Examples: Lists

Data Examples

Announcements

Lists in Environment Diagrams

Assume that before each example below we execute:

\[ s = [2, 3] \]

\[ t = [5, 6] \]

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>append adds one element to a list</td>
<td>s.append(t)</td>
<td>t = 0</td>
</tr>
<tr>
<td></td>
<td>s = [2, 3, [5, 6]]</td>
<td>t = 0</td>
</tr>
<tr>
<td>extend adds all elements in one list to another list</td>
<td>s.extend(t)</td>
<td>t[1] = 0</td>
</tr>
<tr>
<td></td>
<td>s = [2, 3, 5, 6]</td>
<td>t = [5, 0]</td>
</tr>
<tr>
<td>addition &amp; slicing create new lists containing existing elements</td>
<td>a = s + [t]</td>
<td>b = a[1];</td>
</tr>
<tr>
<td></td>
<td>a = [2, 3]</td>
<td>b = [5, 0];</td>
</tr>
<tr>
<td></td>
<td>a[1] = 9</td>
<td>b[1][1] = 0</td>
</tr>
<tr>
<td></td>
<td>a = [2, 9, [5, 0]]</td>
<td>b = [3, 5, 0];</td>
</tr>
</tbody>
</table>
Lists in Environment Diagrams

Assume that before each example below we execute:

\[ s = [2, 3] \]
\[ t = [5, 6] \]

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>append</strong></td>
<td>[ t = s.append(t) ]</td>
<td>[ s = [2, 3, [5, 6]] ]</td>
</tr>
<tr>
<td><strong>extend</strong></td>
<td>[ t = s.extend(t) ]</td>
<td>[ s = [2, 3, 5, 6] ]</td>
</tr>
<tr>
<td><strong>addition</strong> &amp; <strong>slicing</strong></td>
<td>[ a = s + [t] ]</td>
<td>[ a = [2, 3, [5, 6]] ]</td>
</tr>
<tr>
<td><strong>slice assignment</strong></td>
<td>[ t = [0:0] = [5, 6] ]</td>
<td>[ t = [2, 3] ]</td>
</tr>
</tbody>
</table>

Lists in Environment Diagrams

Assume that before each example below we execute:

\[ s = [2, 3] \]
\[ t = [5, 6] \]

<table>
<thead>
<tr>
<th>Operation</th>
<th>Example</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>append</strong></td>
<td>[ t = s.append(t) ]</td>
<td>[ s = [2, 3, [5, 6]] ]</td>
</tr>
<tr>
<td><strong>extend</strong></td>
<td>[ t = s.extend(t) ]</td>
<td>[ s = [2, 3, 5, 6] ]</td>
</tr>
<tr>
<td><strong>addition</strong> &amp; <strong>slicing</strong></td>
<td>[ a = s + [t] ]</td>
<td>[ a = [2, 3, [5, 6]] ]</td>
</tr>
<tr>
<td><strong>slice assignment</strong></td>
<td>[ t = [0:0] = [5, 6] ]</td>
<td>[ t = [2, 3] ]</td>
</tr>
<tr>
<td><strong>pop</strong></td>
<td>[ t = s.pop() ]</td>
<td>[ s = [2] ]</td>
</tr>
<tr>
<td><strong>remove</strong></td>
<td>[ t = s.remove(5) ]</td>
<td>[ s = [2, 3] ]</td>
</tr>
<tr>
<td><strong>slice assignment</strong></td>
<td>[ t = s[0:2] = [1] ]</td>
<td>[ t = [1] ]</td>
</tr>
</tbody>
</table>
Lists in Lists in Lists in Environment Diagrams

```python
t = [1, 2, 3]
t[1:3] = [t]
t.extend(t)
```

9

t = [[1, 2], [3, 4]]
t[0].append(t[1:2])

Examples: Objects

```python
jack
>>> jack
Peon
>>> jack.work() 'Maam, I work'
```

Land Owners

Instance attributes are found before class attributes; class attributes are inherited

```python
class Worker:
    greeting = 'Sir'
def __init__(self):
    self.elf = Worker
def work(self):
    return self.greeting + ', I work'
def __repr__(self):
    return 'Bourgeoisie' + greeting
class Bourgeoisie(Worker):
    greeting = 'Peon'
def work(self):
    print(Worker.work(self))
    return 'I gather wealth'
jack = Worker()
john = Bourgeoisie()
jack.greeting = 'Maam'
```

Examples: Iterables & Iterators
Using Built-In Functions & Comprehensions

What are the indices of all elements in a list s that have the smallest absolute value?

[-4, -3, -2, 3, 2, 4] → [2, 4]  [1, 2, 3, 4, 5] → [8]

What’s the largest sum of two adjacent elements in a list s? (Assume len(s) > 1)

[-4, -3, -2, 3, 2, 4] → 6  [-4, 3, -2, -3, 2, -4] → 1

Create a dictionary mapping each digit d to the lists of elements in s that end with d.

{5, 8, 13, 21, 34, 55, 89} → {1: [21], 3: [13], 4: [34], 5: [5, 55], 8: [8], 9: [89]}

Does every element equal some other element in s?


Examples: Linked Lists

Linked List Exercises

Is a linked list s ordered from least to greatest?

1 ——— 2 ——— 3 ——— 4

Is a linked list s ordered from least to greatest by absolute value (or a key function)?

1 ——— 3 ——— 4

-4 ——— -3 ——— -2 ——— -1 ——— 3

Create a sorted Link containing all the elements of both sorted Links s & t.

1 ——— 5

1 ——— 6

1 ——— 1 ——— 6 ——— 5

Do the same thing, but never call Link.

1 ——— 5

1 ——— 6