More Lists

Loops, List methods, Nested lists, etc.

Data types for multiple values

- Lists
  - The basic solution
- Tuples
  - Faster and safer, but less malleable
- Arrays/Matrices
  - Built on lists, but with a number of properties that make them good for doing science

Looping

for item in **list**:

    block

```
bases = ['A', 'T', 'G', 'C']
for base in bases:
    print base
A
T
G
C
```

Methods

- Methods
  - Groups of functions that variables of a certain type carry around with them
- String methods

```
>>> dna = 'attggc'
>>> dan.upper()
'ATTGGC'
>>> dna.find('gg')
3
```

List Methods

- Add a new value to the end of a list
  - listname.append(new_value)
- Add a new value at a particular position
  - listname.insert(position, new_value)

- These (but not all) methods actually change the variable

```
>>> life_list = ['cardinal', 'bluejay']
>>> life_list.append('sparrow')
>>> print life_list
['cardinal', 'bluejay', 'sparrow']
```

List methods

- They also **return None**

```
>>> print life_list
['cardinal', 'bluejay', 'sparrow']
>>> print life_list.append('robin')
None
>>> print life_list
['cardinal', 'bluejay', 'sparrow',
  'robin']
```

List methods

- Don't do this

```
>>> print life_list
['cardinal', 'bluejay', 'sparrow',
  'robin']
>>> life_list =
  life_list.append('chickadee')

>>> print life_list
```

List methods

- Don't do this

```
>>> print life_list
['cardinal', 'bluejay', 'sparrow', 'robin']
>>> life_list =
  life_list.append('chickadee')

>>> print life_list
None
```

- Demonstration

List methods

- Index
  - list.index(x)
  - Returns the position of the first item whose value is x

```
>>> bases = ['a', 'g', 't', 'c']
>>> bases.index('t')
2
```

Nested lists

- Lists can hold any type of variable
- Therefore they can also hold lists

```
>>> stop_codons = [['Ochre', 'UAA'], ['Amber', 'UAG'], ['Opal', 'UGA']]
>>> stop_codons[0]
['Ochre', 'UAA']
>>> stop_codons[0][1]
'UAA'
```

Nested lists

- Functions and methods for lists work on nested lists as well

```
>>> stop_codons = [['Ochre', 'UAA'], ['Amber', 'UAG'], ['Opal', 'UGA']]
>>> del stop_codons[0]
>>> stop_codons
[['Amber', 'UAG'], ['Opal', 'UGA']]
```