

Функции

Примеры встроенных в Python функций: `print()`, `str()`, `int()`, `float()`, `len()`.

Создаём свою функцию

```
def hello(hour):
    if hour <= 5 or hour >= 23:
        print('Доброй ночи!')
    elif hour >= 6 and hour <= 11:
        print('Доброе утро!')
    elif hour >= 12 and hour <= 17:
        print('Добрый день!')
    elif hour >= 18 and hour <= 22:
        print('Добрый вечер!')
```

Аргументы функции

у name -- значение по умолчанию

```
def say_hello(hour, name=''):
    if hour <= 5 or hour >= 23:
        message = 'Доброй ночи'
    elif hour >= 6 and hour <= 11:
        message = 'Доброе утро'
    elif hour >= 12 and hour <= 17:
        message = 'Добрый день'
    elif hour >= 18 and hour <= 22:
        message = 'Добрый вечер'
    if name != '':
        print(message + ', ' + name + '!')
    else:
        print(message + '!')
```

Возврат значений из функции

```
# функция для вычисления
# периметра прямоугольника
# от англ. calculate, "вычислять"
def calc_perimeter(side_a, side_b):
    return (side_a + side_b) * 2

# функция для вычисления
# площади прямоугольника
def calc_square(side_a, side_b):
    return side_a * side_b

# здесь подготовка результата
def show_info(side_a, side_b):
    p = calc_perimeter(side_a, side_b)
    s = calc_square(side_a, side_b)
    text = 'Периметр = ' + str(p) + ' м., '
    text += 'площадь = ' + str(s) + ' кв.м.'
    return text
```

Вызов функции

In

```
hello(4)    # вызов с аргументом 4
hello(10)   # вызов с аргументом 10
hello(15)   # ещё один вызов функции
hello(20)   # и ещё один вызов
```

Out

```
Доброй ночи!
Доброе утро!
Добрый день!
Добрый вечер!
```

In

```
say_hello(10, 'Тимур')
say_hello(14, 'Елена')
say_hello(20)
```

Out

```
Доброе утро, Тимур!
Добрый день, Елена!
Добрый вечер!
```

При вызове функции можно явно указывать, какому аргументу какое значение соответствует. В таком случае порядок следования аргументов в скобках роли не играет:

In

```
say_hello(hour=10, name='Тимур')
say_hello(name='Елена', hour=14)
say_hello(hour=20)
```

Out

```
Доброе утро, Тимур!
Добрый день, Елена!
Добрый вечер!
```

In

```
a = 8
b = 10
print(show_info(a, b))

# можем произвести расчёты
# и для другого прямоугольника
print(show_info(3, 4))
```

Out

```
Периметр = 36 м., площадь = 80 кв.м.
Периметр = 14 м., площадь = 12 кв.м.
```

Если бы мы ошиблись и забыли вернуть строку `text` из функции `show_info()`, то вывод получился бы странным: `None`. Это специальное значение в Python, и оно обозначает... ничего.