The Sequence Abstraction

- **red**, **orange**, **yellow**, **green**, **blue**, **indigo**, **violet**.
- 0, 1, 2, 3, 4, 5, 6.

There isn't just one sequence type (in Python or in general). This abstraction is a collection of behaviors:

**Length.** A sequence has a finite length.

**Element selection.** A sequence has an element corresponding to any non-negative integer index less than its length, starting at 0 for the first element.

The sequence abstraction is shared among several types.

---

Tuples are Sequences

(Demo)

Box-and-Pointer Notation for Nested Pairs

(1, 2)

Every object is an arrow pointing to a box

Boxes for tuples have multiple parts

((1, 2), (3, 4))

(1, 2) tuple

The Closure Property of Data Types

- A method for combining data values satisfies the closure property if:
  - The result of combination can itself be combined using the same method.
  - Closure is the key to power in any means of combination because it permits us to create hierarchical structures.
  - Hierarchical structures are made up of parts, which themselves are made up of parts, and so on.

Tuples can contain tuples as elements
Recursive Lists

Constructor:
```python
def make_rlist(first, rest):
    """Make a recursive list from its first element and the rest."""
```

Selectors:
```python
def first(s):
    """Return the first element of a recursive list s."""
def rest(s):
    """Return the rest of the elements of a recursive list s."""
```

Behavior condition(s):
If a recursive list $s$ was constructed from first element $f$ and recursive list $r$, then
- $\text{first}(s)$ returns $f$, and
- $\text{rest}(s)$ returns $r$, which is a recursive list.

Implementing the Sequence Abstraction

```python
def len_rlist(s):
    """Return the length of recursive list s."""
    length = 0
    while s != empty_rlist:
        s, length = rest(s), length + 1
    return length

def getitem_rlist(s, i):
    """Return the element at index i of recursive list s."""
    while i > 0:
        s, i = rest(s), i - 1
    return first(s)
```

Length. A sequence has a finite length.
Element selection. A sequence has an element corresponding to any non-negative integer index less than its length, starting at 0 for the first element.
def count(s, value):
    total = 0
    for elem in s:
        if elem == value:
            total = total + 1
    return total

For Statement Execution Procedure

for <name> in <expression>:
    <suite>

1. Evaluate the header <expression>, which must yield an iterable value.

2. For each element value in that sequence, in order:
   A. Bind <name> to that value in the local environment.
   B. Execute the <suite>.  

Name bound in the first frame of the current environment