

Objects

Announcements

Class Statements

(Demo)

Object-Oriented Programming

A method for organizing programs

- Data abstraction
- Bundling together information and related behavior

A metaphor for computation using distributed state

- Each object has its own local state
- Each object also knows how to manage its own local state, based on method calls
- Method calls are messages passed between objects
- Several objects may all be instances of a common type
- Different types may relate to each other

Specialized syntax & vocabulary to support this metaphor

A **class** defines how objects of a particular type behave

An **object** is an instance of a class; the class is its type

A **method** is a function called on an object using a dot expression: `l.append(5)`

Object Construction

Idea: All rationals should have a numerator and a denominator; the **Rational** class should add those attributes to each of its instances

```
>>> r = Rational(2, 3)
>>> r.numer
2
>>> r.denom
3
```

When a class is called:

1. A new instance of that class is created:
2. The `__init__` method of the class is called with the new object as its first argument (named `self`), along with any additional arguments provided in the call expression

An account instance

numer: 2 denom: 3

```
class Rational:
    def __init__(self, n, d):
        ▶ self.numer = n
        ▶ self.denom = d
```

`__init__` is called a constructor

More Tree Practice

(Demo)