

Функции

In [1]: `from datetime import datetime`

```
def get_seconds():
    """Return current seconds"""
    return datetime.now().second
```

get_seconds()

Out[1]: 24

In [2]: `get_seconds.__doc__`

Out[2]: 'Return current seconds'

In [3]: `get_seconds.__name__`

Out[3]: 'get_seconds'

```
def split_tags(tag_string):
    tag_list = []
    for tag in tag_string.split(','):
        tag_list.append(tag.strip())

    return tag_list
```

split_tags('python, coursera, mooc')

Out[4]: ['python', 'coursera', 'mooc']

In [5]: `split_tags()`

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-5-866c00aba286> in <module>()
----> 1 split_tags()

TypeError: split_tags() missing 1 required positional argument: 'tag_string'
```

Аннотация типов

```
def add(x: int, y: int) -> int:
    return x + y
```

```
print(add(10, 11))
print(add('still ', 'works'))
```

21
still works

По ссылке или по значению?

```
def extender(source_list, extend_list):
    source_list.extend(extend_list)
```

```
values = [1, 2, 3]
extender(values, [4, 5, 6])

print(values)
```

[1, 2, 3, 4, 5, 6]

```
def replacer(source_tuple, replace_with):
    source_tuple = replace_with
```

```
user_info = ('Guido', '31/01')
replacer(user_info, ('Larry', '27/09'))

print(user_info)
```

('Guido', '31/01')

Именованные аргументы

```
def say(greeting, name):
    print('{} {}'.format(greeting, name))
```

```
say('Hello', 'Kitty')
say(name='Kitty', greeting='Hello')
```

Hello Kitty!
Hello Kitty!

Область видимости

In [10]: `result = 0`

```
def increment():
    result += 1
    return result

print(increment())
```

```
-----
UnboundLocalError                        Traceback (most recent call last)
<ipython-input-10-da69e363a112> in <module>()
      5     return result
      6
----> 7 print(increment())

<ipython-input-10-da69e363a112> in increment()
      2
      3 def increment():
----> 4     result += 1
      5     return result
      6

UnboundLocalError: local variable 'result' referenced before assignment
```

global & nonlocal

Аргументы по умолчанию

```
In [11]: def greeting(name='it's me...'):
         print('Hello, {}'.format(name))
```

greeting()

Hello, it's me...

```
In [12]: def append_one(iterable=[]):
         iterable.append(1)
```

```
         return iterable
```

```
print(append_one([1]))
```

[1, 1]

```
In [13]: print(append_one())
         print(append_one())
```

[1]
[1, 1]

In [14]: `print(append_one.__defaults__)`

((1, 1),)

```
In [15]: def function(iterable=None):
         if iterable is None:
             iterable = []
```

```
def function(iterable=None):
    iterable = iterable or []
```

Звездочки

```
In [16]: def printer(*args):
         print(type(args))

         for argument in args:
             print(argument)
```

printer(1, 2, 3, 4, 5)

```
<class 'tuple'>
1
2
3
4
5
```

In [17]: `name_list = ['John', 'Bill', 'Amy']`
`printer(*name_list)`

```
<class 'tuple'>
John
Bill
Amy
```

```
In [18]: def printer(**kwargs):
         print(type(kwargs))

         for key, value in kwargs.items():
             print('{}: {}'.format(key, value))
```

printer(a=10, b=11)

```
<class 'dict'>
a: 10
b: 11
```

```
In [19]: payload = {
         'user_id': 117,
         'feedback': {
             'subject': 'Registration fields',
             'message': 'There is no country for old men'
         }
     }

printer(**payload)
```

```
<class 'dict'>
user_id: 117
feedback: {'subject': 'Registration fields', 'message': 'There is no country for old men'}
```