

8 Lists and Dictionaries

Monday, March 11, 2024 1:58 PM

• LISTS

- Ordered sequence of elements
 - don't have to be of the same type
 - can be other sequences.

- Mutable
 - = not a copy
instead, an alias
 - passed by reference
to functions.

• list(x)

• list[i] used to both read and write.

→ methods

- append(x)
- extend(l)
- insert(i, x)
- remove(x)
- pop()
- pop(count)
- reverse()
- sort()
- index(x)
- count(x)

- Operators

[] in +

≠ functions.

max(l)

min(l)

sum(l)

sorted(l)

len(l)

all(l) :- True :- if all elements are non-zero

any(l) :- True :- if any element is non-zero.

[] False are considered "zero"

- Lists can be sliced

```
list[start:end]
```

start and end are optional
-start, -end are also allowed

```
list[start:end:stride]
```

- Iterating over Lists

```
for item in list:
```

→ block.

to avoid Index errors.

generator `enumerate(list)`

more in module `itertools`.

- Modifying a list in a loop. (tricky)

- `for item in list:`

`item` is a temporary variable.

changes to `item` may not reflect on actual list.

→ use positional values

```
for i in range(len(list)):
```

🚩 **Danger:** removing or adding elements in a loop

```
for item in list:
```

→ strange things happen if you modify list in the loop

- List comprehensions.

"fish-based" way of creating lists

List comprehensions.

"Turbocharged" way of creating lists

Syntax: `[expr for loop_var in iterable]`



```
l = []  
for loop_var in iterable:  
    l.append(expr)
```

Syntax 2:

`[expr for loop_var in iterable if condition]`



```
l = []  
for loop_var in iterable:  
    if condition:  
        l.append(expr)
```

List comprehensions can be nested !!! ~~is~~ easily abused.

e.g. Matrix initialization.

```
[[0 for j in range(3)] for i in range(5)]
```

• DICTIONARIES

Def: an unordered collection of $\langle \text{key}, \text{value} \rangle$ pairs.

- keys are unique
- designed to be accessed by key

• Literal: `{key1: val1, key2: val2, key3: val3, ...}`

Keys can only be "immutable" values

• Constructor `dict(x)`

• Operators `dict[key]` `del dict[key]` `k in dict`

- methods

- clear()
- get(key)
- get(key, default)

- pop(key, default)
- update(dict)

- Iterating over a dictionary
 - default by keys.
 - items()
 - values()

- Dictionaries can be nested.
Very common technique for data organization.

—•— EOF