

```
1
2
3
4
5 Введение в 'программирование' {
6   [на Python]
7
8
9
10   < Подготовил: Калайда Матвей >
11
12 }
13
14
```



# Общая 'Информация' ;

Python – это интерпретируемый, высокоуровневый язык программирования, созданный Гвидо ван Россумом в конце 1980-х годов.

Он был разработан с упором на читаемость кода, простоту и эффективность разработки.

1       Общая    ‘Информация’;  
2

3   Python поддерживает несколько  
4 парадигм программирования,  
5 включая процедурное,  
6 объектно-ориентированное и  
7 функциональное программирование.  
8  
9  
10  
11  
12  
13  
14



# История 'языка программирования' }

История Python

связана с языком

программирования ABC,

разработанным в конце 1970-х  
годов в Нидерландах.

Гвидо ван Россум

был разочарован сложностью

языка ABC и решил создать

новый язык, который сочетал бы

простоту и мощь

программирования

}

▼ PYTHON-0.9.1

> demo

> doc

> lib

> src

≡ python.man

≡ README

≡ README.reconstructed

```
1 [Python в значительной
2 степени вдохновлен языками
3 программирования, такими
4 как C, Modula-3, Lisp и
5 Perl. Он имеет
6 синтаксический сахар,
7 который делает код более
8 понятным и читабельным.]
9
10
11
12
13 }
14
```

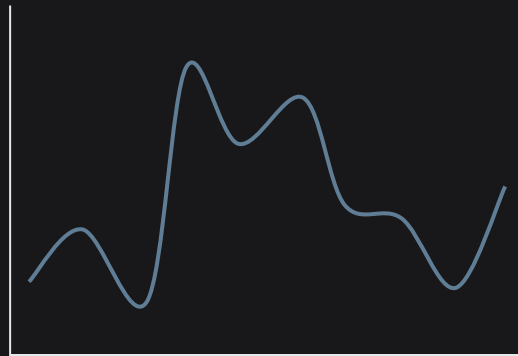




Python является  
одним из самых  
популярных языков  
программирования  
в мире.

По данным исследования за 2022 год самый популярный язык в мире — JavaScript. Рейтинг популярности языков программирования выглядит так:

- 1 место — JavaScript;
- 2 место — Python;
- 3 место — Java;
- 4 место — TypeScript;
- 5 место — C#;
- 6 место — C++;
- 7 место — PHP;
- 8 место — Shell;
- 9 место — C;
- 10 место — Ruby.



```
1
2 Python часто используется
3 в разработке
4 веб-приложений, научных
5 вычислениях, анализе
6 данных, машинном
7 обучении, искусственном
8 интеллекте, автоматизации
9 задач и многих других
10 областях.
11
12 }
13
14
```



python

# Особенности Python

**Простота**

**Понятность**

Python также поддерживает строгий синтаксис  
и динамическую типизацию, что позволяет  
программистам писать более гибкий и  
экономичный код.

}



1 В **Python** есть несколько  
2 основных числовых типов  
3 данных, которые можно  
4 использовать для работы с  
5 числами в программе.  
6

7 1. Целые числа (**int**):  
8 Целочисленные значения  
9 представляются числами без  
10 десятичной части. Например,  
11 **5**, **-3**, **0** являются целыми  
12 числами.  
13 }  
14

## Phyton +

```
1 # Целые числа
2 x = 5
3 y = -3
4 z = 0
5
```

```
1  2. Вещественные числа
2  (float):
3  Вещественные числа
4  представляются числами с
5  десятичной частью. Например,
6  3.14, -2.5, 0.0 – это
7  вещественные числа.
8  Вещественные числа в Python
9  представлены с
10  использованием типа данных
11  float.
12
13 }
14
```

Phyton +

```
1  # Вещественные числа
2  a = 3.14
3  b = -2.5
4  c = 0.0
5
```

### 3. Комплексные числа

(complex):

Комплексные числа в Python  
представлены в виде  $a + bj$ ,  
где  $a$  и  $b$  – вещественные  
числа, а  $j$  – мнимая единица.  
Например,  $2 + 3j$ ,  $-1.5 + 2j$   
– это комплексные числа.

}

Phyton +

```
1 # Комплексные числа
2 d = 2 + 3j
3 e = -1.5 + 2j
```

```
1  4. Длинные целые числа
2  (long):
3  Длинные целые числа
4  представляются целыми
5  числами произвольной
6  длины.
7
8  В Python длинные целые
9  числа представлены с
10 использованием типа
11 данных long.
12
13 }
14
```

Phyton +

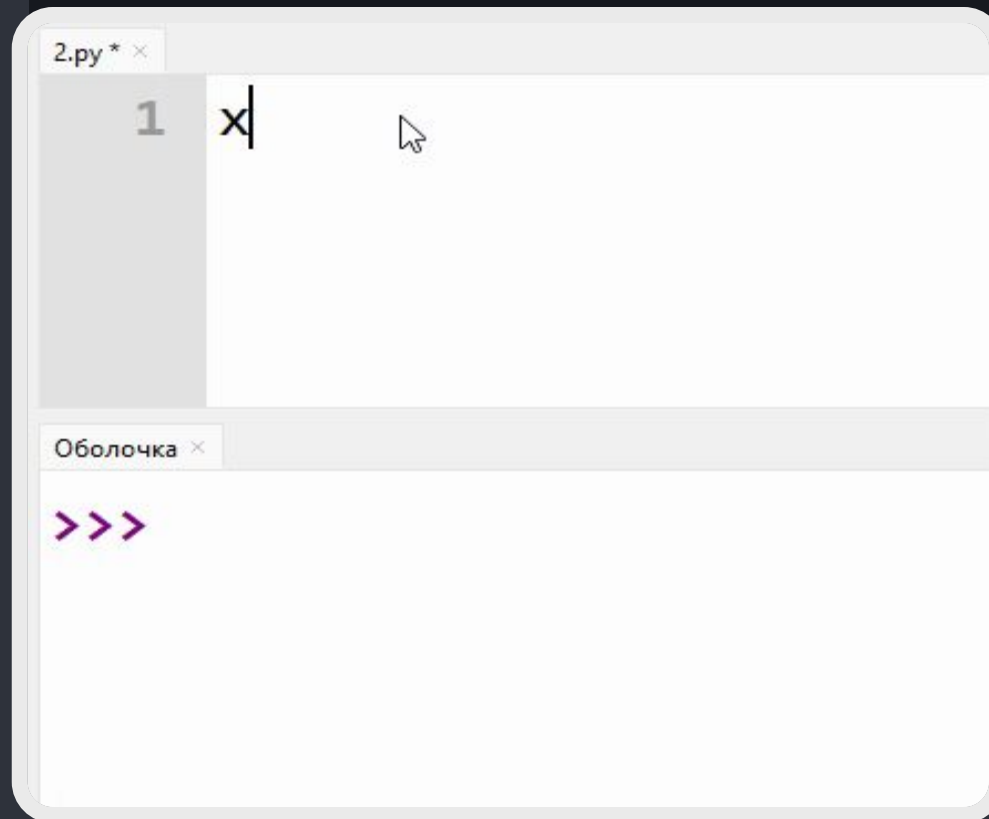
```
1  # Длинные целые числа
2  f = 123456789012345678901234567890
3
```

```
1
2 1. Сложение (+):
3   Операция сложения
4   используется для
5   сложения двух чисел.
6   Это может быть как
7   целые числа, так и
8   вещественные числа.
9
10
11
12
13 }
14
```

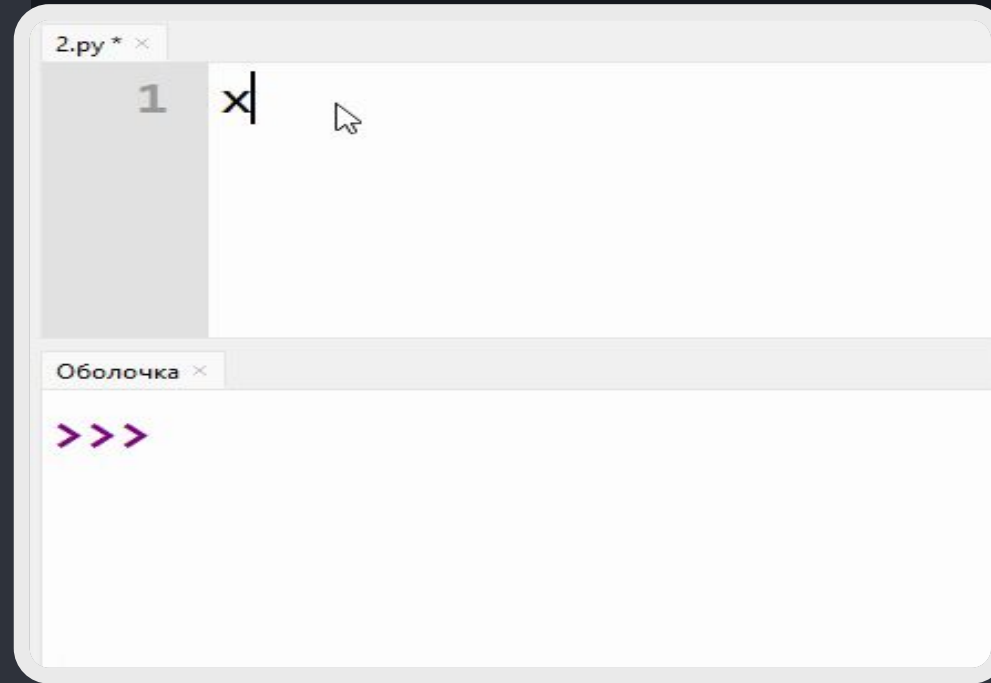
```
2.py * x
1 x = 3+5
2 print(x)
3 y = 3.14+2.5|
```

```
Оболочка x
>>>
```

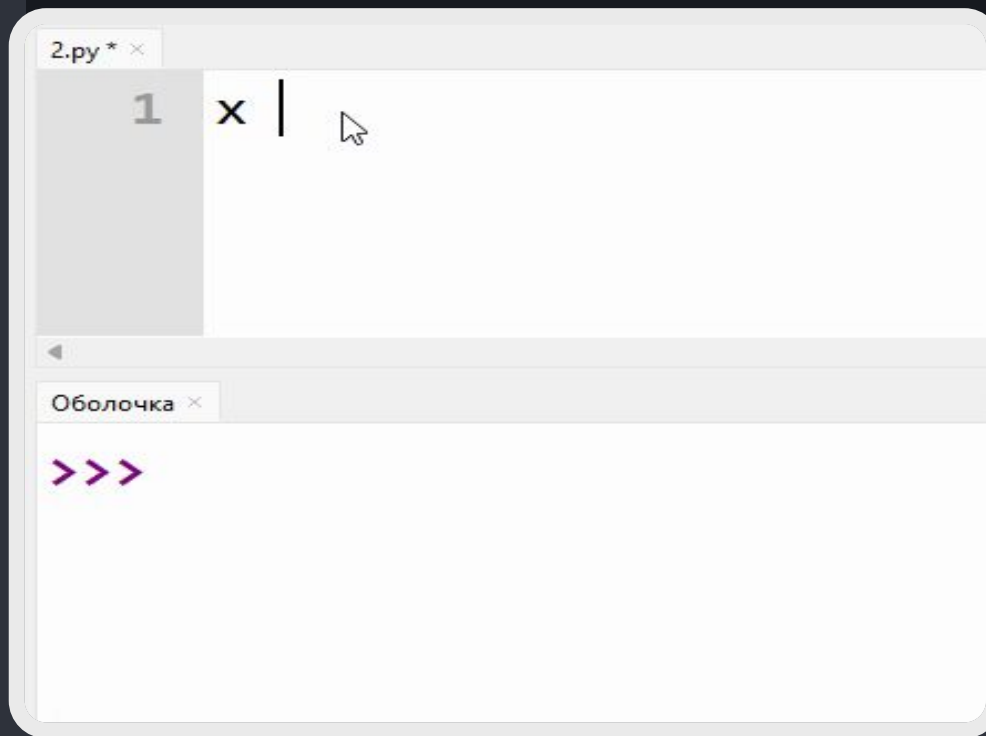
```
1
2
3 2. Вычитание (-):
4   Операция вычитания
5   используется для
6   вычитания одного числа
7   из другого.
8
9
10
11
12
13 }
14
```



```
1
2
3
4 3. Умножение (*):
5   Операция умножения
6   используется для
7   перемножения двух чисел.
8
9
10
11
12
13 }
14
```

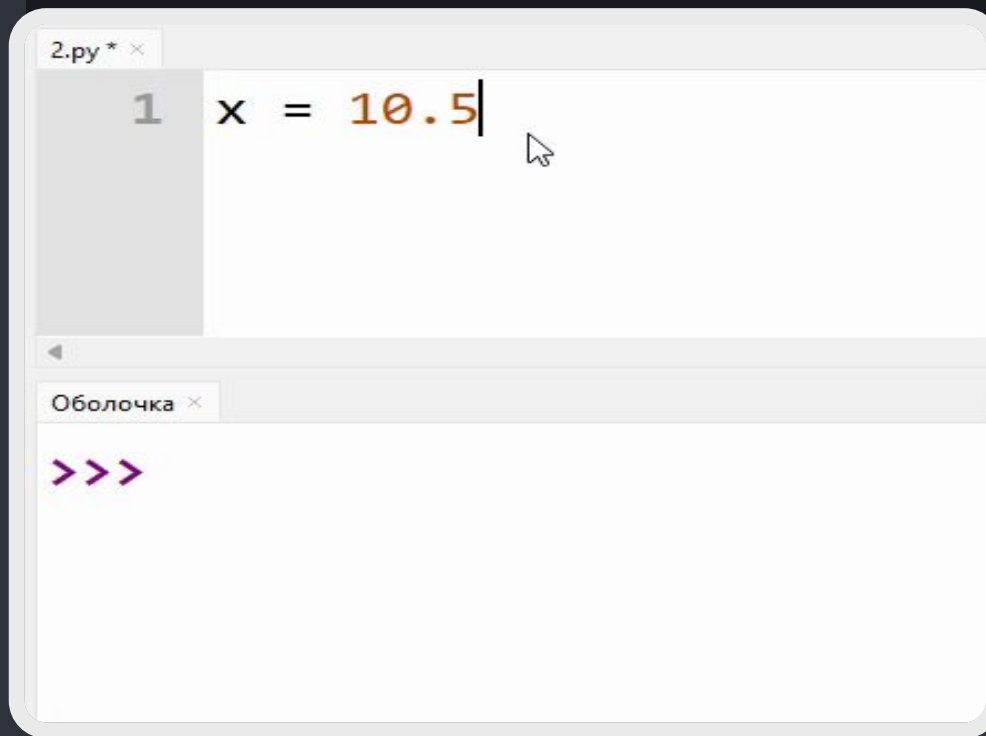


```
1
2 4. Деление (/):
3   Операция деления
4   используется для
5   деления одного числа
6   на другое.
7   Результатом всегда
8   будет вещественное
9   число.
10
11
12
13 }
14
```

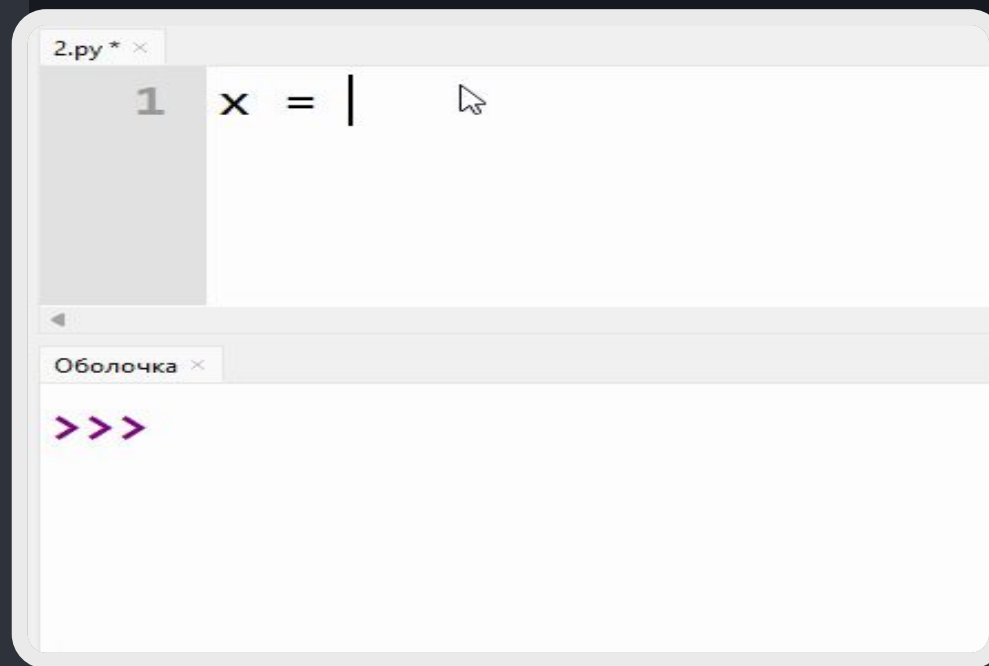




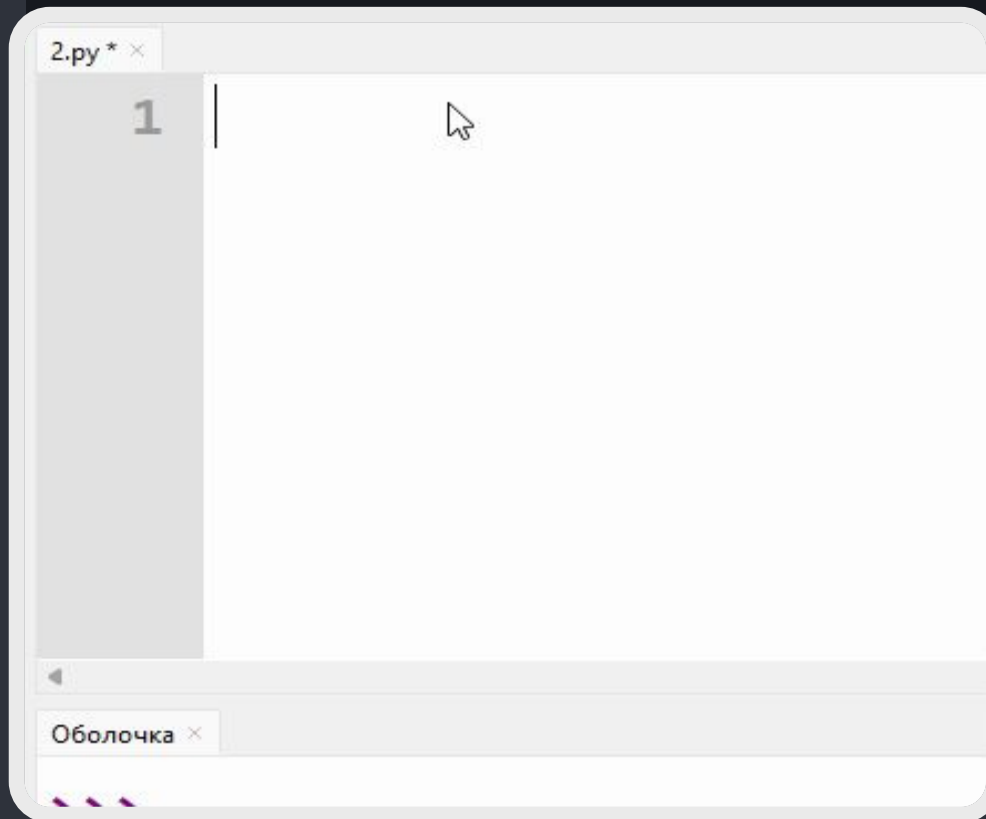
```
1
2 5. Целочисленное
3 деление (//):
4 Операция
5 целочисленного
6 деления возвращает
7 только целую часть
8 результата деления.
9
10
11
12
13 }
14
```



```
1
2 6. Возведение в
3 степень (**):
4 Операция возведения в
5 степень используется
6 для возведения числа
7 в определенную
8 степень.
9
10
11
12
13 }
14
```



```
1
2 В Python также есть
3 встроенные функции,
4 такие как abs(),
5 round(), pow() и
6 другие, которые могут
7 быть использованы для
8 выполнения
9 дополнительных
10 математических
11 операций над числами.
12 }
13
14
```



## Специфика объявления переменных в Python:



1. Имя переменной должно начинаться с буквы или символа подчеркивания (\_).



2. Имя переменной должно состоять только из букв, цифр и символа подчеркивания. Оно не должно содержать пробелы и специальные символы.



3. Имена переменных чувствительны к регистру. Например, переменная age и переменная Age считаются разными переменными.



4. Python рекомендует использовать snake\_case (все буквы строчные, слова разделяются символом подчеркивания) для именования переменных. Это помогает сделать код более читаемым.

## Специфика объявления переменных в Python:



1. Имя переменной должно начинаться с буквы или символа подчеркивания (\_).



2. Имя переменной должно состоять только из букв, цифр и символа подчеркивания. Оно не должно содержать пробелы и специальные символы.



3. Имена переменных чувствительны к регистру. Например, переменная **age** и переменная Age считаются разными переменными.



4. Python рекомендует использовать snake\_case (все буквы строчные, слова разделяются символом подчеркивания) для именования переменных. Это помогает сделать код более читаемым.



В **Python** можно выделить два основных подхода к **типизации**: **статическую** и **динамическую**. **Статическая типизация** подразумевает объявление типов переменных во время компиляции программы. Это означает, что вы должны явно указывать тип каждой переменной при объявлении. **Динамическая типизация**, с другой стороны, позволяет переменным менять тип во время выполнения программы. Это означает, что вы можете объявлять переменные без указания их типа, а тип будет определяться во время выполнения.

В целом, **Python** – это мощный и гибкий **язык программирования**, который используется во многих областях. Его **простота** и **понятность** делают его **доступным** для начинающих программистов, а его мощные возможности позволяют разработчикам создавать сложные программы и приложения.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14

# Спасибо за внимание!

