lecture_21

October 16, 2022

1 Lecture 21

[1]:
```
a = [1, 2, 3, 4]
```

[2]:
```
for i in a:
    print(i)
```

```
1
2
3
4
```

[3]:
```
for i in range(0, 10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

Iterables: `list`, `string`, `range()`, etc.

[4]:
```
class Foo:
    def __init__(self, name):
        self.name = name
```

[5]:
```
f = Foo("Adam")
```

[6]:
```
for i in f:
    print(i)
```

```
---------------------------------------------------------------------------
```

```
TypeError                                 Traceback (most recent call last)
Input In [6], in <cell line: 1>()
----> 1 for i in f:
      2     print(i)

TypeError: 'Foo' object is not iterable
```

[7]:
```
class Library:
    def __init__(self, name, books):
        self.name = name
        self.books = books
```

[8]:
```
library_1 = Library("Open Library", ["Accelerate", "Clean Code", "Algorithms"])
```

[9]:
```
library_1.books
```

[9]:
```
['Accelerate', 'Clean Code', 'Algorithms']
```

[10]:
```
for i in library_1:
    print(i)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [10], in <cell line: 1>()
----> 1 for i in library_1:
      2     print(i)

TypeError: 'Library' object is not iterable
```

[11]:
```
for i in library_1.books:
    print(i)
```

```
Accelerate
Clean Code
Algorithms
```

[15]:
```
class Library:
    def __init__(self, name, books):
        self.name = name
        self.books = books

    def __iter__(self):
        return iter(self.books)
```

[16]:
```
library_2 = Library("Open Library #2", ["Accelerate", "Clean Code",␣
↪"Algorithms"])
```

[17]:
```
for i in library_2:
    print(i)
```

```
Accelerate
Clean Code
Algorithms
```

[23]:
```
class Library:
    def __init__(self, name, books):
        self.name = name
        self.books = books
        self.__idx = 0

    def __iter__(self):
        return self

    def __next__(self):
        if self.__idx >= len(self.books):
            raise StopIteration()
        self.__idx += 1
        return self.__idx, self.books[self.__idx-1]
```

[24]:
```
library_3 = Library("Open Library #3", ["Accelerate", "Clean Code",␣
 ↪"Algorithms"])
```

[25]:
```
for i in library_3:
    print(i)
```

```
(1, 'Accelerate')
(2, 'Clean Code')
(3, 'Algorithms')
```

[27]:
```
class LibraryIterator:
    def __init__(self, books):
        self.books = books
        self.__idx = 0

    def __next__(self):
        if self.__idx >= len(self.books):
            raise StopIteration()
        self.__idx += 1
        return self.__idx, self.books[self.__idx-1]
```

[28]:
```
a = LibraryIterator(["Accelerate", "Clean Code", "Algorithms"])
```

[29]:
```
for i in a:
    print(i)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [29], in <cell line: 1>()
----> 1 for i in a:
      2     print(i)

TypeError: 'LibraryIterator' object is not iterable
```

[30]:
```
class Library:
    def __init__(self, name, books):
        self.name = name
        self.books = books

    def __iter__(self):
        return LibraryIterator(self.books)
```

[31]:
```
library_4 = Library("Open Library #4", ["Accelerate", "Clean Code",␣
 ↪"Algorithms"])
```

[32]:
```
for i in library_4:
    print(i)
```

```
(1, 'Accelerate')
(2, 'Clean Code')
(3, 'Algorithms')
```

[33]:
```
library_4.books.append("The Pragmatic Programmer")
```

[34]:
```
library_4.books
```

[34]:
```
['Accelerate', 'Clean Code', 'Algorithms', 'The Pragmatic Programmer']
```

[35]:
```
for i in library_4:
    print(i)
```

```
(1, 'Accelerate')
(2, 'Clean Code')
(3, 'Algorithms')
(4, 'The Pragmatic Programmer')
```

[36]:
```
next(library_4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [36], in <cell line: 1>()
----> 1 next(library_4)
```

```
TypeError: 'Library' object is not an iterator
```

[37]:
```
a = iter(library_4)
```

[38]:
```
next(a)
```

[38]:
```
(1, 'Accelerate')
```

[39]:
```
next(a)
```

[39]:
```
(2, 'Clean Code')
```

[40]:
```
next(a)
```

[40]:
```
(3, 'Algorithms')
```

[42]:
```
next(a)
```

[42]:
```
(4, 'The Pragmatic Programmer')
```

[43]:
```
t = "test string"
```

[44]:
```
next(t)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [44], in <cell line: 1>()
----> 1 next(t)

TypeError: 'str' object is not an iterator
```

[45]:
```
ti = iter(t)
```

[46]:
```
next(ti)
```

[46]:
```
't'
```

[47]:
```
next(ti)
```

[47]:
```
'e'
```

[48]:
```
next(ti)
```

[48]:
```
's'
```

[49]:
```
library_4[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Input In [49], in <cell line: 1>()
----> 1 library_4[0]

TypeError: 'Library' object is not subscriptable
```

[50]:
```
d = {
    "a": 1,
    "b": 2
}
```

[51]:
```
for i in d:
    print(i)
```

```
a
b
```

[52]:
```
for k, v in d.items():
    print(k, v)
```

```
a 1
b 2
```

[53]:
```
for v in d.values():
    print(v)
```

```
1
2
```

[56]:
```
import itertools
```

[58]:
```
for i in itertools.count(42):
    print(i)
    if i == 52:
        break
```

```
42
43
44
45
46
47
48
49
```

```
50
51
52
```

[60]:
```
for i in itertools.count(42, 3):
    print(i)
    if i >= 52:
        break
```

```
42
45
48
51
54
```

[61]:
```
type(itertools.count(42, 3))
```

[61]: itertools.count

[62]:
```
a = iter(itertools.count(42, 3))
```

[63]:
```
type(a)
```

[63]: itertools.count

[64]:
```
a = [12, 24, 42]
```

[69]:
```
for idx, elem in enumerate(itertools.cycle(a)):
    print(idx, elem)
    if idx == 10:
        break
```

```
0 12
1 24
2 42
3 12
4 24
5 42
6 12
7 24
8 42
9 12
10 24
```

[74]:
```
y = 3
list(map(lambda x: x**y, range(10)))
```

[74]: [0, 1, 8, 27, 64, 125, 216, 343, 512, 729]

[75]:
```
a
```

[75]:
```
[12, 24, 42]
```

[76]:
```
t
```

[76]:
```
'test string'
```

[78]:
```
for i in itertools.chain(a, t):
    print(i)
```

```
12
24
42
t
e
s
t

s
t
r
i
n
g
```

[79]:
```
c = [a, t]
```

[80]:
```
c
```

[80]:
```
[[12, 24, 42], 'test string']
```

[81]:
```
for i in itertools.chain.from_iterable(c):
    print(i)
```

```
12
24
42
t
e
s
t

s
t
r
i
```

```
n
g
```

[82]:
```
a
```

[82]:
```
[12, 24, 42]
```

[83]:
```
for i in zip(a, itertools.cycle('xy')):
    print(i)
```

```
(12, 'x')
(24, 'y')
(42, 'x')
```

[ ]:
```

```