

Python: функции | Я.Шпора

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

Создать и вызвать 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!')
```

```
hello(4) # Вызов с аргументом 4.  
hello(10) # Вызов с аргументом 10.  
hello(15) # Ещё один вызов функции.  
hello(20) # И ещё один вызов.
```

```
# Будет напечатано:  
# Доброй ночи!  
# Доброе утро!  
# Добрый день!  
# 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 + '!')

# Вызовы функции.
say_hello(10, 'Тимур')
say_hello(14, 'Елена')
say_hello(20)

# Будет напечатано:
# 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:

```

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

# Будет напечатано:
# Доброе утро, Тимур!
# Добрый день, Елена!
# Добрый вечер!

```

Вернуть значения из функции

```

# Функция для вычисления периметра прямоугольника.
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

a = 8
b = 10
print(show_info(a, b))
# Можно рассчитать площадь другого прямоугольника.
print(show_info(3, 4))

# Будет напечатано:
# Периметр = 36 м., площадь = 80 кв.м.
# Периметр = 14 м., площадь = 12 кв.м.

```

Если бы из функции `show_info()` не вернули строку `text`, то результатом работы программы было бы `None`. Это специальное значение в Python, и оно обозначает «ничего».