



**DS**  
программирование

# Функции



# Введение



## Вопросы для повторения



## Вопросы для повторения

1

2

3

4

5



## Вопросы для повторения

1

Как создать строку в Python?

2

3

4

5



## Вопросы для повторения

1

Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.

2

3

4

5



## Вопросы для повторения

1

Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.

2

Какие операции можно выполнять со строками?

3

4

5



## Вопросы для повторения

1

**Как создать строку в Python?** нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.

2

**Какие операции можно выполнять со строками?** Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.

3

4

5





## Вопросы для повторения

1

Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.

2

Какие операции можно выполнять со строками? Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.

3

Что такое цикл? Какие типы циклов вы изучили?

4

5



## Вопросы для повторения

1

**Как создать строку в Python?** нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.

2

**Какие операции можно выполнять со строками?** Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.

3

**Что такое цикл? Какие типы циклов вы изучили?** конструкция в программировании, которая позволяет выполнять повторяющиеся операции или блоки кода. | `for ... / while ...`

4

5



## Вопросы для повторения

- 1** Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.
- 2** Какие операции можно выполнять со строками? Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.
- 3** Что такое цикл? Какие типы циклов вы изучили? конструкция в программировании, которая позволяет выполнять повторяющиеся операции или блоки кода. | `for ... / while ...`
- 4** Что такое словарь в Python и как получить значение по ключу?
- 5**



## Вопросы для повторения

- 1** Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.
- 2** Какие операции можно выполнять со строками? Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.
- 3** Что такое цикл? Какие типы циклов вы изучили? конструкция в программировании, которая позволяет выполнять повторяющиеся операции или блоки кода. | `for ... / while ...`
- 4** Что такое словарь в Python и как получить значение по ключу? коллекция, которая хранит пары ключ-значение. Чтобы получить значение из словаря по ключу, используется синтаксис `my_dict[key]`, где `my_dict` - имя словаря, а `key` - ключ, по которому нужно получить значение.
- 5**



## Вопросы для повторения

- 1** Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.
- 2** Какие операции можно выполнять со строками? Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.
- 3** Что такое цикл? Какие типы циклов вы изучили? конструкция в программировании, которая позволяет выполнять повторяющиеся операции или блоки кода. | `for ... / while ...`
- 4** Что такое словарь в Python и как получить значение по ключу? коллекция, которая хранит пары ключ-значение. Чтобы получить значение из словаря по ключу, используется синтаксис `my_dict[key]`, где `my_dict` - имя словаря, а `key` - ключ, по которому нужно получить значение.
- 5** Что такое условие в программировании и как используется оператор `if`?



## Вопросы для повторения

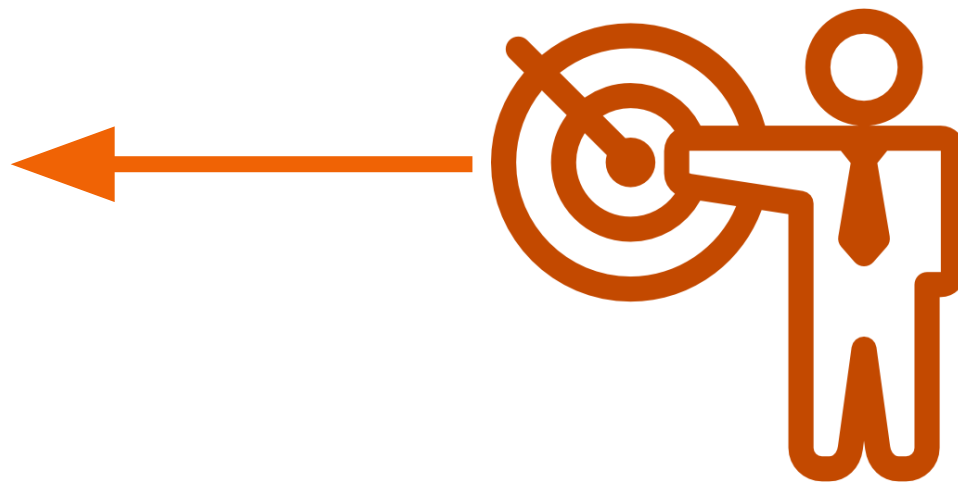
- 1** Как создать строку в Python? нужно заключить текст в одинарные или двойные кавычки, например: `my_string = 'Привет, мир!'` или `my_string = "Hello, world!"`.
- 2** Какие операции можно выполнять со строками? Со строками можно выполнять операции конкатенации (объединение строк), получение подстроки, поиск подстроки, замена символов и другие.
- 3** Что такое цикл? Какие типы циклов вы изучили? конструкция в программировании, которая позволяет выполнять повторяющиеся операции или блоки кода. | `for ... / while ...`
- 4** Что такое словарь в Python и как получить значение по ключу? коллекция, которая хранит пары ключ-значение. Чтобы получить значение из словаря по ключу, используется синтаксис `my_dict[key]`, где `my_dict` - имя словаря, а `key` - ключ, по которому нужно получить значение.
- 5** Что такое условие в программировании и как используется оператор `if`? выражение, которое может быть истинным или ложным. Оператор `if` используется для выполнения блока кода, если условие истинно

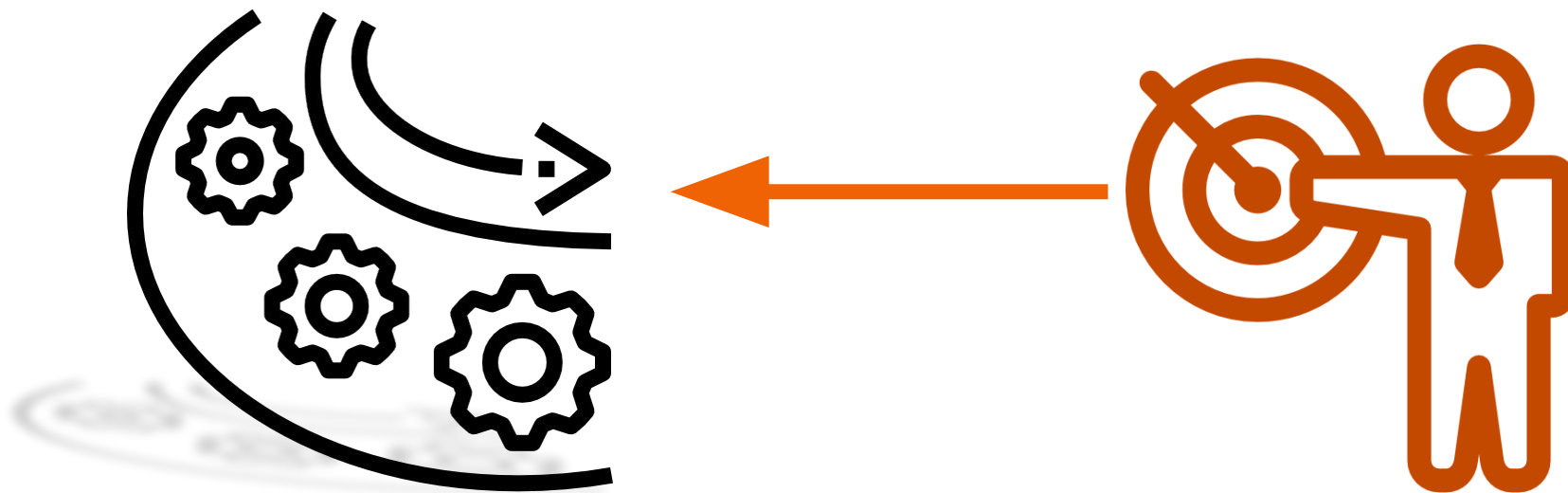


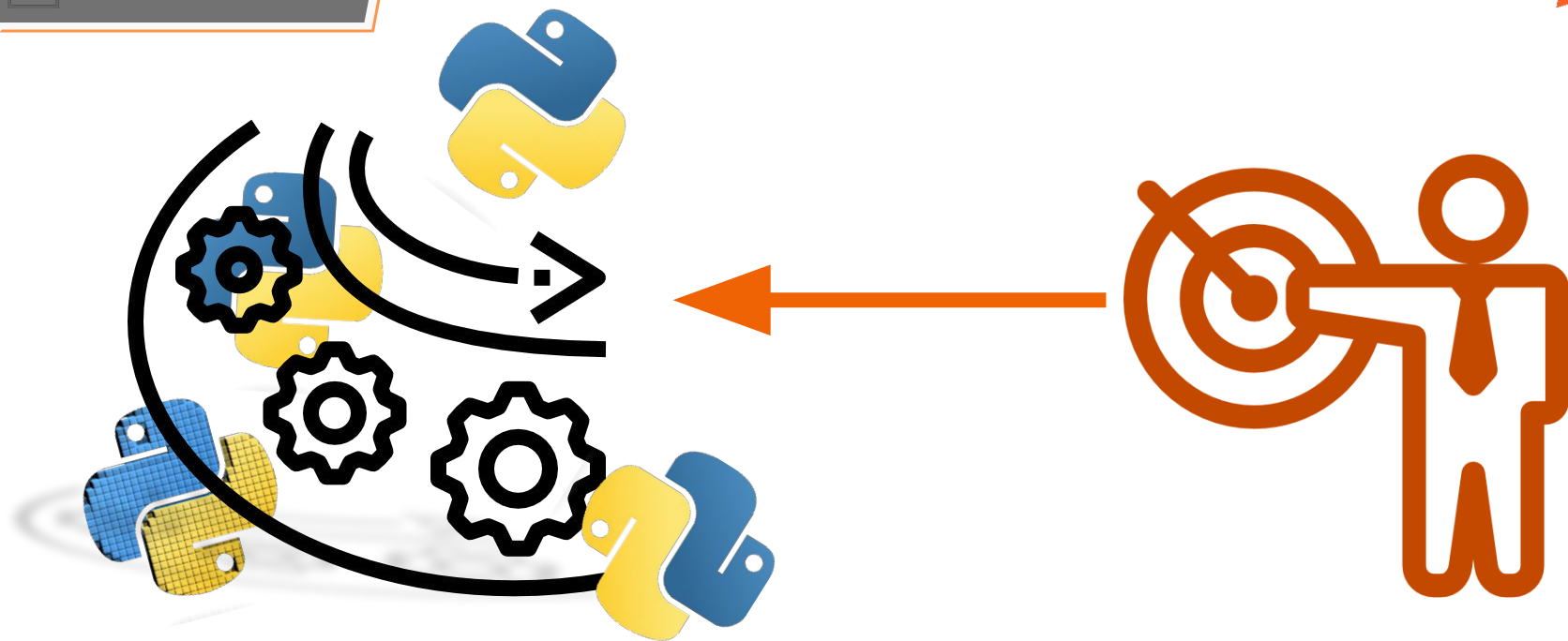
# Введение

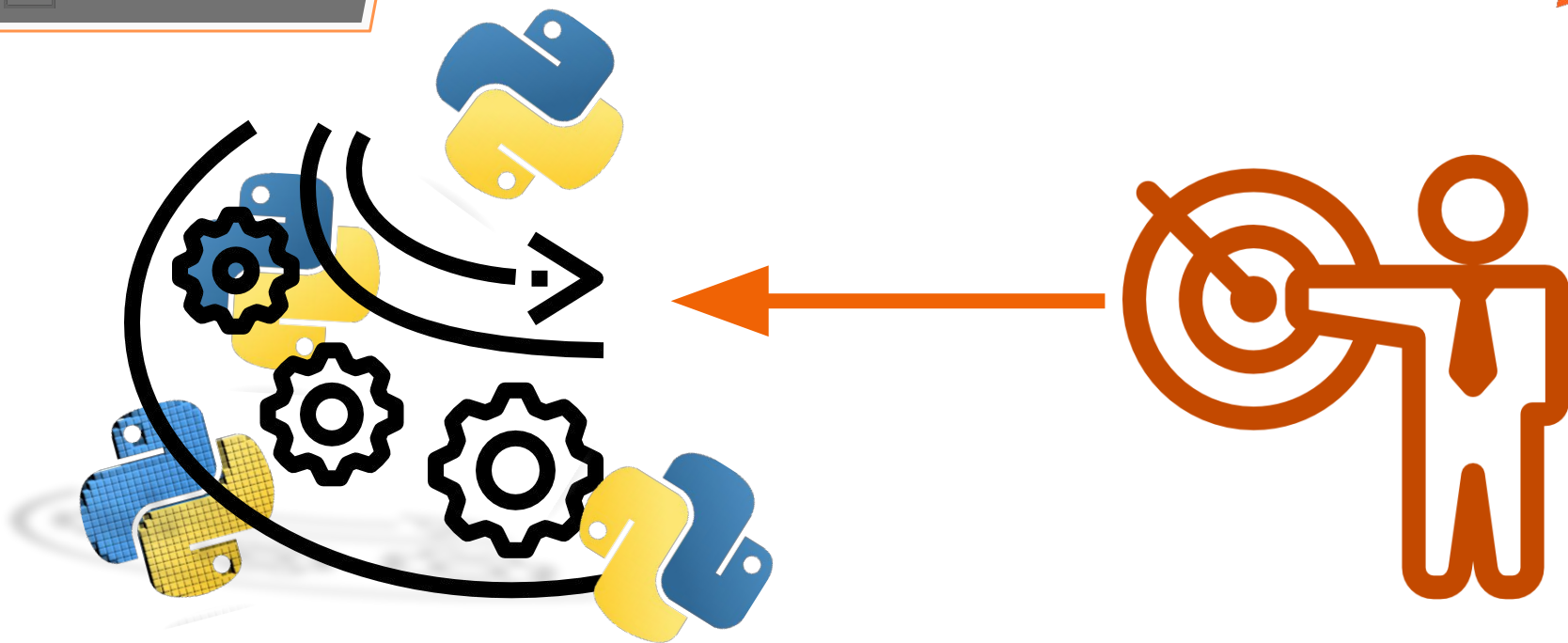














```
def имя_функции(аргументы) :  
    # блок кода функции  
    # выполняемые операции  
    # возвращаемые значения
```



```
def имя_функции(аргументы) :  
    # блок кода функции  
    # выполняемые операции  
    # возвращаемые значения
```

**Функция** – это блок кода, который может принимать некоторые входные значения (аргументы), выполнять определенные действия и возвращать результат.



**return**



**return** – оператор, позволяющий функции **вернуть** значение в место, где она была вызвана. Весь код, написанный после вызова оператора **return**, **не будет выполняться**.





**return** – оператор, позволяющий функции **вернуть** значение в место, где она была вызвана. Весь код, написанный после вызова оператора **return**, **не будет выполняться**.



**return** – оператор, позволяющий функции **вернуть** значение в место, где она была вызвана. Весь код, написанный после вызова оператора **return**, **не будет выполняться**.

Функции могут быть вызваны **любое** количество раз внутри других функций.





## Аргументы функции



Аргументы функции



```
graph TD; A[Аргументы функции] --> B[ ]; A --> C[ ]; A --> D[ ]
```

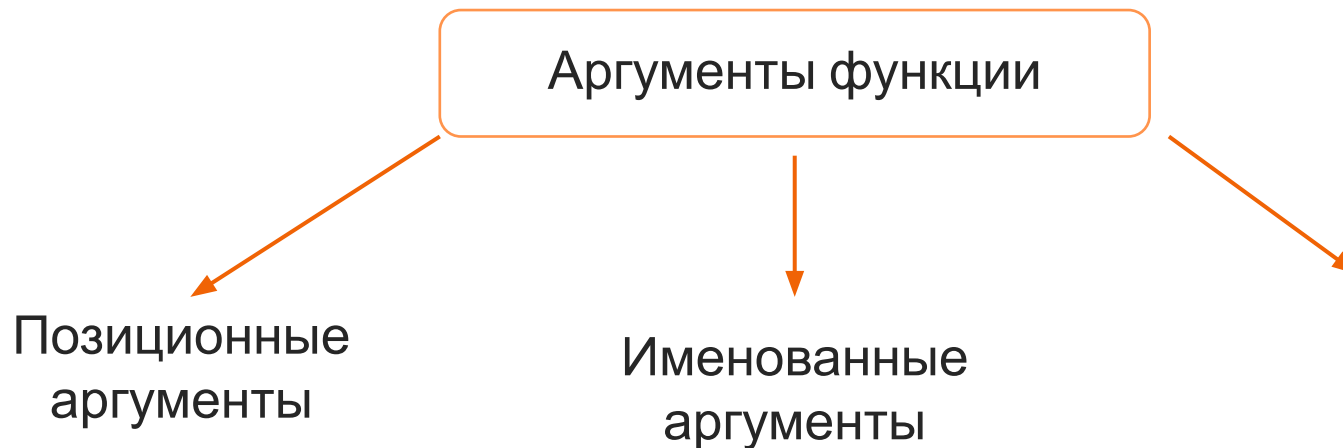


Аргументы функции



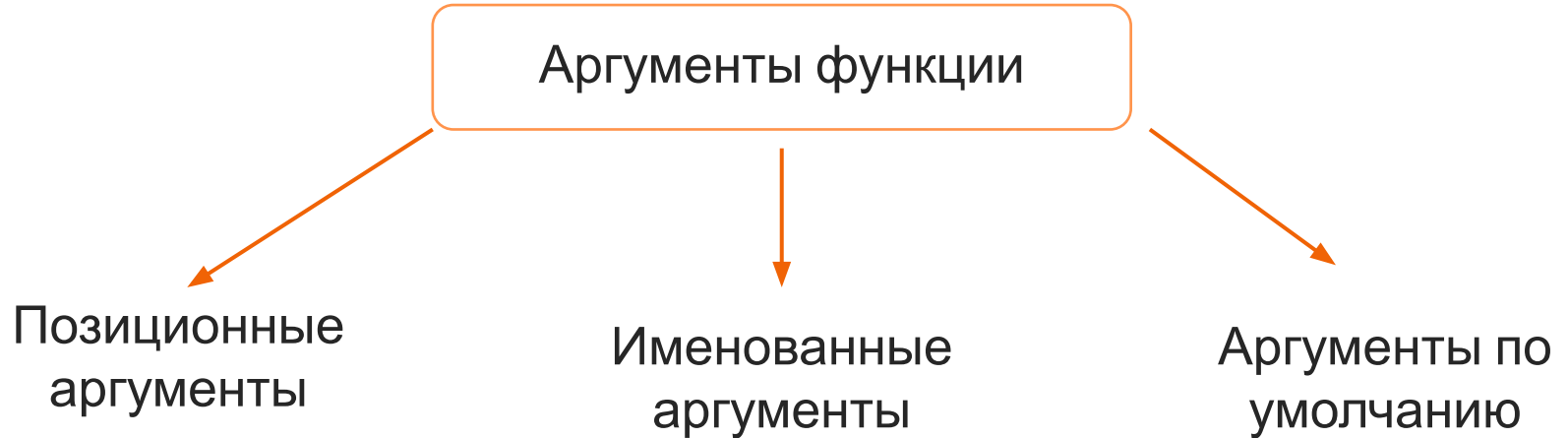
```
graph TD; A[Аргументы функции] --> B[Позиционные аргументы]; A --> C[ ]; A --> D[ ]
```

Позиционные  
аргументы









```
def circle_area(rad):  
    pi = 3.14  
    return pi * rad**2  
result = sum_square (3, 4)
```



```
def circle_area(rad):  
    pi = 3.14  
    return pi * rad**2  
result = sum_square(3, 4)
```

Позиционный аргумент



Аргументы функции

Позиционные аргументы

Именованные аргументы

Аргументы по умолчанию

Именованный аргумент

```
def circle_area(rad):  
    pi = 3.14  
    return pi * rad**2  
result = sum_square (3, 4)
```

Позиционный аргумент



Аргументы функции

Позиционные аргументы

Именованные аргументы

Аргументы по умолчанию

Именованный аргумент

```
def circle_area(rad=1):  
    pi = 3.14  
    return pi * rad**2  
result = sum_square(3, 4)
```

Аргумент по умолчанию

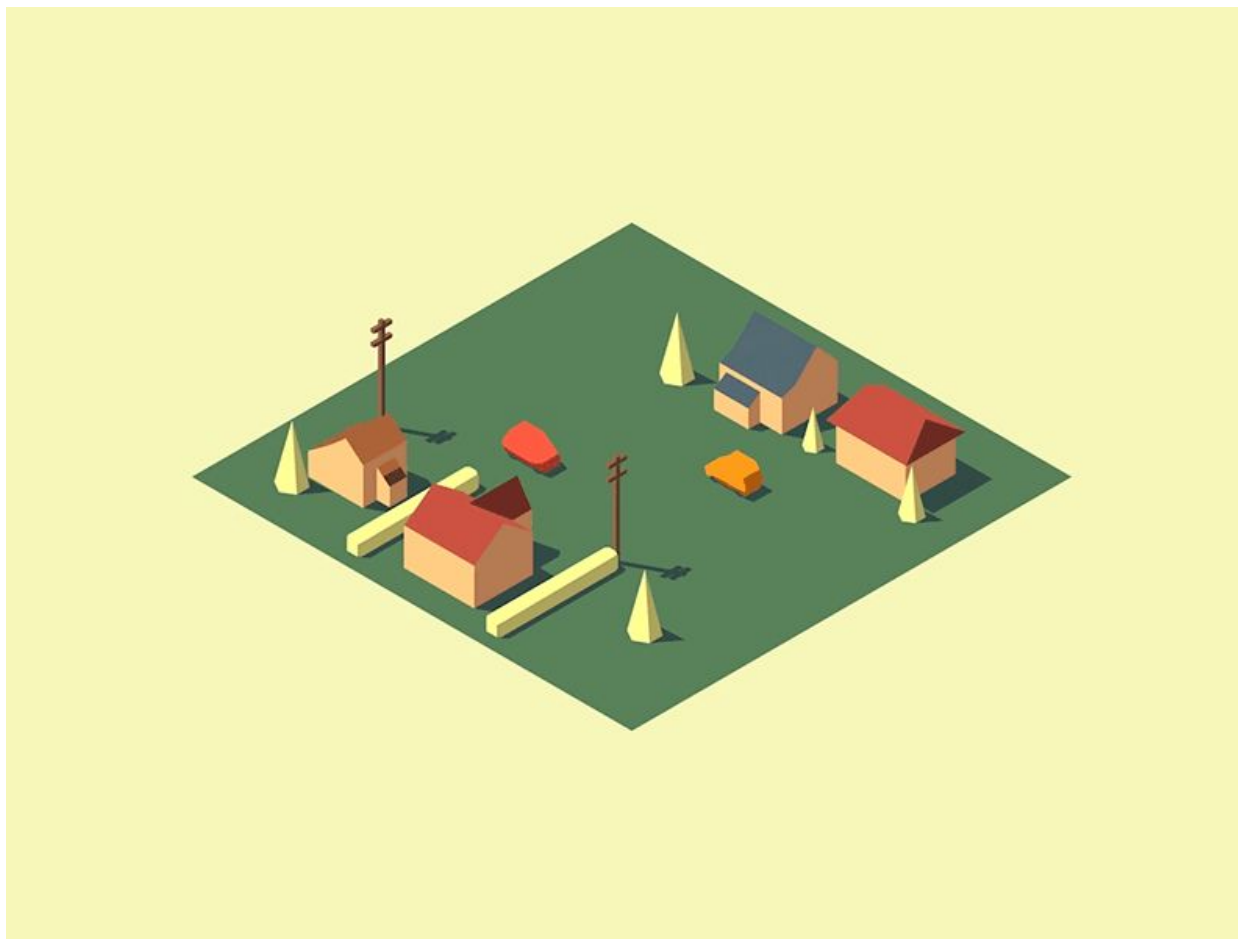




## Рекурсивная функция —



**Рекурсивная функция** — это функция, которая в своем теле содержит обращение к самой себе







$$F(n) = 10 - F(n - 1)$$



$$F(n) = 10 - F(n - 1)$$

$$F(5)$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) =$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) =$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) =$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = \dots$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = \dots$$

$$F(n) = 10 - F(n - 1)$$

$$F(1) = 1$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) =$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = \dots$$

$$F(n) = 10 - F(n - 1)$$

$$F(1) = 1$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = 10 - 10 + 10 - 10 + F(1) =$$





$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = \dots$$

$$F(n) = 10 - F(n - 1)$$

$$F(1) = 1$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = 10 - 10 + 10 - 10 + F(1) = 10 - 10 + 10 - 10 + 1 =$$



$$F(n) = 10 - F(n - 1)$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = \dots$$

$$F(n) = 10 - F(n - 1)$$

$$F(1) = 1$$

$$F(5) = 10 - F(4) = 10 - 10 + F(3) = 10 - 10 + 10 - F(2) = 10 - 10 + 10 - 10 + F(1) = 10 - 10 + 10 - 10 + 1 = 1$$





**Библиотека** – файл с кодом, который можно подключить к своей программе и воспользоваться его содержимым для решения своих задач



**Библиотека** – файл с кодом, который можно подключить к своей программе и воспользоваться его содержимым для решения своих задач

**import** – команда, предназначенная для подключения библиотек



**Библиотека** – файл с кодом, который можно подключить к своей программе и воспользоваться его содержимым для решения своих задач

**import** – команда, предназначенная для подключения библиотек

Конструкцией `from <название> import *` лучше пользоваться при написании **небольших** программ



ИТОГ





## Что такое функция?





## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат



## Что такое функция?

*Функция – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат*

*С помощью какого ключевого слова можно объявить функцию?*



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`

С помощью какого оператора функция возвращает значение?



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`

С помощью какого оператора функция возвращает значение?

`return`



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`

С помощью какого оператора функция возвращает значение?

`return`

## Что такое рекурсивная функция?



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`

С помощью какого оператора функция возвращает значение?

`return`

## Что такое рекурсивная функция?

**Рекурсивная функция** – это функция, вызывающая сама себя



## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?

`def`

С помощью какого оператора функция возвращает значение?

`return`

## Что такое рекурсивная функция?

**Рекурсивная функция** – это функция, вызывающая сама себя

С помощью какого оператора можно подключить библиотеку?





## Что такое функция?

**Функция** – блок кода, принимающий входные значения, выполняющий определенные действия и возвращающий результат

С помощью какого ключевого слова можно объявить функцию?  
`def`

С помощью какого оператора функция возвращает значение?  
`return`

## Что такое рекурсивная функция?

**Рекурсивная функция** – это функция, вызывающая сама себя

С помощью какого оператора можно подключить библиотеку?  
`import`



# Контрольная работа по Модулю № 1



# Контрольная работа по Модулю № 1

## Занятие 7

Уроки:

Теория

Видео

ДЗ

Функции

1 (0/1)

2 (0/2)

4 (0/0/4)

Точка контроля

Итоговая аттестация

Начать

45 мин

20



# Контрольная работа по Модулю № 1

## Занятие 7

Уроки:

Теория

Видео

ДЗ

Функции

1 (0/1)

2 (0/2)

4 (0/0/4)

Точка контроля

Итоговая аттестация

Начать

45 мин

20

Удачи!





**DS**  
программирование

**Спасибо за внимание!**

---