

Программирование на языке Python. Базовый уровень

Модуль 1. Введение в программирование

Тема 1.2. Ввод и вывод данных (занятие 1)



Особенности Python

1. Любую программу языке Python можно представить, как набор лексем (допустимых символов), записанных в определенном порядке и по определенным правилам.
2. Программа на языке Python может содержать достаточное количество комментариев, каждый комментарий начинается с символа решетка «#».
3. Литералы представляют собой значения, заданные в коде программы, например, числа (25) или строки (“привет”).
4. Каждая строка в программе на языке Python не должна заканчиваться точкой запятой, но если есть необходимость записать несколько инструкций в одну строчку, то их можно разделять точкой с запятой.



В языке Python нет специального раздела описания переменных, в котором указывается тип

Вывод данных, команда print



Кавычки могут быть одинарными или двойными.

Только обязательно ставим одинаковые до и после текста.

Пример.

```
print('Сегодня второе занятие')  
print("Сегодня второе занятие")
```



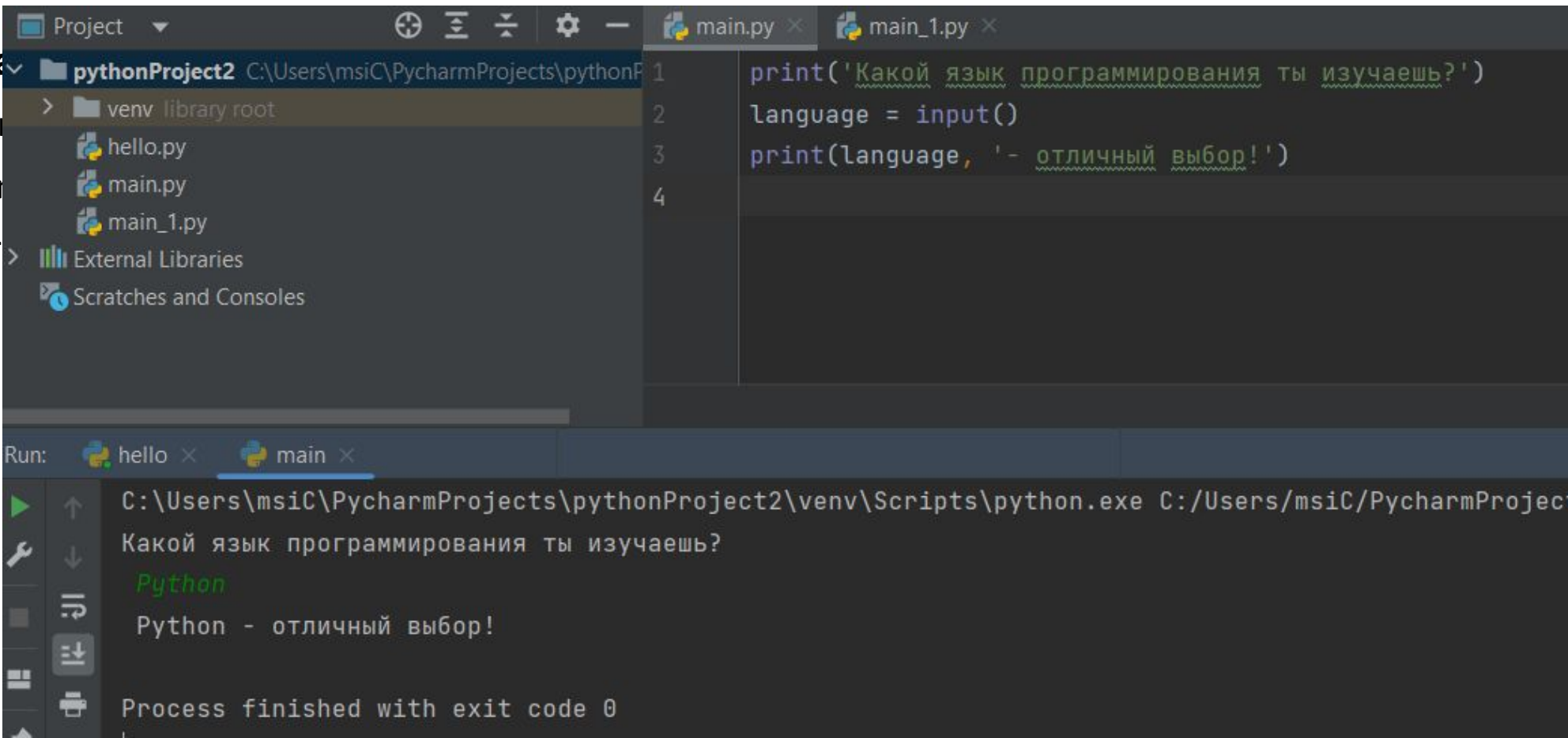
Команда print() позволяет указывать несколько аргументов, в таком случае их надо отделять запятыми.

Пример.

```
print('Я хочу', 'научиться  
программировать', 'на ', 'Python!')
```

Ввод данных, команда input

Для считывания данных в языке Python используется команда input().



The screenshot shows the PyCharm IDE interface. The left sidebar displays the project structure for 'pythonProject2', including a 'venv' directory and files 'hello.py', 'main.py', and 'main_1.py'. The main editor window shows the code in 'main.py':

```
1 print('Какой язык программирования ты изучаешь?')
2 language = input()
3 print(language, '- отличный выбор!')
4
```

Below the editor, the 'Run' console shows the execution of the code:

```
Run: hello x main x
C:\Users\msiC\PycharmProjects\pythonProject2\venv\Scripts\python.exe C:/Users/msiC/PycharmProject
Какой язык программирования ты изучаешь?
Python
Python - отличный выбор!
Process finished with exit code 0
```

Параметр sep

Необязательный параметр `sep` команды `print()` позволяет установить строку, с помощью которой будут разделены аргументы при печати.

Пример.

```
print('a', 'b', 'c', sep='*')  
print('d', 'e', 'f', sep='**')
```

Результатом выполнения такого кода будет:

```
a*b*c  
d**e**f
```



Параметр end

Если перевод строки делать не нужно или требуется указать специальное окончание, то следует явно указать значение для параметра end.

Пример.

```
print('a', 'b', 'c', end='@')  
print('d', 'e', 'f', end='@@')
```

Результатом выполнения такого кода будет:

```
a b c@d e f@@
```

Программный код `print('Python')` равнозначен коду `print('Python', end='\n')`



Имя переменной



- В имени переменной используйте только латинские буквы a-z, A-Z, цифры и символ нижнего подчеркивания (_);
- Имя переменной не может начинаться с цифры;
- Имя переменной по возможности должно отражать её назначение.
- Python – регистрочувствительный язык. Переменная name и Name – две совершенно разные переменные.
- Значение переменной – сохраненная в ней информация. Это может быть текст, число и т.д.

Правильно

my_int

int4

MY_INT

another_int

Неправильно

my-int

4int4

\$MY_INT

another int

Форматированный вывод

Синтаксис функции выглядит следующим образом:

```
<строка>.format( <формат>)
```

«строка» представляет собой значение для форматированного вывода,

«формат» – спецификации формата 'Mini-Language'

f – format только в сокращенном виде, он нужен для того чтобы можно было что-то вставить в таких скобках {}.

Пример

```
name = "World"
print(f"Hello, {name}")
out: "Hello, World"
# Теперь без f
print("Hello, {name}")
out: Hello, {name}
# Теперь при помощи просто format
print("Hello, {}".format(name))
out: "Hello, World"
# Но можно сделать и по другому
print("Hello, %s" % name)
out: "Hello, World"
```


Множественное присваивание переменной



Python позволяет присваивать одно значение нескольким переменным одновременно.

Пример 1.

```
x = y = z = 0  
print(x)  
print(y)  
print(z)
```

Пример 2.

```
j, k, l = "8host",  
1.05, 16  
print(j)  
print(k)  
print(l)
```

PEP 8 – Python Enhanced Proposal

Рекомендация 1. Избегайте использования пробелов перед открывающей скобкой, после которой начинается список аргументов функции.

Правильно: `print('Follow PEP8!')`

Неправильно: `print ('Follow PEP8!')`

Рекомендация 2. После запятой нужен пробел.

Правильно: `print('PEP8', 'Rocks!')`

Неправильно: `print('PEP8','Rocks!')`

Рекомендация 3. Не отделяйте пробелами знак «равно», когда он употребляется для обозначения значения параметра по умолчанию.

Правильно: `print('My name', 'is', 'Python', sep='**', end='+')`

Неправильно: `print('My name', 'is', 'Python', sep = '**', end = '+')`

