Python: collections

Сегодня

- Строки
- Списки
- Словари
- Кортежи
- Множества

Строки и кодировки

- в байте восемь битов
- ASCII — 128 символов
- Unicode 6.0 — 109384 символов (!)

Строки в Python 2

str ('string'):

- представляет собой последовательность байтов
- требует меньше памяти для хранения

unicode (u'строка'):

- представляет собой текст
- подходит для локализации

Строки в Python 3

str (’some data’) $\Rightarrow$ **bytes** (b’some data’)

unicode (u’строка’) $\Rightarrow$ **str** (’строка’)

Объявление строк

```
1  ordinary_string = 'hello'      # == "hello"
2  print(ordinary_string)
3
4  unicode_string  = u'hello'
5  print(unicode_string)
6
7  multiline_string = '''
8    This is multiline text.
9    It's stored in variable as written.
10 '''
11 print(multiline_string)
```

String are immutables

```
1    a = 'string'
2    ref_a = a
3    print(a)
4    print(ref_a)
5
6    a = 'new ' + a
7    print(a)
8    print(ref_a)
```

String: методы

```
a = ' ...hello, my dear FrIeNdS!... '
print(a)

print(a.center(80, '_'))
print(a.count('e'))
print(a.endswith('!'))
print(a.find('my'))

print(a.isspace())

print(a.lower())
print(a.upper())
print(a.title())

print(a.strip(' .'))
print(a.replace(',', '!'))
print(a.split(' '))
```

String: пример

```
1 usr_input = input("Input some words in english: ")
2
3 words_upper = []
4 for word in usr_input.split(' '):
5     if word.isupper():
6         words_upper.append(word)
7
8 print('Number of uppercase words:', len(words_upper))
9 print('List of them:', words_upper)
```

Вопросы

- Как создать строку, содержащую кавычки?
- Как создать копию строки?
- Как вывести строку справа-налево?

List — универсальный контейнер

```
1  # basic
2  print([1, 2, 3])
3
4  # from map
5  def cube(x): return x*x*x
6  print(map(cube, range(1, 11)))
7
8  # from filter
9  def f(x): return x % 3 == 0 or x % 5 == 0
10 print(filter(f, range(2, 25)))
11
12 # list comprehension
13 print([x**2 for x in range(10)])
```

List: методы

```
1  lst = ['a', 1, ]
2  print(lst)
3
4  lst.append('a')
5  print(lst)
6
7  lst.extend([1, 2, 3])
8  print(lst)
9
10 lst.insert(3, 'I')
11 print(lst)
12
13 lst.pop()
14 print(lst)
15
16 print(lst.index(1))
17
18 lst.reverse()
19 print(lst)
20
```

Вопросы

- Как удалить элемент из списка?
- Как отсортировать список по своим правилам?
- Как создать копию списка?

Dict — хранилище типа ключ-значение

```
1  tel = {'jack': 4098, 'sape': 4139}
2  tel['guido'] = 4127
3
4  print(tel)
5  print(tel.keys())
6  print(tel.values())
7
8  # dict comprehension
9  print({x: x**2 for x in (2, 4, 6)})
10
```

Dict: обход

```
knights = {
    'gallahad': 'the pure',
    'robin':    'the brave',
}
```

```
for k, v in knights.items():
    print(k, v)

```

```
for k in knights():
    print(k, knights[k])

```

Dict: пример

```
1  infile = open('input.txt', 'r')
2  words = infile.read().split()
3  infile.close()
4
5  counts = {}
6  for w in words:
7      counts[w] = counts.get(w, 0) + 1
8
9  outfile = open('output.txt', 'w')
10 for w in sorted(counts, key=counts.get, reverse=True):
11     outfile.write("| {:20} | {} time(s) |\n".format(
12     w, counts[w]))
13 outfile.close()
```

Нет предела совершенству

```
1  from collections import Counter
2
3  with open('input.txt', 'r') as infile:
4      words = infile.read().split()
5
6  counts = Counter(words)
7
8  with open('output.txt', 'w') as outfile:
9      for w,n in counts.most_common():
10         outfile.write("| {:20} | {} time(s) |\n".format(
11                         w, n))
```

Tuple — неизменяемый список

```
1 a = (1, 2)
2 b = tuple([3, 4])
3 print(a)
4 print(b)
5
6 a, b = b, a
7 print(a)
8 print(b)
9
```

Вопросы

- Зачем нужен tuple?

Set — математичекое множество

- Хранит уникальные элементы
- Порядок не важен
- Определены операции над множествами

Set: операции

```
1  a = set('abracadabra')
2  b = set('alacazam')
3
4  print a
5  print b
6
7  print a - b              # letters in a but not in b
8  print a | b              # letters in either a or b
9
10 print a & b              # letters in both a and b
11 print a ^ b              # letters in a or b but not both
```

Вопросы

- Как убрать дубликаты из списка?

Спасибо за внимание!

- https://github.com/budnyjj/courses_python
- https://vk.com/budnyjj
- budnyjj@gmail.com