

Функции

Python

Зачем вообще нужно?

Функции — одна из ключевых конструкций в программировании. Без них невозможно сделать практически ничего.

Чтобы выразить любую произвольную операцию, в программировании существует понятие функция. Функции бывают встроенные и добавленные программистом. С одной встроенной функцией мы уже знакомы — это `print()`.

Примеры встроенных функций?

```
# Вызов функции len с параметром 'Hello!'  
result = len('Hello!')  
print(result)  # => 6
```

Параметры или аргументы – это информация, которую функция получает при вызове. На основе этой информации функция обычно вычисляет и выдает результат.

```
result = pow(2, 3)  # 2 * 2 * 2  
print(result)  # => 8
```

Способы объявления

Функции в Python похожи на математические функции из алгебры. Например, в алгебре функция определяется как-то так: $f(x) = x * 2$

Левая часть определяет функцию f , принимающую один параметр, x . А правая часть – это определение функции, которое использует переданный параметр x , чтобы произвести вычисление и вернуть результат. В этом случае значением функции является ее параметр, умноженный на два.

```
def имя\_функции(параметры) :  
    определение _функции
```


Математическая функция $f(x) = x * 2$ в Python будет выглядеть вот так:

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

Ключевое слово `return` используется для определения значения, которое функция возвращает при вызове.

Вызов функции:

```
f(2)
```

У функции может быть один параметр, несколько параметров или вообще их не быть. Чтобы определить функцию, не требующую параметров, оставьте круглые скобки пустыми.

```
def f():  
    return 1 + 1  
  
result = f()  
print(result)    # 2
```

Если хотите, чтобы функция принимала больше одного параметра, отделите каждый параметр в скобках запятой.

```
def f(x, y, z):  
    return x + y + z
```

```
result = f(1, 2, 3)  
print(result)  # 6
```

Наконец, функция не обязана содержать инструкцию `return`. Если функции нечего возвращать, она возвращает значение `None`.

```
def f():  
    z = 1 + 1  
    result = f()  
    print (result)  # None
```