

СПЕЦИАЛЬНЫЕ СИМВОЛЫ

 Модуль 2. Занятие 2.



Тема **и цель**

Тема:

Применение специальных символов
в языке Python

Цель занятия:

Изучение способов применения
специальных символов в языке
Python



Глоссарий

- 1. Символ (в информатике)** — это знаки, которые имеют собственное содержание, и в то же время могут в общем виде иметь некоторый другой смысл.
- 2. Специальный символ** — это специальный управляющий символ.



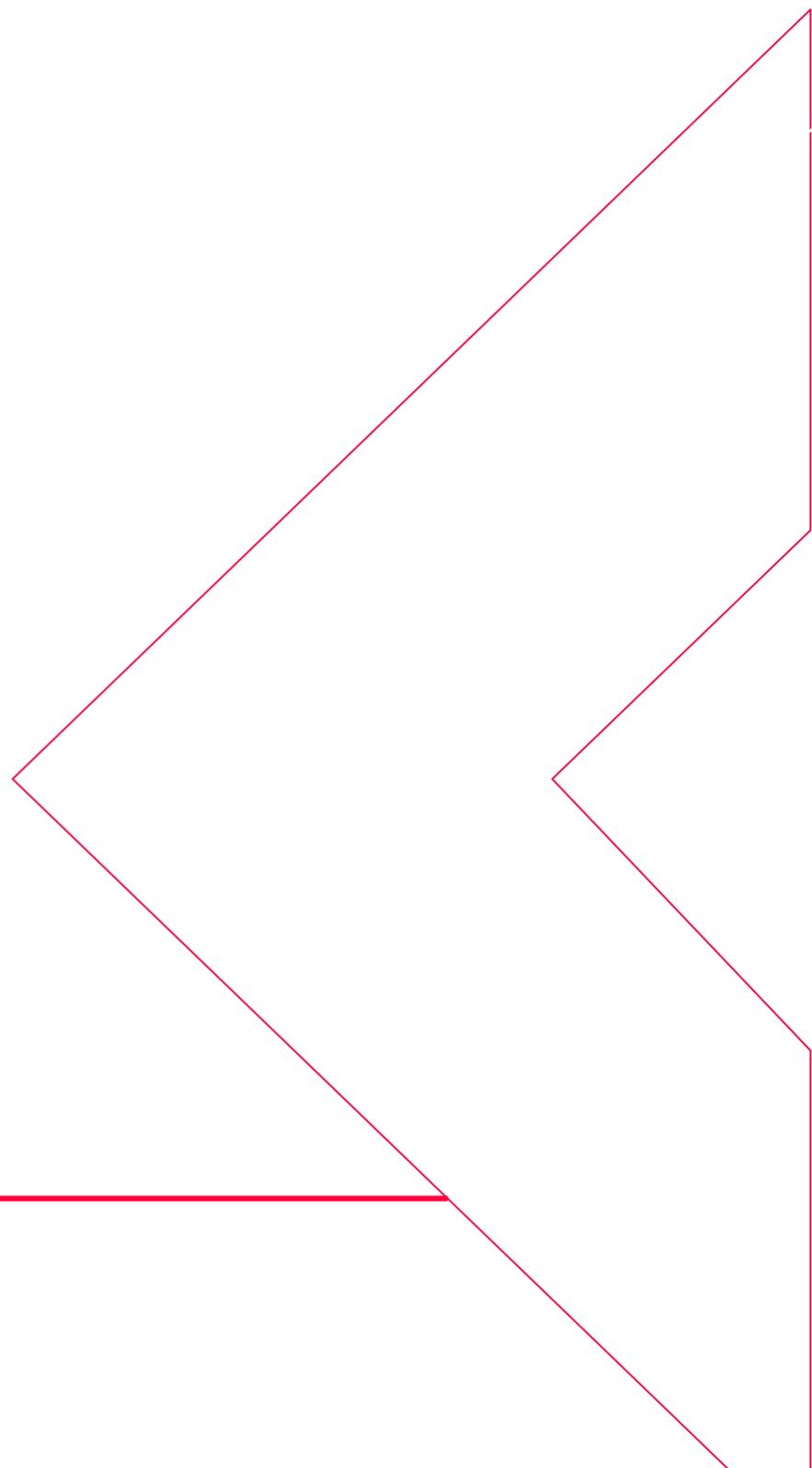
Подтемы

1. Что такое среда разработки?
2. Для чего используется тип переменных `int` в языке Python?
3. Что такое команда `print()` и как ее можно применять в языке Python?
4. Как можно работать с несколькими операторами одновременно в языке Python?
5. Для чего служит ключевое слово `global` в языке Python?



Литералы

СТРОК



Строки в апострофах и в кавычках



```
S = 'spam"s'
```

```
S = "spam's"
```

Строки в апострофах И В КАВЫЧКАХ



"Сырые" строки - подавляют экранирование

Если перед открывающей кавычкой стоит символ 'r' (в любом регистре), то механизм экранирования отключается.

```
s = r'C:\newt.txt'
```

Но, несмотря на назначение, "сырая" строка не может заканчиваться символом обратного слэша. Пути решения:

```
s = r'\n\n\'[:-1]
```

```
s = r'\n\n' + '\\'
```

```
s = '\\n\n'
```

Строки в апострофах И В КАВЫЧКАХ



Строки в тройных апострофах или кавычках

Главное достоинство строк в тройных кавычках в том, что их можно использовать для записи многострочных блоков текста. Внутри такой строки возможно присутствие кавычек и апострофов, главное, чтобы не было трех кавычек подряд.

```
>>> c = '''это очень большая
... строка, многострочный
... блок текста'''
>>> c
'это очень большая\nстрока, многострочный\nблок текста'
>>> print(c)
это очень большая
строка, многострочный
```

блок текста

Строки в апострофах И В КАВЫЧКАХ



```
string = "some\ttext\nnew line here"
```

```
print(string)
```

```
some  text
```

```
new line here
```

```
string = r"D:\dir\new"
```

Строки в апострофах И В КАВЫЧКАХ



Символ	Назначение
<code>\n</code>	Перевод каретки на новую строку
<code>\b</code>	Возврат каретки на один символ назад
<code>\f</code>	Перевод каретки на новую страницу
<code>\r</code>	Возврат каретки на начало строки
<code>\t</code>	Горизонтальная табуляция
<code>\v</code>	Вертикальная табуляция
<code>\a</code>	Подача звукового сигнала
<code>\N</code>	Идентификатор базы данных
<code>\u, \U</code>	16-битовый и 32-битовый символ Unicode
<code>\x</code>	Символ в 16-ричной системе исчисления
<code>\o</code>	Символ в 8-ричной системе исчисления
<code>\0</code>	Символ Null

Строки в апострофах И В КАВЫЧКАХ



```
print('first\nsecond')
```

```
first
```

```
second
```

```
string = "text"
```

```
number = 10
```

```
newString = "this is %s and digit %d" % (string, number)
```

```
print(newString)
```

```
this is text and digit 10
```

Строки в апострофах и в кавычках



```
print(newString)
```

```
text
```

```
string = "text"
```

```
newString = "%+10s" % string
```

Строки в апострофах И В КАВЫЧКАХ



Символ	Назначение
%d, %i, %u	Число в 10-ричной системе исчисления
%x, %X	Число в 16-ричной системе исчисления с буквами в нижнем и верхнем регистре
%o	Число в 8-ричной системе исчисления
%f, %F	Число с плавающей точкой
%e, %E	Число с плавающей точкой и экспонентой в нижнем и верхнем регистре
%c	Одиночный символ
%s, %r	Строка из литерала и обычная
%%	Символ процента

Строки в апострофах И В КАВЫЧКАХ



```
string = "text"

number = 10

newString = "this is {0} and digit {1}".format(string, number)

print(newString)

this is text and digit 10
```

Строки в апострофах и в кавычках



```
string = "text"  
  
newString = "{:*^10}".format(string)  
  
print(newString)  
  
***text***
```

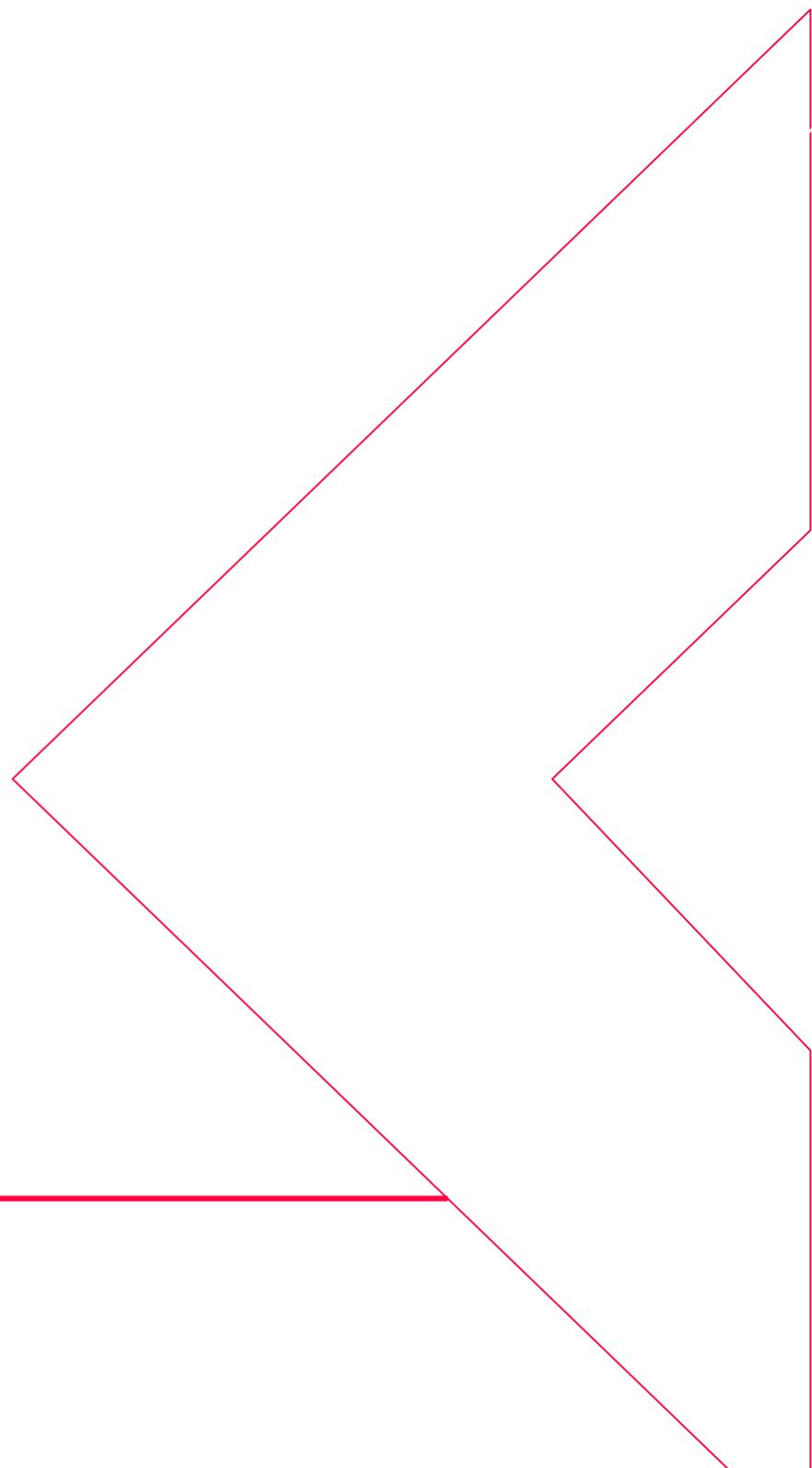
Строки в апострофах И В КАВЫЧКАХ



Символ	Назначение
'<'	Выравнивание строки по левому краю с символами-заполнителями справа
'>'	Выравнивание строки по правому краю с символами-заполнителями слева
'='	Выравнивание с символами-заполнителями после знака числа, но перед его цифрами
'^'	Выравнивание строки по центру с символами-заполнителями с обеих сторон
'+'	Применение знака для любых чисел
'-'	Применение знака для отрицательных чисел и ничего для положительных
' '	Применение знака для отрицательных чисел и пробела для положительных



Практические задачи





Задача 1

Написать программу с символом выравнивания строки по левому краю с символами-заполнителями справа.



Решение

Напишем код для решения данной практической задачи и посмотрим на вывод:

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in a file named `main.py`:

```
1  
2 a = 'abc'  
3 print(a.ljust(3))  
4  
5
```

Below the editor, the Run console shows the output of the program:

```
C:\Users\user\PycharmProjects\pythonProject3\venv  
abc  
  
Process finished with exit code 0
```



Задача 2

Написать программу с применением горизонтальной табуляции.



Решение

Напишем код для решения данной программы и посмотрим на вывод:

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in `main.py`:

```
1  
2 s = 'q\tw\te'  
3 print('Строка :', s)  
4  
5
```

The code is executed, and the Run console at the bottom shows the output:

```
C:\Users\user\PycharmProjects\pythonProject3\venv  
Строка : q w e  
  
Process finished with exit code 0
```



Задача 3

Написать программу с переводом каретки на новую строку.



Решение

Напишем код для решения данной программы и посмотрим на вывод:

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in `main.py`:

```
1  
2 s = 'q\nw\ne'  
3 print('Строка :', s)  
4  
5
```

The Run window at the bottom shows the execution output:

```
C:\Users\user\PycharmProjects\pythonProject3\venv  
Строка : q  
w  
e  
  
Process finished with exit code 0
```



Задача 4

Написать программу, в которой используется следующая строка: 'a' = ['qwerty']. Необходимо вывести первый символ, затем применить горизонтальную табуляцию. После третьего символа применить вертикальную табуляцию.



Решение

Напишем код для решения данной программы и посмотрим на вывод:

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in `main.py`:

```
1  
2 s = 'q\twe\nrty'  
3 print('Строка :', s)  
4  
5
```

The code uses a string with escape characters to create a multi-line output. The output window at the bottom shows the result of running the code:

```
Run: main ×  
C:\Users\user\PycharmProjects\pythonProject3\venv  
Строка : q we  
rty  
  
Process finished with exit code 0
```



Задача 5

Создать программу, в которой происходит подсчет элементов в строке при помощи функции `len()` с произвольным содержимым



Решение

Напишем код для решения данной программы и посмотрим на вывод:

The screenshot shows the PyCharm IDE interface. The main editor window displays the following Python code in `main.py`:

```
1  
2 s = 'qwerty'  
3 print('Количество элементов :', len(s))  
4  
5
```

Below the editor, the Run console shows the output of the program:

```
Run: main x  
C:\Users\user\PycharmProjects\pythonProject3\venv  
Количество элементов : 6  
Process finished with exit code 0
```



Вопросы

1. Что такое строка?
2. Для чего используется тип переменных `str` в языке Python?
3. Что такое команда `input()` и как ее можно применять в языке Python?
4. Как можно работать с несколькими списками одновременно в языке Python?
5. Для чего служит ключевое слово `local` в языке Python?