

СОЗДАНИЕ ВЫВОДА СКРИПТА PYTHON

Атаева Г.И.¹, Минич Л.С.² Email: Ataeva6104@scientifictext.ru

¹Атаева Гульсина Исроиловна – преподаватель,
кафедра информационных технологий,
Бухарский государственный университет;

²Минич Людмила Станиславовна – учитель информатики,
Специализированная государственная школа № 36,
г. Бухара, Республика Узбекистан

Аннотация: понимание того, как создавать выходные данные из сценария Python, является хорошей отправной точкой для тех, кто плохо знаком с языком программирования Python. Вы мгновенно получаете обратную связь о своих операторах Python от интерактивного интерпретатора Python и можете экспериментировать с правильным синтаксисом. В статье рассматривается функция печати. Функция печати представляет собой группу операторов Python, которые ставятся вместе как единое целое для выполнения конкретной задачи.

Ключевые слова: функция печати, аргумент, листинг, цитата, кавычки.

CREATING A PYTHON SCRIPT OUTPUT

Ataeva G.I.¹, Minich L.S.²

¹Ataeva Gulsina Isroilovna – Lecturer,
DEPARTMENT OF INFORMATION TECHNOLOGIES,
BUKHARA STATE UNIVERSITY;

²Minich Lyudmila Stanislavovna - Teacher of Informatics,
SPECIALIZED STATE SCHOOL № 36,
BUKHARA, REPUBLIC OF UZBEKISTAN

Abstract: understanding how to generate output from a Python script is a good starting point for anyone new to the Python programming language. You get instant feedback on your Python statements from the interactive Python interpreter and can experiment with the correct syntax. The article discusses the print function. A print function is a group of Python statements that are put together as a unit to accomplish a specific task.

Keywords: print function, argument, listing, quote, quotes.

УДК 004.43

Изучение функции печати. Функция печати представляет собой группу операторов Python, которые ставятся вместе как единое целое для выполнения конкретной задачи. Вы просто вводите один оператор Python и он выполняет задачу за вас.

Кстати: «новая» функция печати. В Python 2 печать не является функцией. Это стало функцией, когда был создан Python 3. Это важно знать, если вам когда-либо будет поручено преобразовать скрипт из версии 2 в версию 3.

Задача функции печати - выводить элементы. Элементы для вывода стоит правильно называть *аргументом*.

Базовый синтаксис функции печати следующий:
print (аргумент).

Стандартная библиотека функций. Функция печати называется *встроенной* функцией, потому что она является частью стандартной библиотеки функций Python. Чтобы использовать эту функцию, ничего особенного делать не нужно. Она предоставляется для использования при установке Python.

Аргумент – это часть печати функции, ими могут быть символы, такие как ABC или 123. Это также могут быть значения, выраженные в переменных.

Использование символов в качестве аргументов функции печати.

Для отображения символов (также называемых *строковыми литералами*) с помощью функции печати необходимо заключить символы в одинарные или двойные кавычки. В листинге 1 показано использование пары одинарных кавычек для заключения символов (предложения), чтобы его можно было использовать в качестве аргумента функции печати.

Листинг 1. Использование пары одиночных кавычек для заключения символов

```
>>>
print ("Это пример использования одинарных кавычек.")
Это пример использования одинарных кавычек.
>>>
```

В листинге 2 показано использование двойных кавычек с функцией печати. Вы можете видеть, что результирующий вывод в Листинге 1 и листинге 2 не содержит кавычек, а содержит только символы.

Листинг 2. Использование пары двойных кавычек для заключения символов.

```
>>>
print ("Это пример использования двойных кавычек.")
Это пример использования двойных кавычек.
>>>
```

Лучшие выберите один тип цитат и придерживайтесь его. Если вы хотите использовать одинарные кавычки для заключения строковых литералов в аргумент функции печати, используйте их последовательно. Если вы предпочитаете двойные кавычки, последовательно используйте их. Хотя Python это не волнует, использование одинарных кавычек в одном аргументе функции печати, а затем двойных кавычек в следующем считается плохим тоном. Перемешивание кавычек взад и вперед затрудняет чтение кода людьми.

Иногда вам нужно вывести символьную строку, содержащую одинарную кавычку, чтобы показать владение или сокращение. В таком случае вы должны заключить аргумент функции печати в двойные кавычки, как показано в листинге 3.

Листинг 3. Защита одиночной кавычки с помощью двойных кавычек.

```
>>>
print ("This example protects the output's single quote.")
This example protects the output's single quote.
>>>
```

В других случаях вам необходимо вывести строку символов, содержащих двойные кавычки, например, для цитаты. В листинге 4 показан пример защиты цитаты с использованием одинарных кавычек в аргументе.

Листинг 4. Защита двойных кавычек с помощью одинарных кавычек.

```
>>>
print ('Я сказал: «Мне нужно защитить свою цитату!» и сделал это.')
Я сказал: «Мне нужно защитить свою цитату!» так и сделал.
>>>
```

Форматирование вывода с помощью функции печати. Вы можете выполнять различные функции форматирования вывода с помощью функции печати. Например, вы можете вставить одну пустую строку, используя функцию печати без аргументов, например:

```
print()
```

Ниже приведён короткий сценарий Python, который вставляет пустую строку между двумя другими строками вывода.

```
>>>
print ("This is the first line.")
```

```
print ()
print ("This is the first line after a blank line.")
>>>
```

Другой способ форматирования вывода с помощью функции печати - тройные кавычки. Тройные кавычки - это просто три набора двойных кавычек (" " ").

В листинге 5 показано, как использовать тройные кавычки для вставки символа перевода строки (нажатием клавиши Enter). Когда вывод отображается, каждый внедренный символ перевода строки заставляет следующее предложение появляться на следующей строке. Таким образом, перевод строки перемещает ваш вывод на следующую новую строку. Обратите внимание, что вы не можете увидеть символ перевода строки, встроенный в каждую строку кода - вы можете увидеть только его эффект в выводе.

Листинг 5. Использование тройных кавычек.

```
>>>
print ("\" \"Это первая строка.
... Это вторая строка.
... Это третья строка.\" \"")
Это первая строка.
Это вторая строка.
Это третья строка.
>>>
```

Вы можете многое сделать с помощью функции печати для отображения и форматирования вывода! Фактически, вы можете потратить весь этот час, просто играя с форматированием вывода. Однако есть дополнительные важные основы Python, которые вам необходимо изучить, например, сценарии форматирования для удобства чтения.

Список литературы / References

1. Хаятов Х.У., Атаева Г.И., Хайдаров О.Р. Функции и элементы OpenGL, используемые для построения основных форм в C# // Universum: технические науки: электрон. научн. журн., 2020. 11 (80). [Электронный ресурс]. Режим доступа: <https://7universum.com/ru/tech/archive/item/10915/> (дата обращения: 19.01.2021).
2. Исмоилова М.Н., Имомова Ш.М. Интерполяция функции // Вестник науки и образования, 2020. № 3(81). Часть 3. С. 5.
3. Зарипова Г., Рузиева К. Использование интерактивных методов в процессе обучения студентов компьютерным технологиям// Проблемы педагогики, 2018. № 7 (39). С. 5-8.
4. Акабировва Л.Х., Атаева Г.И. Особенности уроков с применением информационных технологий // Проблемы педагогики. № 2 (47), 2020. С. 42-44.
5. Хаятов Хуршиджон Усманович. Методическая система эвристического обучения информатике в высшем образовании // Academy, 2020. № 7 (58).
6. Fayzieva D.Kh. USING SOFTWARE FOR TEACHING FOREIGN LANGUAGES. [Электронный ресурс]. Academy. № 9 (60), 2020. Режим доступа: <http://academicjournal.ru/> (дата обращения: 19.01.2021).
7. Мурадова Г.Б., Эргашева Э.Б. Аддиктивное поведение в виртуальном мире // Academy. № 9 (60), 2020. С. 40.