More on Lists

Traversing a List

Use a *for loop* to read the elements of a list.

```
cpsc115 = ['andy', 'betty', 'carol']
for student in cpsc115:
    print student
```

Traversing a List

Use a *for loop* to read the elements of a list.

```
cpsc115 = ['andy', 'betty', 'carol']
for student in cpsc115:
    print student

Output:
andy
betty
carol
```

Traversing a List

Use a *for loop* to read the elements of a list.

```
cpsc115 = []
for student in cpsc115:
    print student
```

Traversing a List

Use a *for loop* to read the elements of a list.

```
cpsc110 = []
for student in cpsc115:
    print student

Output:
```

Traversing a List

Use *indexes* to update elements of a list.

```
cpsc115 = ['andy', 'betty', 'carol']
for i in range(len(cpsc115)):
    cpsc115[i] = cpsc115[i]+'@trincoll.edu'
print cpsc115
```

Traversing a List

Use *indexes* to update elements of a list.

```
cpsc115 = ['andy', 'betty', 'carol']
for i in range(len(cpsc115)):
    cpsc115[i] = cpsc115[i]+'@trincoll.edu'
print cpsc115
```

Output:

```
['andy@trincoll.edu','betty@trincoll.edu','carol@trincoll.edu']
```

Concatenating Lists

```
>>> b = [1,2,3,4]
>>> c = [4,5,6,7]
>>> a = b + c
```

Concatenating Lists

```
>>> b = [1,2,3,4]
>>> c = [4,5,6,7]
>>> a = b + c
>>> print a
[1,2,3,4,4,5,6,7]
```

Multiplying Lists

```
>>> b = [0,1]
>>> a = b * 3
```

Multiplying Lists

```
>>> b = [0,1]
>>> a = b * 3
>>> print a
[0,1,0,1,0,1]
```

Slices

```
>>> t = [0,1,2,3,4,5,6,7,8]
>>> t[1:3]
[1,2]
>>> t[:4]
[0,1,2,3]
>>> t[3:]
[3,4,5,6,7,8]
>>> t[:]
[0,1,2,3,4,5,6,7,8]
```

Assigning Slices

```
>>> t = [0,1,2,3,4,5,6,7,8]
>>> t[1:3] = [0,0]
[0,0,0,3,4,5,6,7,8]
>>> t[1:3] = [1,1,1,1]
[0,1,1,1,1,3,4,5,6,7,8]]
```

Appending

To *append* an element to a list,

```
>>> a = []
```

Appending

To *append* an element to a list,

```
>>> a = []
>>> a.append(1)
```

Appending

To *append* an element to a list,

```
>>> a = []
>>> a.append(1)
>>> a.append(2)
```

Appending

To *append* an element to a list,

```
>>> a = []
>>> a.append(1)
>>> a.append(2)
>>> print a
```

Lists

To *append* an element to a list,

```
>>> a = []
>>> a.append(1)
>>> a.append(2)
>>> print a
[1, 2]
```

Appending

To *append* a list to another list,

Appending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
```

Appending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
```

Appending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
>>> a.append(b)
```

Appending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
>>> a.append(b)
>>> print a
```

Appending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
>>> a.append(b)
>>> print a
[1, 2, 3, [4, 5]]
```

Extending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
>>> a.extend(b)
>>> print a
```

Extending

To *append* a list to another list,

```
>>> a = [1, 2, 3]
>>> b = [4, 5]
>>> a.extend(b)
>>> print a
[1, 2, 3, 4, 5]
```

Reduce Operation

Reduce a list to a single element, e.g.,

```
def add_all(t):
    total = 0
    for x in t:
        total += x
    return total

sum = add_all([1,2,3,4])
print sum
```

Reduce Operation

Reduce a list to a single element, e.g.,

```
def add_all(t):
    total = 0
    for x in t:
        total += x
    return total

sum = add_all([1,2,3,4])
print sum
10
```

Map Operation

Map a function onto all elements of a list giving a new list.

```
def capitalize_all(t):
    res = []
    for s in t:
        res.append(s.capitalize())
    return res

b = capitalize_all(['alpha', 'beta'])
print b
```

Map Operation

Map a function onto all elements of a list giving a new list.

```
def capitalize_all(t):
    res = []
    for s in t:
        res.append(s.capitalize())
    return res

b = capitalize_all(['alpha', 'beta'])
print b
['Alpha', 'Beta']
```

Filter Operation

Filter a list into a sublist.

```
def only_upper(t):
    res = []
    for s in t:
        if s.isupper():
            res.append(s)
    return res
b = only_upper(['a','B','C','d'])
print b
```

Inclass Exercises

- Reduce: Write a function *max()* that returns the largest element in a list of numbers.
- Map: Write a function *squarelist()* that squares each element of a list of numbers.
- Filter: Write a function *oddlist()* that returns a list of the odd elements of a list of numbers.

Filter Operation

Filter a list into a sublist.

```
def only_upper(t):
    res = []
    for s in t:
        if s.isupper():
            res.append(s)
    return res
b = only_upper(['a','B','C','d'])
print b
['B','C']
```