61A Lecture 16

Announcements

String Representations

String Representations

An object value should behave like the kind of data it is meant to represent

For instance, by producing a string representation of itself Strings are important: they represent language and programs

In Python, all objects produce two string representations: $^\circ \mbox{The str}$ is legible to humans

•The **repr** is legible to the Python interpreter

The str and $\operatorname{\mathbf{repr}}$ strings are often the same, but not always

The repr String for an Object

The repr function returns a Python expression (a string) that evaluates to an equal object

repr(object) -> string Return the canonical string representation of the object. For most object types, eval(repr(object)) == object.

The result of calling repr on a value is what Python prints in an interactive session

>>> 12e12
1200000000000.0
>>> print(repr(12e12))
1200000000000.0

Some objects do not have a simple Python-readable string >>> repr(min)
'<built-in function min>'

The str String for an Object

Human interpretable strings are useful as well:

>>> from fractions import Fraction
>>> half = Fraction(1, 2)
>>> repr(half)
'Fraction(1, 2)'
>>> str(half)
'1/2'

The result of calling str on the value of an expression is what $\ensuremath{\mathsf{Python}}$ prints using the print function:

>>> print(half)
1/2

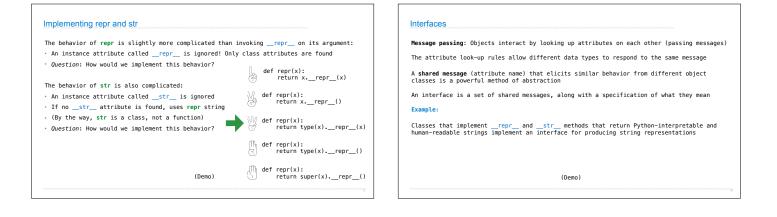
(Demo)

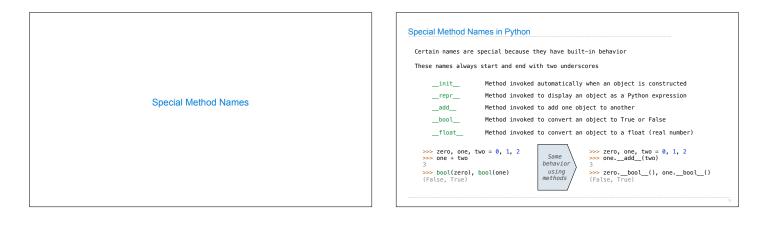


Polymorphic function: A function that applies to many (poly) different forms (morph) of data

repr invokes a zero-argument method __repr__ on its argument

str invokes a zero-argument method __str__ on its argument





A polymorphic function might take two or more arguments of different types
Type Dispatching: Inspect the type of an argument in order to select behavior
Type Coercion: Convert one value to match the type of another
>>> Ratio(1, 3) + 1 Ratio(4, 3)
>>> 1 + Ratio(1, 3) Ratio(4, 3)
>>> from math import pi
>>> Ratio(1, 3) + pi 3,4749259869231266
(Demo)