

Python Базовый

Модули Python


Python Базовый

Introduction



Бондаренко Кирилл

Senior Data scientist, CreatorIQ

 [profile.php?id=100011447245832](https://www.facebook.com/profile.php?id=100011447245832)

 [kirill-bond/](https://www.linkedin.com/in/kirill-bond/)

 [@bond.kirill.alexandrovich](https://www.telegram.me/bond.kirill.alexandrovich)



Python Базовый

Тема урока

Модули Python

Python Базовый

План урока

1. Что такое модули и какие из них будут рассмотрены в рамках данного урока
2. Разбор модулей и практические примеры
3. Как написать свой модуль
4. Решение задач по пройденным модулям

Python Базовый

Какие модули Python будут в уроке

- **collections** - модуль с расширенными структурами данных
- **math** - модуль готовых математических функций
- **random** - модуль для генерации случайных значений
- **itertools** - модуль с готовыми функциями комбинаторики
- **re** - модуль поддержки регулярных выражений
- **datetime** - модуль даты и времени

Чтобы использовать модуль в коде, нужно сделать импорт:

```
import module_name
```

Python Базовый

Модуль collections

```
1  import collections
2
3  counts = collections.Counter([1, 1, 2, 3, 3, 3])
4  print(counts)  # Counter({3: 3, 1: 2, 2: 1})
5
6  dict_of_lists = collections.defaultdict(list)
7  for value in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
8      if value % 2 == 0:
9          dict_of_lists['even'].append(value)
10     else:
11         dict_of_lists['odd'].append(value)
12 print(dict_of_lists)
13 # {'odd': [1, 3, 5, 7, 9], 'even': [2, 4, 6, 8, 10]}
14
15 d = collections.deque([1, 2, 3])
16 d.popleft()  # [2, 3]
17 d.append(4)  # [2, 3, 4]
18 d.appendleft(6)  # [6, 2, 3, 4]
19 d.pop()  # [6, 2, 3]
20
21
```

Python Базовый

Модуль math

```
1  import math
2
3  print(math.sqrt(16)) # 4
4  print(math.pi) # 3.141592653589793
5  print(math.e) # 2.718281828459045
6  print(math.ceil(2.1)) # 3
7  print(math.floor(2.1)) # 2
8  print(math.dist([1, 0, 0], [0, 1, 0])) # 1.4142
9  print(math.isnan(math.nan)) # True
10 print(math.exp(2)) # ~math.e ** 2 ~ 7.38905609893065
11 print(math.log(8, 2)) # 3.0 (if no base, it's natural log ie. base~2.7)
12
13
```

Python Базовый

Модуль random

```
1 import random
2
3 print(random.random()) # random value from 0 to 1
4 print(random.randint(0, 10)) # random value in giver arguments range
5 print(random.choice([1, 2, 3, 4, 5])) # random element from iterable object
6 random.seed(42) # random generator start number
7 seq = [1, 2, 3, 4, 6]
8 print(random.shuffle(seq)) # returns nothing, shuffles the sequence
9 print(seq)
10
11
```


Python Базовый

Модуль itertools

```
1  import itertools
2
3  combs = itertools.combinations([1, 2, 3, 4], 2)
4  print(list(combs)) # [(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)]
5  permutations = itertools.permutations([1, 2, 3, 4], 2)
6  print(list(permutations))
7  # [(1, 2), (1, 3), (1, 4), (2, 1), (2, 3), (2, 4), (3, 1), (3, 2), (3, 4), (4, 1), (4, 2), (4, 3)]
8  print(list(itertools.combinations_with_replacement([1, 2, 3], 2)))
9  # [(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)]
10 print(list(itertools.accumulate([1, 2, 3, 4, 5]))) # [1, 3, 6, 10, 15]
11 print(list(itertools.accumulate([1, 2, 3, 4, 5], lambda a, b: a - b))) # [1, -1, -4, -8, -13]
12 print(list(itertools.filterfalse(lambda x: x % 2, range(10))))
13 # [0, 2, 4, 6, 8]
14
15
```

Python Базовый

Модуль re

```
1 import re
2
3 text = "Hi @mr_alex ! My name is John and I love coffee. #coffeelover #2021"
4
5 print(re.sub(r"[.!?\\"-]", '', text))
6 print([element for element in re.finditer(r"#[A-Za-z0-9]*", text)])
7 print(re.split(r"[.!?\\"-]", text))
8
9
```

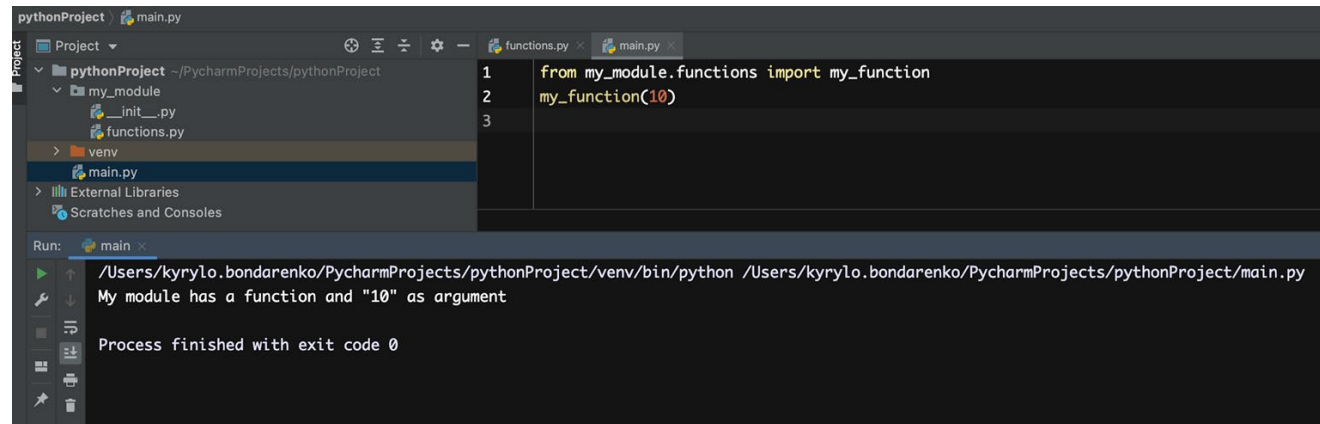
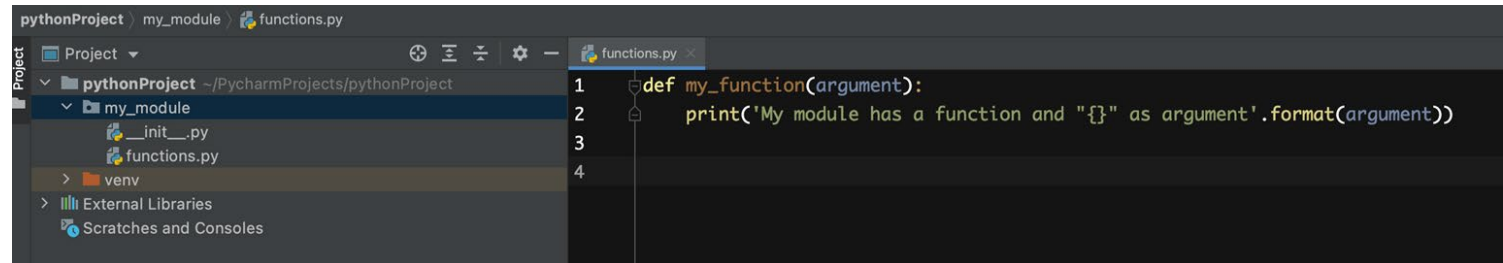
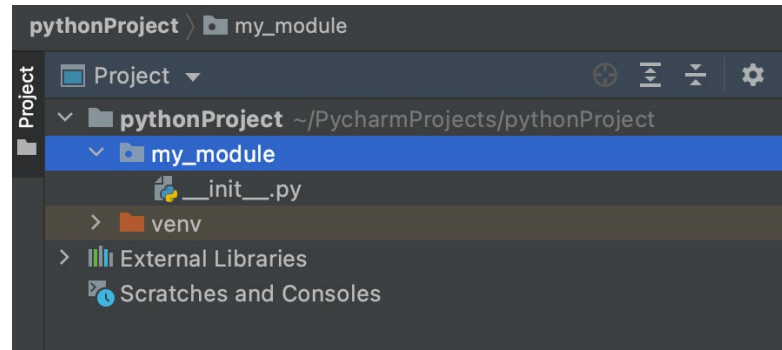
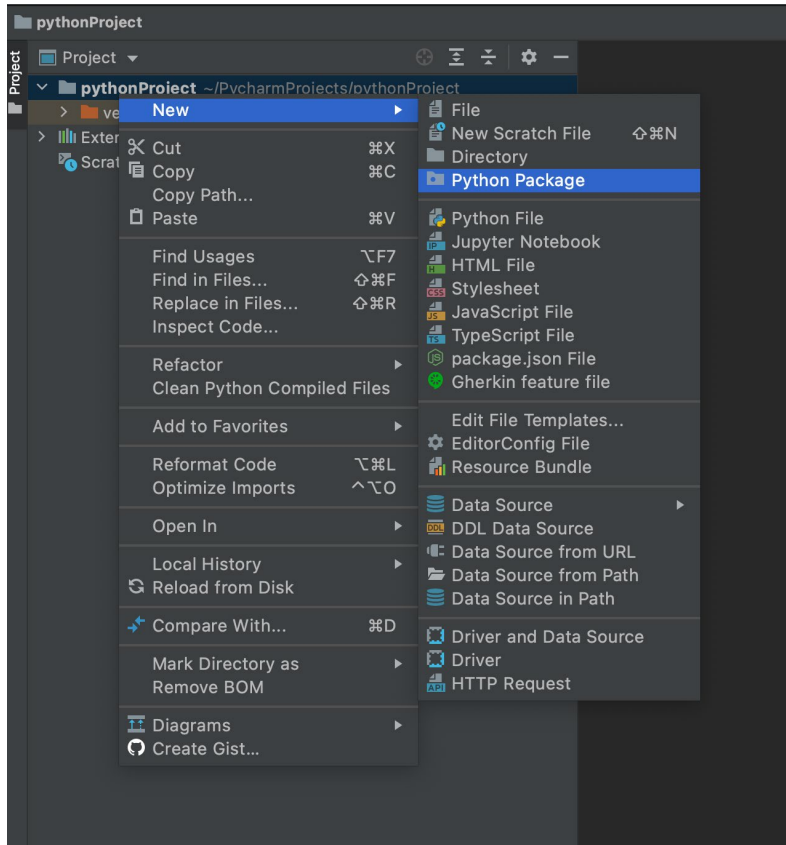
Python Базовый

Модуль datetime

```
1  from datetime import datetime
2
3  now = datetime.now()
4  print(now)
5  print(datetime.time(now))
6  print(now.year, now.month, now.day)
7
8  date_string = "21 June, 2018"
9  date_object = datetime.strptime(date_string, "%d %B, %Y")
10 print(date_object)
11
```

Python Базовый

Пишем свой модуль



Python Базовый

Задачи: часть 1

Есть строка: `"blue car is parked right behind the blue building with blue walls and red door"`

1. Написать функцию `process`, которая на вход принимает строку. Первым делом внутри функции найдите самое часто встречаемое слово (модуль `combinations`) и вынесите код в отдельный метод `get_max_count_word`.
2. Найдите в тексте все комбинации (пары по 2 слова) с самым часто встречаемым словом. Вынесите код в отдельный метод `get_max_count_word_combinations`, который вернет список кортежей.
3. Выберите случайную комбинацию из списка, полученного в пункте 2, и верните ее в методе `process`.

Python Базовый

Решение

```
1 import collections
2 import itertools
3 import random
4
5
6 def get_max_count_word(counts_dict):
7     max_count = 0
8     max_count_word = None
9     for word, count in counts_dict.items():
10         if count > max_count:
11             max_count = count
12             max_count_word = word
13     return max_count_word
14
15
16 def get_max_count_word_combinations(words, max_count_word):
17     combinations = itertools.combinations(words, 2)
18     max_count_word_combinations = [comb for comb in combinations if max_count_word in comb]
19     return max_count_word_combinations
20
21
22 def process(text):
23     words = text.split(' ')
24     counts = collections.Counter(words)
25     max_count_word = get_max_count_word(counts)
26     max_count_word_combinations = get_max_count_word_combinations(words, max_count_word)
27     random_combination = max_count_word_combinations[random.randint(0, len(max_count_word_combinations) - 1)]
28     return random_combination
29
30
31 process(text="blue car is parked right behind the blue building with blue walls and red door")
32
33
```

Python Базовый

Задачи: часть 2

Есть строка: `"I want to be a professor, but only one thing I can solve is $x=\exp(8)+\sqrt{17}-\log(5)$ "`

1. Объявите функцию `process`, которая будет принимать на вход строку. Первым делом найдите само математическое выражение в переданном тексте.
2. Сделайте отдельный словарь, в который запишите ключами имена всех операций как в тексте, а их значениями - соответствующие функции в пакете `math`, то есть `{ 'exp': math.exp, ... }`.
3. Разделите подстроку с выражением по знаку `"+"`. Итерируя каждый элемент в полученном списке выделите в нем отдельно название функции и отдельно значение. Из ранее сделанного словаря возьмите по полученному имени функцию и передайте в нее полученное значение. По очереди добавляйте полученные значения в новую переменную `result = 0`.
4. Сделайте переменную `"Result = {response}; Date {date}"`
5. Используя метод строки `format`, заполните значение `response = result` и значение `date =` текущей дате в формате день/месяц/год без времени. (Используйте `strftime` и формат `"%d/%m/%y"`)
6. Полученную строку выведите в консоль.

Python Базовый

Решение

```
1 import re
2 import math
3 import datetime
4
5 t = "I want to be a professor, but only one thing I can solve is  $x=\exp(8)+\sqrt{17}+\log(5)$ "
6
7
8 def process(text):
9     re_match = re.search('x=', text)
10    equation = text[re_match.span()[1]:]
11    ops = {'exp': math.exp, 'sqrt': math.sqrt, 'log': math.log}
12    actions = equation.split('+')
13    result = 0
14    for action in actions:
15        action = action.replace('(', ' ').replace(')', '')
16        op, value = action.split(' ')
17        result += ops[op](int(value))
18    report = "Result = {response}; Date {date}".format(
19        response=result,
20        date=datetime.datetime.now().strftime('%d/%m/%y')
21    )
22    print(report)
23
24
25 process(t)
26
27
```


Информационный видеосервис для разработчиков программного обеспечения



Проверка знаний

TestProvider.com



Проверьте как Вы усвоили данный материал на TestProvider.com

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

Python Базовый

Спасибо за внимание! До новых встреч!



Бондаренко Кирилл
Senior Data scientist, CreatorIQ

