Objects

- Objects represent information.
- They consist of data and behavior, bundled together to create abstractions.
- Objects can represent things, but also properties, interactions, and processes.
- A type of object is called a class; classes are first-class values in Python.
- Object-oriented programming:
  - A metaphor for organizing large programs.
  - Special syntax that can improve the composition of programs.
- In Python, every value is an object.
  - All objects have attributes.
  - A lot of data manipulation happens through object methods.
  - Functions do one thing; objects do many related things.

Example: Strings

Representing Strings: the ASCII Standard

- American Standard Code for Information Interchange
- 109,000 characters
- 93 scripts (organized)
- Enumeration of character properties, such as case
- Supports bidirectional display order
- A canonical name for every character

- "Bell" (a) "Line feed" (ln)
- Layout was chosen to support sorting by character code
- Rows indexed 2-5 are a useful 6-bit (64 element) subset
- Control characters were designed for transmission

Mutation Operations
Some Objects Can Change

Tuples are Immutable Sequences

Mutation Can Happen Within a Function Call

Tuples are Immutable Sequences

Identity Operators

Identity

Equality

Identical objects are always equal values

(Deemo)
Mutable Default Arguments are Dangerous

A default argument value is part of a function value, not generated by a call.

```python
>>> def f(s=[]):
...     s.append(3)
...     return len(s)
... >>> f()
1
>>> f() 
2
>>> f()
3
```

Each time the function is called, `s` is bound to the same value!