NPTEL MOOC

PROGRAMMING, DATA STRUCTURES AND ALGORITHMS IN PYTHON

Week 5, Lecture 6

Madhavan Mukund, Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

Doing nothing

- Recall: reading a number from the keyboard

```
while(True):
    try:
        userdata = input("Enter a number: ")
        usernum = int(userdata)
    except ValueError:
        print("Not a number. Try again")
    else:
        break
```

Doing nothing

- What if we just want to repeat the loop on an error?

```
while(True):
  try:
    userdata = input("Enter a number: ")
    usernum = int(userdata)
  except ValueError:
    # Do nothing
  else:
    break
```

Doing nothing

- Blocks such as `except:`, `else:`, …cannot be empty
- Use `pass` for a null statement

```
while(True):
    try:
        userdata = input("Enter a number: ")
        usernum = int(userdata)
    except ValueError:
        pass
    else:
        break
```

Removing a list entry

- Want to remove `l[4]`?

  ```
  del(l[4])
  ```

- Automatically contracts the list and shifts elements in `l[5:]` left

- Also works for dictionaries

- `del(d[k])` removes the key `k` and its associated value

Undefining a value

- In general, `del(x)` removes the value associated with `x`, makes `x` undefined

```
x = 7
del(x)
y = x+5

NameError: name 'x' is not defined
```

Checking undefined name

- Assign a value to x only if x is undefined

```
try:
   x
except NameError:
   x = 5
```

The value None

- None is a special value used to denote “nothing”
- Use it to initialise a name and later check if it has been assigned a valid value

```
x = None
…

if x is not None:
   y = x
```

- Exactly one value None
- `x is None` is same as `x == None`

Summary

- Use `pass` for an empty block
- Use `del()` to remove elements from a list or dictionary
- Use the special value `None` to check if a name has been assigned a valid value