Ch8 Lists and Dictionaries

Friday, November 3, 2023 8:04 AM

- LISTS

- Ordered sequence of elements.
- mutable = ≡ not a copy
  - passed by reference.

list(x) .. constructor

list[i] [read & write.

- methods
  - append(x)
  - extend(l)
  - insert(i, x)
  - remove(X)
  - pop()
  - pop(i)
  - reverse()
  - sort()
  - index(X)
  - count(X)
- Operators:
  - [] in
  - del +
- functions
  - sorted(l)
  - max(l)
  - min(l)
  - sum(l)
  - all(l) : TRUE if all elements are "non-zero"
  - any(l) : TRUE if any element is "non-zero"
  - len(l)

- Lists can be nested.

- Lists can be sliced.

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

  - end
  - start

  : position from the end of the list.

- Iterating over Lists.

```
for item in iterable:
    block.
```

↑ to avoid IndexError

  - generators / iterators.

```
range(n)
enumerate(l)
```

More in itertools. like zip(l1, l2)

- Modifying a List in a Loop.

```
for e in l:
```

e is a temporary variable.
changes to e may not reflect on actual list.

  - Use positional values

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

- Danger: removing or adding elements in a loop

```
for e in l:
```

↑

```
for e in l:
```

↑ strange things can happen if you modify l

- Remedy: create a new list.

- List comprehensions.

  turbocharged way of creating lists

  [expr for loop_var in iterable]

  ⇓

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

- DICTIONARIES

  Def: a collection of (key, value) pairs.
  - specially design to search for keys.
  - not ordered.
  - keys have to be unique.

  - Literal { key1 : value1 , key2 : value2 , ... }
  - Constructor dict(x)
  - Operators dict[k] del dict[k] k in dict.
  - methods.
    - clear()

- Methods.
  - clear()
  - get(key, default)
  - pop(key, default)
  - update(dict2)

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

- Dictionary nesting
  Commonly used to structure data