

# Функции

In [1]:

```
# создаем функцию
def square(number):
    result = number ** 2
    return result

# print(square(10))
my_square = square(10)
print(my_square)
```

100

In [2]:

```
# help(print)
?print
```

In [3]:

```
# пишем docstring к своей функции
def square(number):
    """
    this is my function
    """
    result = number ** 2
    return result
```

In [4]:

```
?square
```

In [5]:

```
# функция без параметров
def square_2():
    user_input = int(input('Введите число'))
    result = user_input ** 2
    return result

square_2()
```

Введите число21

Out[5]:

441

In [6]:

```
# функция с двумя параметрами
def power(number, number_2):
    result = number ** number_2
    return result

power(4, 10)
```

Out[6]:

1048576

In [7]:

```
# функция с параметром по умолчанию
def power(number, number_2=2):
    result = number ** number_2
    return result

power(10)
```

Out[7]:

100

In [8]:

```
# если не указан return, функция всегда возвращает None!
def square_3(number):
    result = number ** 2
    print(result)
#     return None
#     return
print(square_3(5))
```

25

None

## Области видимости

In [9]:

```
number = 5
power = 3

def power_2():
    number = 6
    power = 2
    return number ** power

print(power_2())
print(number ** power)
```

36

125

In [10]:

```
number = 5
power = 3

def power_2():
    number = 6
    return number ** power

print(power_2())
print(number ** power)
```

216  
125

In [11]:

```
number = 5
power = 3

def power_2():
    return number ** power

power_2()
```

Out[11]:

125

In [12]:

```
def power_2():
    number = 6
    power = 2
    some_number = 1
    return number ** power

print(some_number)
```

```
-----
-
NameError                                Traceback (most recent call last)
t)
<ipython-input-12-461955134ac4> in <module>
      5     return number ** power
      6
----> 7 print(some_number)

NameError: name 'some_number' is not defined
```

In [13]:

```
# global

name = 'James'
def say_hi():
    global name
    name = 'Oleg'
    print('Hello', name)

say_hi()

print(name)
```

Hello Oleg  
Oleg

## lambda-функции

In [14]:

```
def func(x, y):
    return x + y
```

In [15]:

```
func = lambda x, y: x + y
```

In [16]:

```
func = lambda x, y: x + y
func(1, 6)
```

Out[16]:

7

In [17]:

```
nums = [1, 2, 3, 4, 5]
```

In [18]:

```
# even_numbers = lambda x: x % 2 == 0
odd_even_numbers = lambda x: 'even' if x % 2 == 0 else 'odd'
```

In [19]:

```
for num in nums:
    # print(even_numbers(num))
    print(odd_even_numbers(num))
```

odd  
even  
odd  
even  
odd

In [20]:

```
# функция map
list(map(odd_even_numbers, nums))
```

Out[20]:

```
['odd', 'even', 'odd', 'even', 'odd']
```

In [21]:

```
[odd_even_numbers(num) for num in nums]
```

Out[21]:

```
['odd', 'even', 'odd', 'even', 'odd']
```

In [22]:

```
def square(x):
    return x**2
```

In [23]:

```
for i in map(square, range(100, 105)):
    print(i)
```

```
10000
10201
10404
10609
10816
```

In [24]:

```
list(map(square, range(100,105)))
```

Out[24]:

```
[10000, 10201, 10404, 10609, 10816]
```

## Args and kwargs

In [25]:

```
api_request(1, 2, 3)
```

```
-----
-
NameError                                Traceback (most recent call las
t)
<ipython-input-25-285ff5011a16> in <module>
----> 1 api_request(1, 2, 3)

NameError: name 'api_request' is not defined
```

In [26]:

```
api_request(report=2, date=1, output=3)
```

```
-----
-
NameError                                Traceback (most recent call las
t)
```

```
<ipython-input-26-ebd349e3de86> in <module>
----> 1 api_request(report=2, date=1, output=3)
```

**NameError:** name 'api\_request' is not defined

In [27]:

```
def api_request(*params):
    date_start = params[0]
    date_end = params[1]
    print(params)
    print(date_start, date_end)
```

In [28]:

```
api_request('2019-01-01', '2019-01-31', ';sdfsdgsdg', 'sdgsdgsdg', 'sdgsdg')
```

```
('2019-01-01', '2019-01-31', ';sdfsdgsdg', 'sdgsdgsdg', 'sdgsdg')
2019-01-01 2019-01-31
```

In [29]:

```
def api_requets(**params):
    return params
```

In [30]:

```
api_requets(a=1, b=2, c=3, date='2020-01-01')
```

Out[30]:

```
{'a': 1, 'b': 2, 'c': 3, 'date': '2020-01-01'}
```

In [31]:

```
def api_request(**params):
    date_start = params['date_start']
    date_end = params['date_end']
    print(params)
    print(date_start, date_end)
```

In [32]:

```
api_request(date_start='2019-01-31', report='traffic', date_end='2019-01-01')
```

```
{'date_start': '2019-01-31', 'report': 'traffic', 'date_end': '2019-01-0
1'}
2019-01-31 2019-01-01
```

## Попрактикуемся

Хотим классифицировать жителей в файле по возрастам:

- до 18 лет - children
- 19-65 - young
- старше 65 - retiree

In [33]:

```
with open('adult.csv') as f:
    for line in f:
        print(line)
        break
```

```
-----
-
FileNotFoundError                                Traceback (most recent call last)
```

```
<ipython-input-33-a3bcc6641403> in <module>
```

```
----> 1 with open('adult.csv') as f:
      2     for line in f:
      3         print(line)
      4         break
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'adult.csv'
```

In [34]:

```
# подготовим файл для работы, разобьем строки по запятой и удалим пробельные символы
i = 0
with open('adult.csv', 'r') as f:
    for line in f:
        print(line.strip().split(','))

        i = i + 1
        if i > 5:
            break
```

```
-----
-
FileNotFoundError                                Traceback (most recent call last)
```

```
<ipython-input-34-0021c7f05987> in <module>
```

```
1 # подготовим файл для работы, разобьем строки по запятой и удалим
  пробельные символы
2 i = 0
----> 3 with open('adult.csv', 'r') as f:
      4     for line in f:
      5         print(line.strip().split(','))
```

```
FileNotFoundError: [Errno 2] No such file or directory: 'adult.csv'
```

In [35]:

```
# выделим первый "столбец" в отдельную переменную age
i = 0
with open('adult.csv', 'r') as f:
    for line in f:
        age, *other_columns = line.strip().split(',')

        if i > 0:
            print(age)
            print(other_columns[1])
        i += 1
    if i > 5:
        break
```

```
-----
-
FileNotFoundError                                Traceback (most recent call las
t)
<ipython-input-35-5378fd1b5eeb> in <module>
      1 # выделим первый "столбец" в отдельную переменную age
      2 i = 0
----> 3 with open('adult.csv', 'r') as f:
      4     for line in f:
      5         age, *other_columns = line.strip().split(',')

FileNotFoundError: [Errno 2] No such file or directory: 'adult.csv'
```

Добавим классификацию возрастов



In [36]:

```
i = 0
with open('adult.csv', 'r') as f:
    for line in f:
        age, *other_columns = line.strip().split(',')

        if i > 0:
            if int(age) <= 18:
                age_group = 'children'

            elif int(age) <= 60:
                age_group = 'young'

            else:
                age_group = 'retiree'

            print(age, age_group)

        i += 1
```

-----  
-  
**FileNotFoundError** Traceback (most recent call last):

<ipython-input-36-d06b61bf8227> in <module>

```
1 i = 0
----> 2 with open('adult.csv', 'r') as f:
3     for line in f:
4         age, *other_columns = line.strip().split(',')
5
```

**FileNotFoundError**: [Errno 2] No such file or directory: 'adult.csv'

Что тут нехорошо:

- сложно потестировать все случаи (в нашем цикле нужных случаев может и не оказаться)
- наш цикл стал довольно громоздким, а мы только начали
- эта классификация может потребоваться еще в 100500 местах кода
- данные могут быть кривыми и скрипт будет падать с ошибкой

Условия вычисления возрастной группы:

1. В строке 15 столбцов
2. Столбец с возрастом первый по счету
3. Возраст должен быть целым числом в адекватных пределах
4. Могут добавиться еще требования, о которых мы пока не знаем

In [37]:

```
def age_is_correct(age, lower_age=0, upper_age=120):  
    """  
    Проверка корректности возраста age по следующим правилам:  
    1. Целое число  
    2. В адекватных пределах  
  
    Возвращает True или False. Пример  
    age_is_correct(15)  
    True  
  
    age_is_correct(121)  
    False  
  
    age_is_correct(-5)  
    False  
    """  
  
    if str.isnumeric(age):  
        if lower_age <= int(age) <= upper_age:  
            return True  
  
    return False  
  
age_is_correct('5')
```

Out[37]:

True

In [38]:

```
age_is_correct('125aaa')
```

Out[38]:

False

In [39]:

```
def number_of_columns(line, separator=','):  
    """Возвращает количество столбцов в строке line с разделителем separator"""  
  
    return len(line.split(separator))
```

In [40]:

```
def line_is_correct(line):
    """
    Проверка строки на корректность. Проверяются условия:
    1. В строке 15 столбцов (пригодится с упражнения)
    2. Столбец с возрастом первый по счету
    3. Возраст должен быть целым числом в адекватных пределах
    """
    age = line.strip().split(',')[0]

    if number_of_columns(line) == 15:
        if age_is_correct(age):
            return True

    return False
```

In [41]:

```
# Добавляем функции в наш цикл
i = 0
with open('adult.csv', 'r') as f:
    for line in f:

        if line_is_correct(line):
            age, *other_columns = line.strip().split(',')
            if int(age) <= 18:
                age_group = 'children'

            elif int(age) <= 60:
                age_group = 'young'

            else:
                age_group = 'retiree'

            print(age, age_group)

        i += 1
    if i > 10:
        break
```

```
-----
-
FileNotFoundError                                Traceback (most recent call las
t)
<ipython-input-41-0217d19550e5> in <module>
      1 # Добавляем функции в наш цикл
      2 i = 0
----> 3 with open('adult.csv', 'r') as f:
      4     for line in f:
      5

FileNotFoundError: [Errno 2] No such file or directory: 'adult.csv'
```

In [42]:

```
# вынесем классификацию возраста в отдельную функцию
def age_classification(age):
    """
    Возвращает возрастную категорию для возраста age (можно передать как строку).
    Классификация категорий:
        - до 18 лет - children
        - 19-60 - young
        - старше 65 - retiree

    Пример
    age_classification('18')
    'children'

    age_classification(65)
    'retiree'
    """

    if int(age) <= 18:
        return 'children'
    elif int(age) <= 65:
        return 'young'
    elif int(age) > 65:
        return 'retiree'
    return 'boom'
```

In [43]:

```
age_classification(140)
```

Out[43]:

```
'retiree'
```

In [44]:

```
age_classification(18)
```

Out[44]:

```
'children'
```

In [45]:

```
?age_classification
```

In [46]:

```
# реализуем главную функцию
def main():
    with open('adult.csv', 'r') as f:
        for line in f:
            if line_is_correct(line):
                age, *columns = line.strip().split(',')
                print(age, age_classification(age))
```

In [47]:

```
main()
```

-----  
-  
**FileNotFoundError** Traceback (most recent call last)

<ipython-input-47-263240bbee7e> in <module>

----> 1 main()

<ipython-input-46-3e64a7b8ac07> in main()

1 # реализуем главную функцию

2 def main():

----> 3 with open('adult.csv', 'r') as f:

4 for line in f:

5 if line\_is\_correct(line):

**FileNotFoundError**: [Errno 2] No such file or directory: 'adult.csv'

In [ ]: