

Основы языка Python

Аксентьев Артем Алексеевич

Chillers

3 марта 2020 г.

О чем пойдет речь...

- ❶ Почему стоит изучить Python?
- ❷ Хранение и запуск программ
- ❸ Первая программа
 - Основные типы данных
 - Ввод-вывод данных
 - Файловый ввод-вывод
- ❹ Условия и циклы
- ❺ Повторное использование кода
 - Функции
 - Модули
 - Написание своего модуля

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;
- 2 Кроссплатформенность и многофункциональность;

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;
- 2 Кроссплатформенность и многофункциональность;
- 3 Большое количество модулей.

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;
- 2 Кроссплатформенность и многофункциональность;
- 3 Большое количество модулей.

Особенности языка:

- 1 Python - интерпретируемый язык;

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;
- 2 Кроссплатформенность и многофункциональность;
- 3 Большое количество модулей.

Особенности языка:

- 1 Python - интерпретируемый язык;
- 2 Python это интерактивный язык.

Особенности языка Python

Достоинства языка:

- 1 Низкий порог вхождения;
- 2 Кроссплатформенность и многофункциональность;
- 3 Большое количество модулей.

Особенности языка:

- 1 Python - интерпретируемый язык;
- 2 Python это интерактивный язык.

Недостатки:

- 1 Python медленный.

Как запускать программы?

Программа - файл

Код программы на компьютере хранится в виде файла с расширением «.ру»(но не всегда) и с кодировкой «UTF-8».

Как запускать программы?

Программа - файл

Код программы на компьютере хранится в виде файла с расширением «.ру»(но не всегда) и с кодировкой «UTF-8».

Среды разработки

- 1 IDLE Python;

Как запускать программы?

Программа - файл

Код программы на компьютере хранится в виде файла с расширением «.py»(но не всегда) и с кодировкой «UTF-8».

Среды разработки

- ❶ IDLE Python;
- ❷ IDE:
 - PyCharm;
 - Visual Studio.

Как запускать программы?

Программа - файл

Код программы на компьютере хранится в виде файла с расширением «.ру»(но не всегда) и с кодировкой «UTF-8».

Среды разработки

- ❶ IDLE Python;
- ❷ IDE:
 - PyCharm;
 - Visual Studio.
- ❸ Редакторы кода:
 - Atom;
 - SublimeText;
 - Visual Studio Code

Как запускать программы?

Программа - файл

Код программы на компьютере хранится в виде файла с расширением «.ру»(но не всегда) и с кодировкой «UTF-8».

Среды разработки

- ❶ IDLE Python;
- ❷ IDE:
 - PyCharm;
 - Visual Studio.
- ❸ Редакторы кода:
 - Atom;
 - SublimeText;
 - Visual Studio Code
- ❹ Jupyter Notebook.

Первая программа

Python Hello world

```
1 print("Hello, world")
```

C Hello world

```
1 #include <stdio.h>
2 int main (void)
3 {
4     printf("Hello, World!\n");
5     getchar();
6     return 0;
7 }
```

Основные типы данных

Основные типы в Python:

- Целое число - `int`
- Дробные числа - `float`
- Строки - `str`
- Список - `list`
- Словарь - `dict`
- Множество - `set`

Типы данных

```
a = 5
b = 3.4
c = "Hello"
d = 'world'
e = [1, 2, 4]
f = {1 : "One", 2: "Two", "Three":
    ↪ 3}
g = {1, 2, 3, 4, 4}
```

Ввод данных пользователем

input

```
1 >>>a = input("Введите число от 1 до 5: ")
2 #Введите число от 1 до 5: 4
3 >>>print(a)
4 4
```


Файловый ввод-вывод

Для работы с файлом, как и в любом другом языке программирования его следует открыть:

```
f = open('text.txt', 'r')
```

Режим	Обозначение
"r"	открытие на чтение (является значением по умолчанию).
"w"	открытие на запись, содержимое файла удаляется, если файла не существует, создается новый.
"x"	открытие на запись, если файла не существует, иначе исключение.
"a"	открытие на дозапись, информация добавляется в конец файла.
"+"	открытие на чтение и запись
"b"	открытие в двоичном режиме.
"t"	открытие в текстовом режиме (является значением по умолчанию).

Чтение из файла

Чтение из файла

```
1 >>> f = open('text.txt')
2 >>> f.read(1)
3 'H'
4 >>> f.read()
5 'ello world!\nThe end.\n\n'
6 >>> f.close()
```

Чтение из файла

Построчное чтение

```
1 >>> f = open('text.txt')
2 >>> for line in f:
3 ...     line
4 ...
5 'Hello world!\n'
6 '\n'
7 'The end.\n'
8 '\n'
9 >>> f.close()
```

Запись в файл

Построчная запись

```
1 >>> переменная = ["Hello, world", "Привет, мир", "12313"]
2 >>> f = open('text.txt', 'w')
3 >>> for line in переменная:
4 ...     f.write(line + '\n')
5 ...
6 13
7 12
8 6
9 >>> f.close()
```

УСЛОВИЯ

if-elif-else

```
1 a = int(input("Введите первое число: "))
2 b = int(input("Введите второе число: "))
3
4 if a > b:
5     print("Первое число больше второго")
6 elif a < b:
7     print("Второе число больше первого")
8 else:
9     print("Числа равны")
```

Циклы

Обход по коллекции

```
1 list_a = [1, 2, 5]
2 sum = 0
3 for i in list_a:
4     sum += i
5
6 i = 0
7 sum = 0
8 while i < len(list_a):
9     sum += list_a[i]
10    i += 1
```

Циклы

```
for _ in range(stop)
```

```
>>> for i in range(4):
```

```
...     print(i)
```

```
...
```

```
0
```

```
1
```

```
2
```

```
3
```

Циклы

```
for _ in range(stop)
```

```
>>> for i in range(4):
```

```
...     print(i)
```

```
...
```

```
0
```

```
1
```

```
2
```

```
3
```

```
for _ in range(start, stop)
```

```
>>> for i in range(2, 4):
```

```
...     print(i)
```

```
...
```

```
2
```

```
3
```


Циклы

```
for _ in range(stop)
```

```
>>> for i in range(4):  
...     print(i)
```

```
...
```

```
0
```

```
1
```

```
2
```

```
3
```

```
for _ in range(start, stop, step)
```

```
>>> for i in range(1, 4, 2):  
...     print(i)
```

```
...
```

```
1
```

```
3
```

```
for _ in range(start, stop)
```

```
>>> for i in range(2, 4):  
...     print(i)
```

```
...
```

```
2
```

```
3
```

DRY

DRY - Don't repeat yourself. Принцип DRY формулируется как: «Каждая часть знания должна иметь единственное, непротиворечивое и авторитетное представление в рамках системы».

Функции

```
1 def func_name(arg):  
2     print(arg)  
3     return arg  
4  
5  
6 a = func_name(5)  
7 print(a)
```

Работа с модулями

Подключение модуля

```
import math  
  
math.factorial(100)
```

Использование псевдонимов

```
from math import e, pi as p  
  
print(e)  
print(p)
```

Работа с модулями

Импорт атрибутов

```
from math import *  
  
print(e)  
print(pi)
```

Свой модуль

modul1.py

```
1 def hello():
2     print('Hello, world!')
3
4 __all__ = ['hello()']
5 not_import = 10
6 if __name__ == "__main__":
7     hello()
```

Спасибо за внимание!