’, ‘c’, ‘d’, ‘e’]

sublist = mylist[1:3] #splice from index 1 to 2
print(sublist) # [b’, ‘c’]

print(mylist) # [‘a’, ‘b’, ‘c’, ‘d’, ‘e’]

last_item = mylist[-1] # get the last item in the list

print(last_item) # ‘e’
print(mylist[:2]) # [‘a’, ‘b’]
print(mylist[3:]) # [‘d’, ‘e’]
```

More List Operations

```
fib = [1,1,2,3,5,8]


# concatenation
new_list = fib[1:3] + fib[4:]
print new_list # [1,2,5,8]

# what about multiplication?
new_list = [‘a’, ‘b’, ‘c’] * 2
print new_list # [‘a’,’b’,’c’,’a’,’b’,’c’]
```

Example

Create a list of points on the line y = x

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
        return “Point(%s, %s)”\
            % (self.x, self.y)

    def __eq__(self, other):
        return type(other) == Point\
            and other.x == self.x\
            and other.y == self.y
```

```
def main():
    points = []
    for i in range(10):
        points.append(Point(i, i))


    print(points)
```

List Comprehensions

```
#python provides a concise way to create a new list from a list
numbers = [1,2,3,4,5,6,7,8,9,10]
#collect only odd numbers
odds = [n for n in numbers if n % 2 != 0] # odds = [1,3,5,7,9]

#you can also append a value to a list
odds = [] # an empty list
#iterate over numbers from 0 to 9
for n in range(10):
    #if odd number
    if n % 2 != 0:
        odds.append(n)
print(odds) #[1,3,5,7,9]
```

List Comprehensions Continued

```
# Create a function and name it double:
def double(x):
  return x*2

# If you now just print that function with a value in it, it should look like this:
print(double(10))  # 20

 [double(x) for x in range(10)] # [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

# You can put in conditions:
 [double(x) for x in range(10) if x%2==0]  # [0, 4, 8, 12, 16]
```