Special Methods

First Lesson. (Most Important)

Special Methods

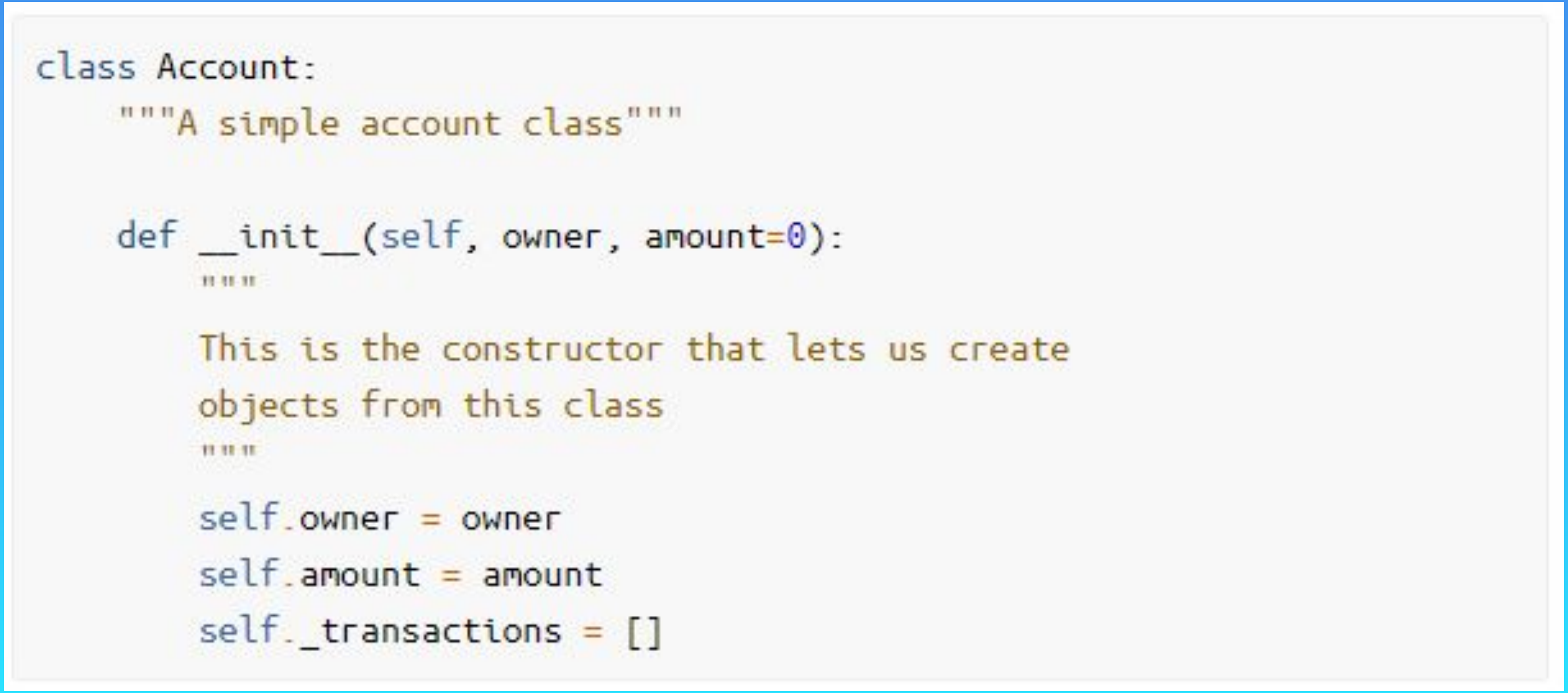
- __len__
- __init__
- __repr__
- __str__
- __len__,__getitem__,__reversed
- __eq__,__lt__
- __add__
- __call__

__len__

```
class NoLenSupport:
    pass

>>> obj = NoLenSupport()
>>> len(obj)
TypeError: "object of type 'NoLenSupport' has no len()"
```

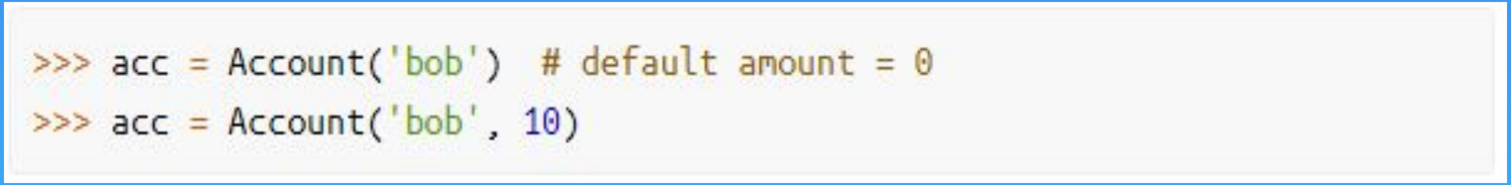
```
class LenSupport:
    def __len__(self):
        return 42

>>> obj = LenSupport()
>>> len(obj)
42
```

__init__



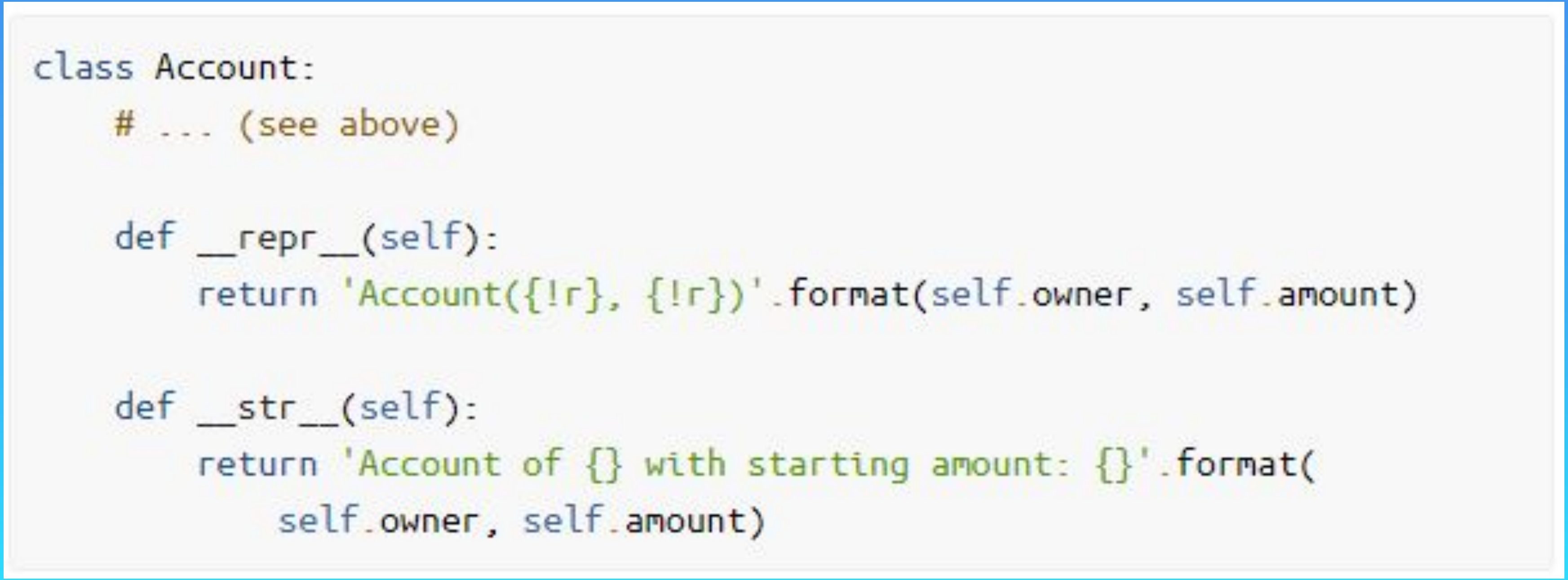
```python
class Account:
    """A simple account class"""

    def __init__(self, owner, amount=0):
        """
        This is the constructor that lets us create
        objects from this class
        """
        self.owner = owner
        self.amount = amount
        self._transactions = []
```

__init__



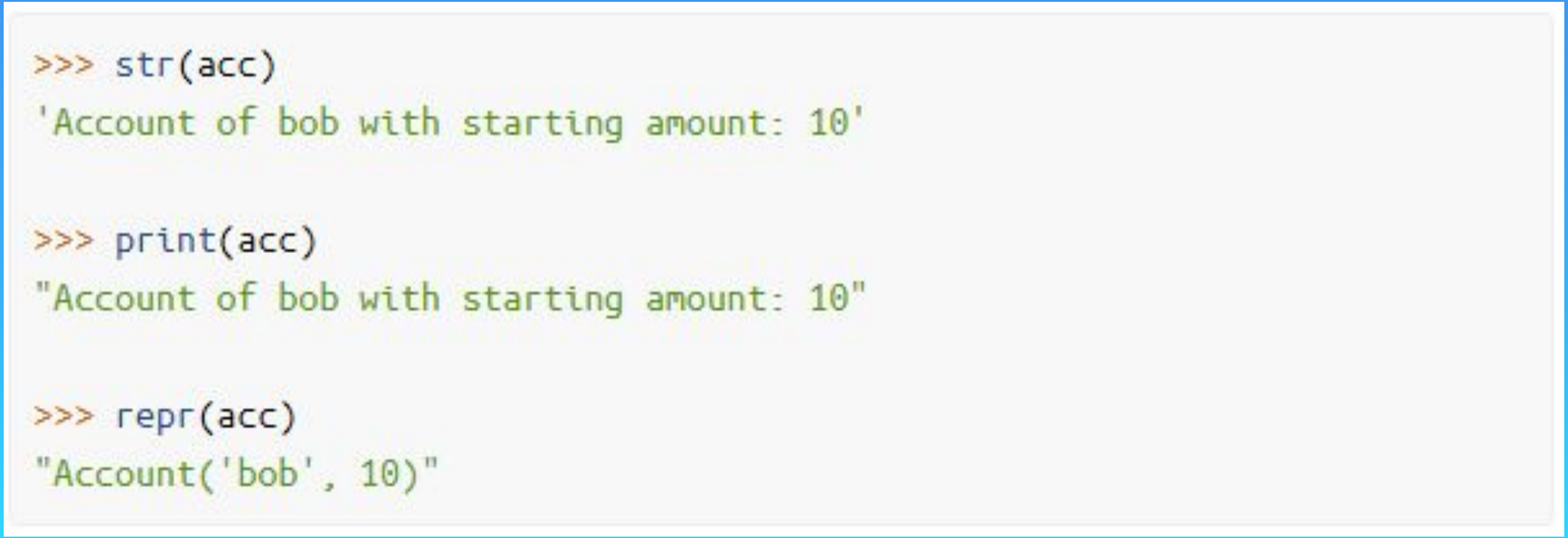
```
>>> acc = Account('bob')  # default amount = 0
>>> acc = Account('bob', 10)
```

__repr__,__str__



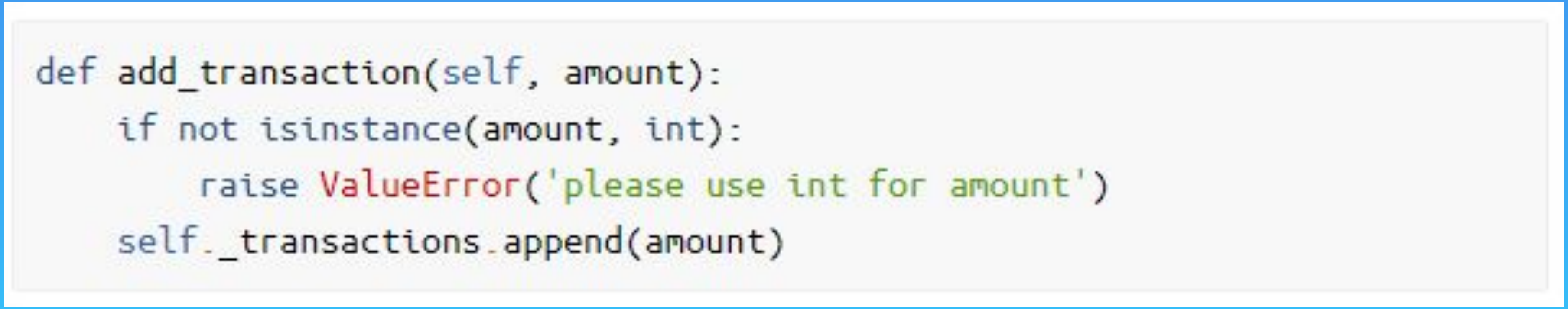
```python
class Account:
    # ... (see above)

    def __repr__(self):
        return 'Account({!r}, {!r})'.format(self.owner, self.amount)

    def __str__(self):
        return 'Account of {} with starting amount: {}'.format(
            self.owner, self.amount)
```

__repr__,__str__

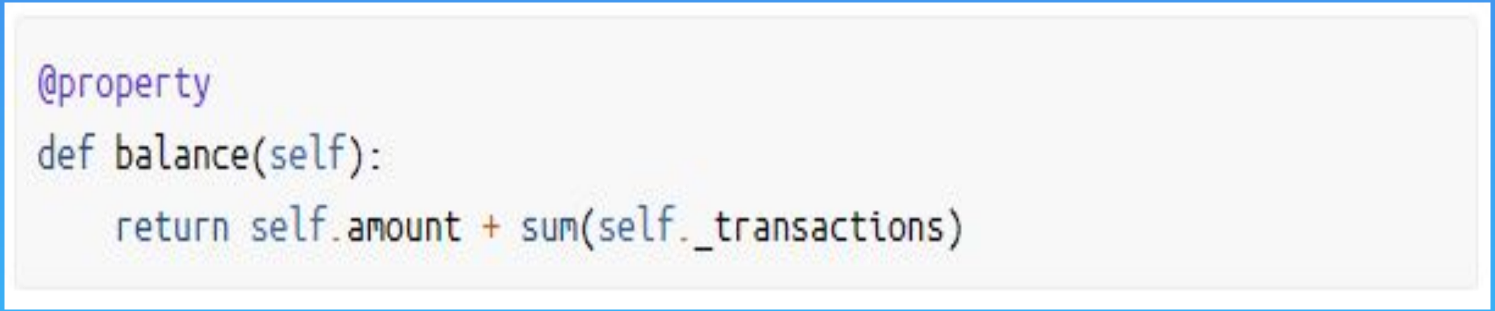


```
>>> str(acc)
'Account of bob with starting amount: 10'

>>> print(acc)
"Account of bob with starting amount: 10"

>>> repr(acc)
"Account('bob', 10)"
```

__len__,__getitem__

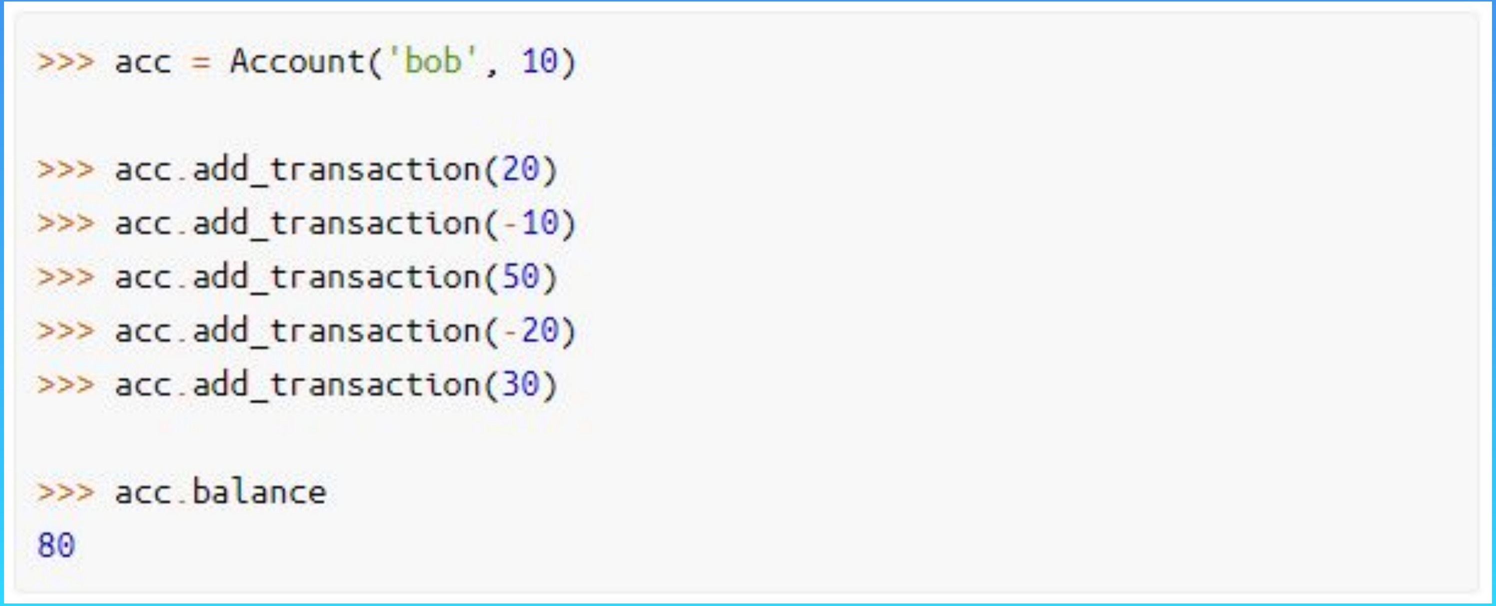

```
def add_transaction(self, amount):
    if not isinstance(amount, int):
        raise ValueError('please use int for amount')
    self._transactions.append(amount)
```

__len__,__getitem__



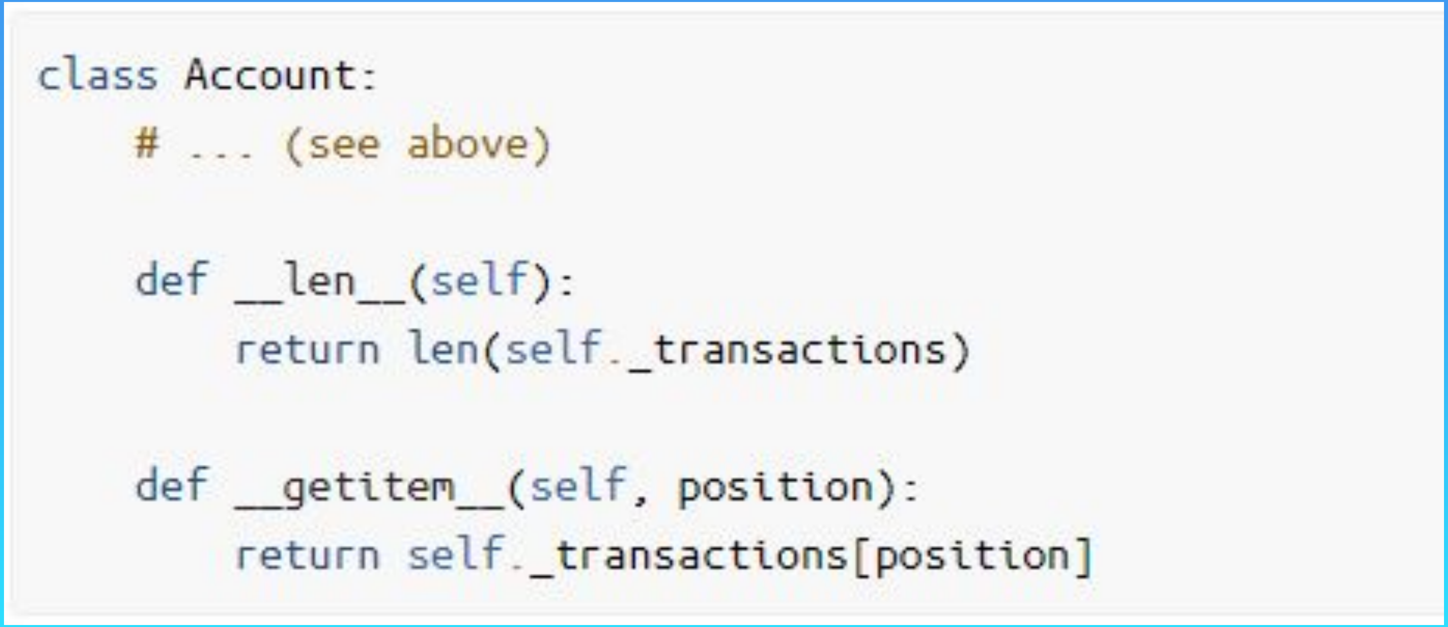
```
@property
def balance(self):
    return self.amount + sum(self._transactions)
```

__len__,__getitem__



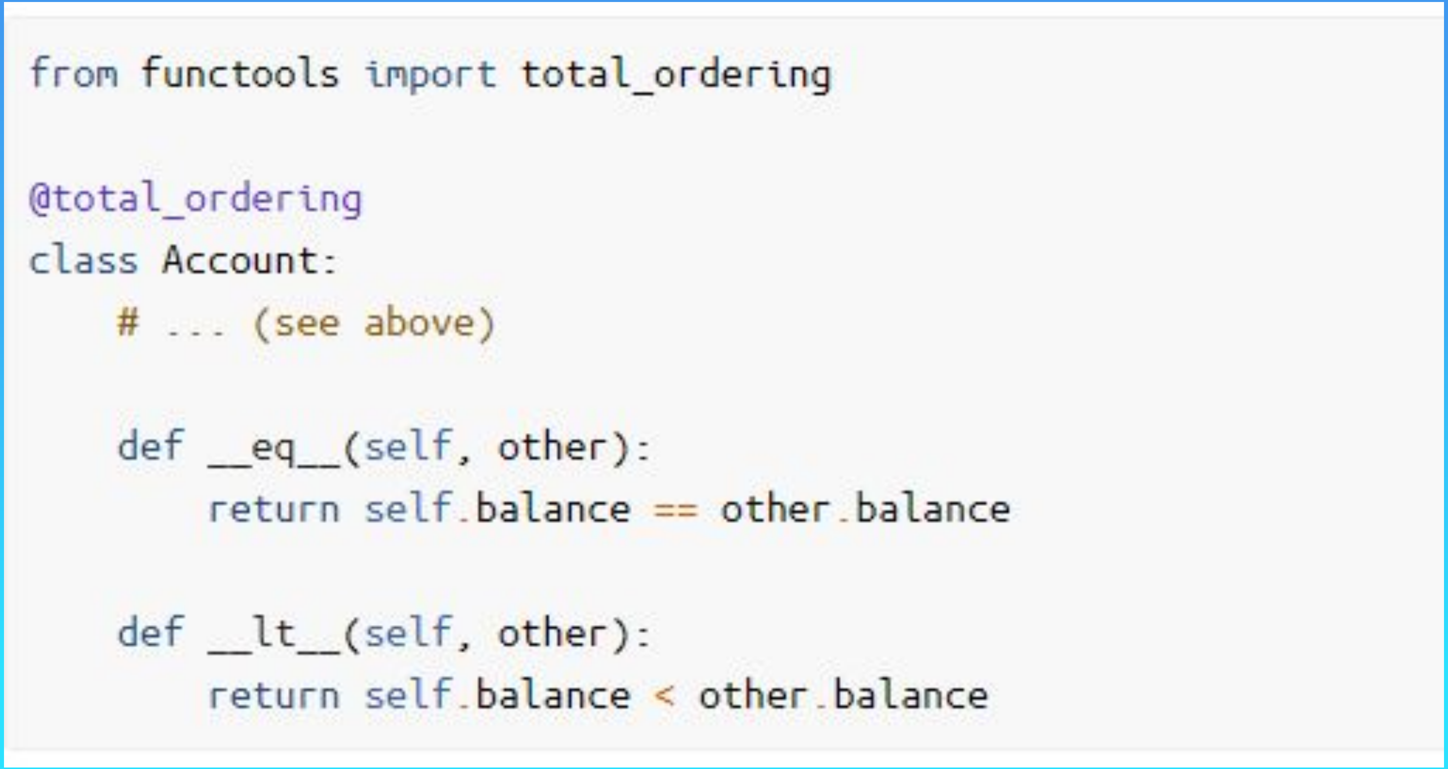
```
>>> acc = Account('bob', 10)

>>> acc.add_transaction(20)
>>> acc.add_transaction(-10)
>>> acc.add_transaction(50)
>>> acc.add_transaction(-20)
>>> acc.add_transaction(30)

>>> acc.balance
80
```

__len__, __getitem__

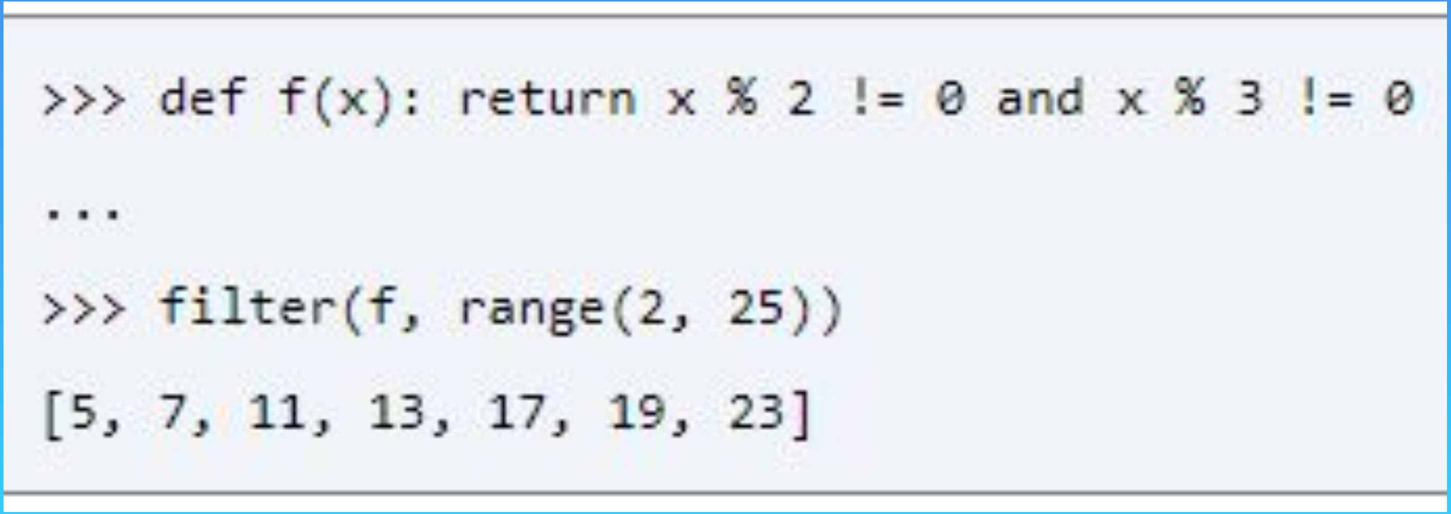


```python
class Account:
    # ... (see above)

    def __len__(self):
        return len(self._transactions)

    def __getitem__(self, position):
        return self._transactions[position]
```

Lessthan & Equalto

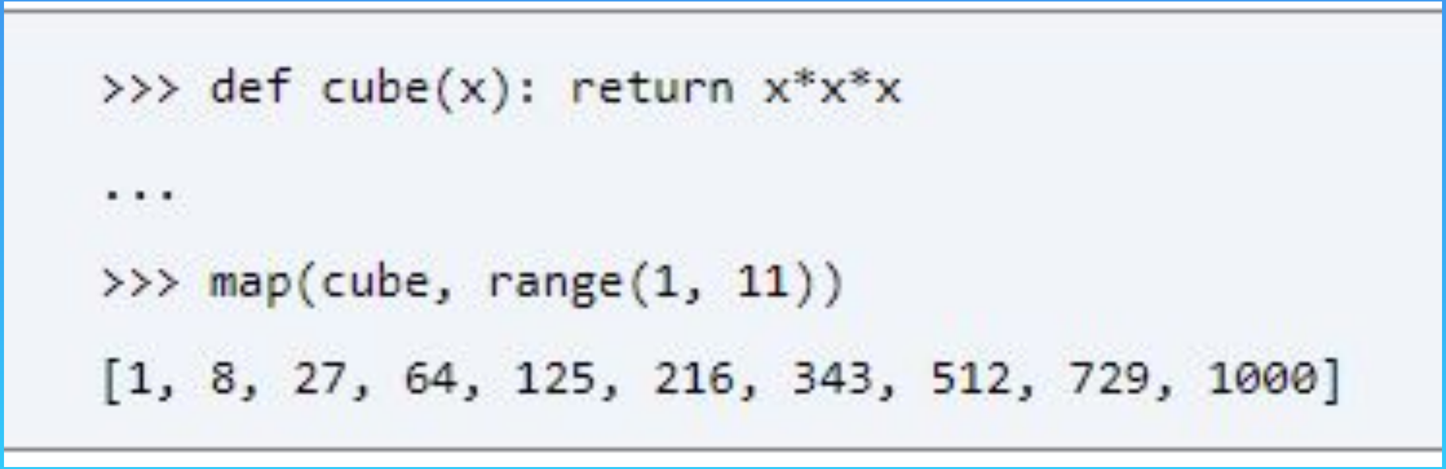

```python
from functools import total_ordering

@total_ordering
class Account:
    # ... (see above)

    def __eq__(self, other):
        return self.balance == other.balance

    def __lt__(self, other):
        return self.balance < other.balance
```

Functional Programming - Filter



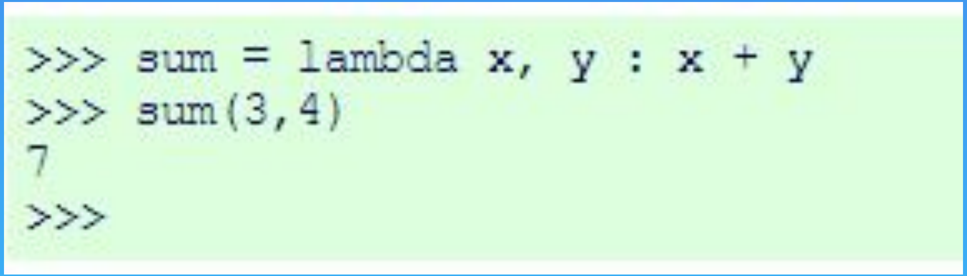
```
>>> def f(x): return x % 2 != 0 and x % 3 != 0
...
>>> filter(f, range(2, 25))
[5, 7, 11, 13, 17, 19, 23]
```

Functional Programming - Map

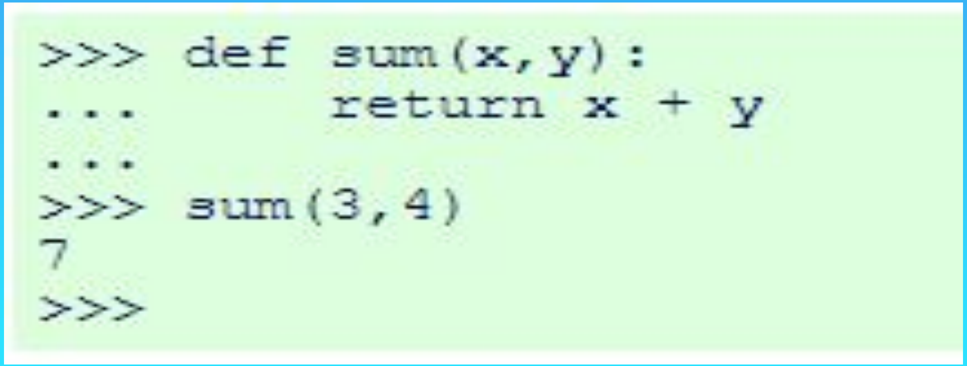


```
>>> def cube(x): return x*x*x

...

>>> map(cube, range(1, 11))

[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
```

Lambda Expressions



```
>>> sum = lambda x, y : x + y
>>> sum(3,4)
7
>>>
```



```
>>> def sum(x,y):
...         return x + y
...
>>> sum(3,4)
7
>>>
```

Special Method to Add 2 classes.

Assignment

Special Method to Compare 2 classes.

Assignment

Web Scrapping

Second Lesson

Using urllib in Python

Since HTTP is so common, we have a library that does all the socket work for us and makes web pages look like a file

```
import urllib.request, urllib.parse, urllib.error

fhand =
urllib.request.urlopen('http://data.lessthan60.in/romeo.txt')
for line in fhand:
    print(line.decode().strip())
```

urllib1.py

What is Web Scraping?

- When a program or script pretends to be a browser and retrieves web pages, looks at those web pages, extracts information, and then looks at more web pages

- Search engines scrape web pages - we call this “spidering the web” or “web crawling”

http://en.wikipedia.org/wiki/Web_scraping
http://en.wikipedia.org/wiki/Web_crawler

Why Scrape?

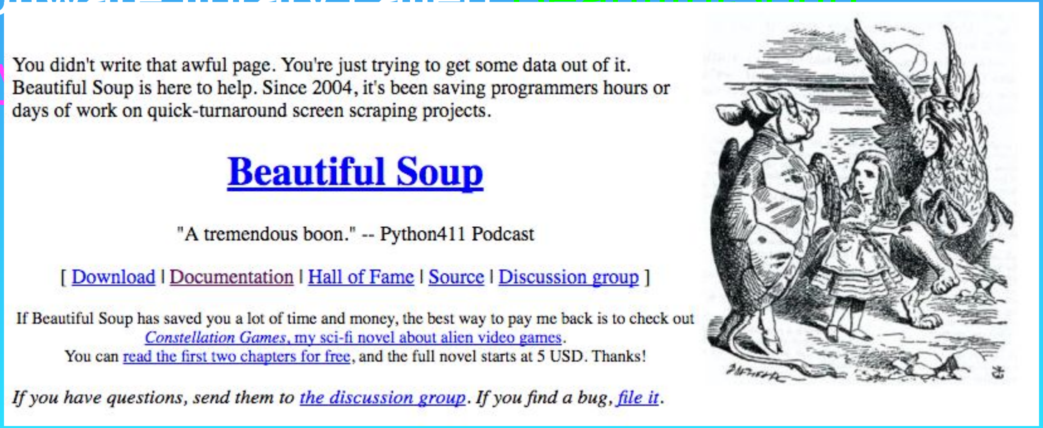
- Pull data - particularly social data - who links to who?
- Get your own data back out of some system that has no “export capability”
- Monitor a site for new information
- Spider the web to make a database for a search engine

Scraping Web Pages

- There is some controversy about web page scraping and some sites are a bit snippy about it.
- Republishing copyrighted information is not allowed
- Violating terms of service is not allowed

The Easy Way - Beautiful Soup

- You could do string searches the hard way
- Or use the free software library called BeautifulSoup from www.crummy


You didn't write that awful page. You're just trying to get some data out of it. Beautiful Soup is here to help. Since 2004, it's been saving programmers hours or days of work on quick-turnaround screen scraping projects.

Beautiful Soup

"A tremendous boon." -- Python411 Podcast

[Download | Documentation | Hall of Fame | Source | Discussion group]

If Beautiful Soup has saved you a lot of time and money, the best way to pay me back is to check out Constellation Games, my sci-fi novel about alien video games.
You can read the first two chapters for free, and the full novel starts at 5 USD. Thanks!

If you have questions, send them to the discussion group. If you find a bug, file it.

https://www.crummy.com/software/BeautifulSoup/

BeautifulSoup Installation

```
# To run this, you can install BeautifulSoup

# Or download the file
# and unzip it in the same directory as this file

import urllib.request, urllib.parse, urllib.error
from bs4 import BeautifulSoup

...
```

urllinks.py

BeautifulSoup Data

```
html_doc = """

<html><head><title>The Dormouse's story</title></head>

<body>

<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were

<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,

<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and

<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;

and they lived at the bottom of a well.</p>

<p class="story">...</p>
```

BeautifulSoup prettify

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
print(soup.prettify())
```

BeautifulSoup Parsing

```
soup.title

# <title>The Dormouse's story</title>

soup.title.name

# u'title'

soup.title.string

# u'The Dormouse's story'

soup.title.parent.name

# u'head'

soup.p

# <p class="title"><b>The Dormouse's story</b></p>
```

BeautifulSoup Parsing

```
soup.p['class']
# u'title'
soup.a
# <a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>
soup.find_all('a')
# [<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>,
#  <a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
#  <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
soup.find(id="link3")
# <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>
```

BeautifulSoup Parsing

```
for link in soup.find_all('a'):
    print(link.get('href'))
# http://example.com/elsie
# http://example.com/lacie
# http://example.com/tillie
```

```
import urllib.request, urllib.parse, urllib.error
from bs4 import BeautifulSoup

url = input('Enter - ')
html = urllib.request.urlopen(url).read()
soup = BeautifulSoup(html, 'html.parser')

# Retrieve all of the anchor tags
tags = soup('a')
for tag in tags:
    print(tag.get('href', None))
```

```
python urllinks.py
Enter - http://lessthan60.in/page1.htm
http:// http://lessthan60.in /page2.htm
```

Parse and extract profile from MaheshRakheja.com

Assignment

Parse and create money converter

Assignment