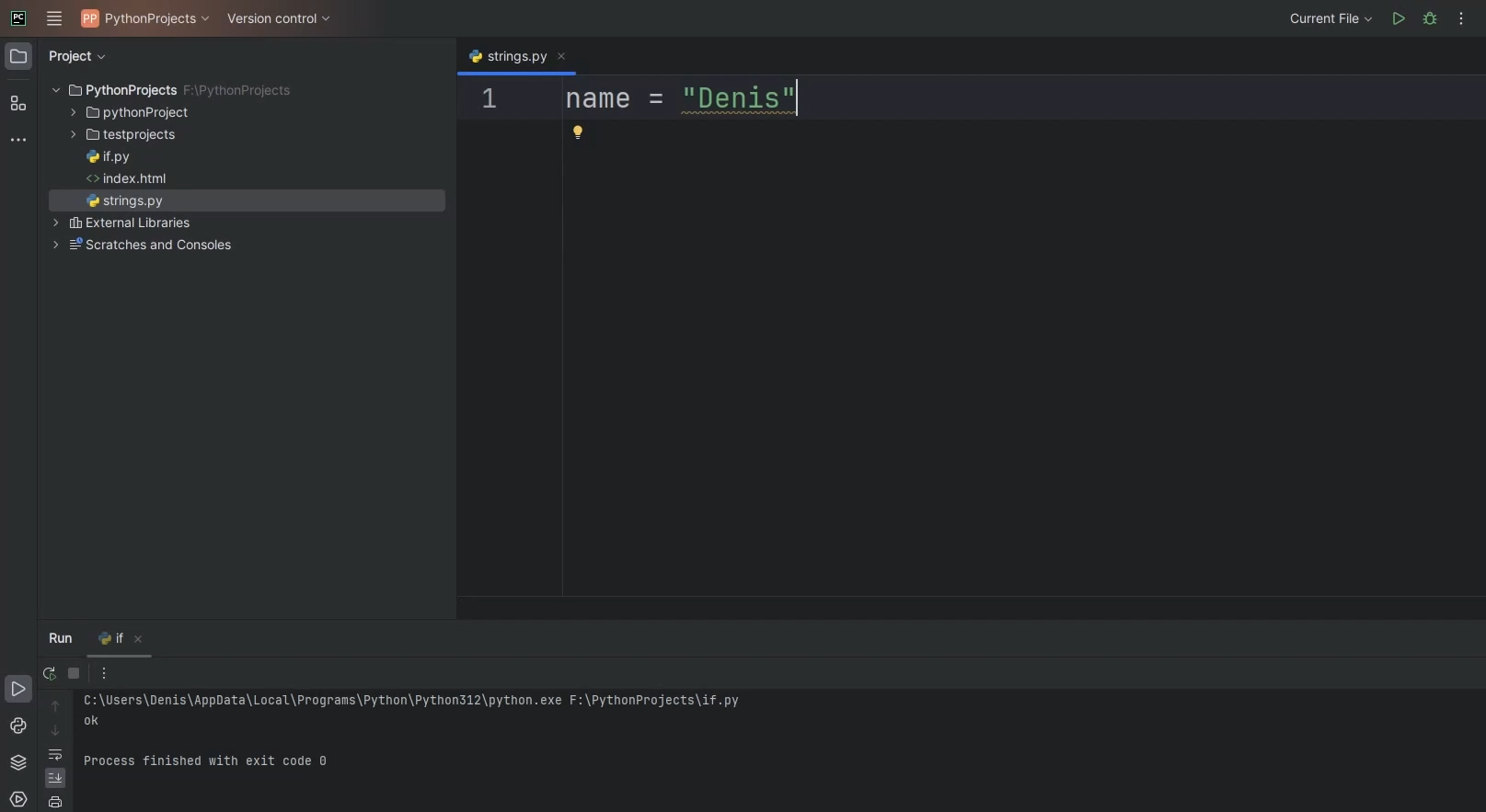
**2023/09/20 00:00|Лекция. Строки и индексация строк**

**Строки и индексация строк**

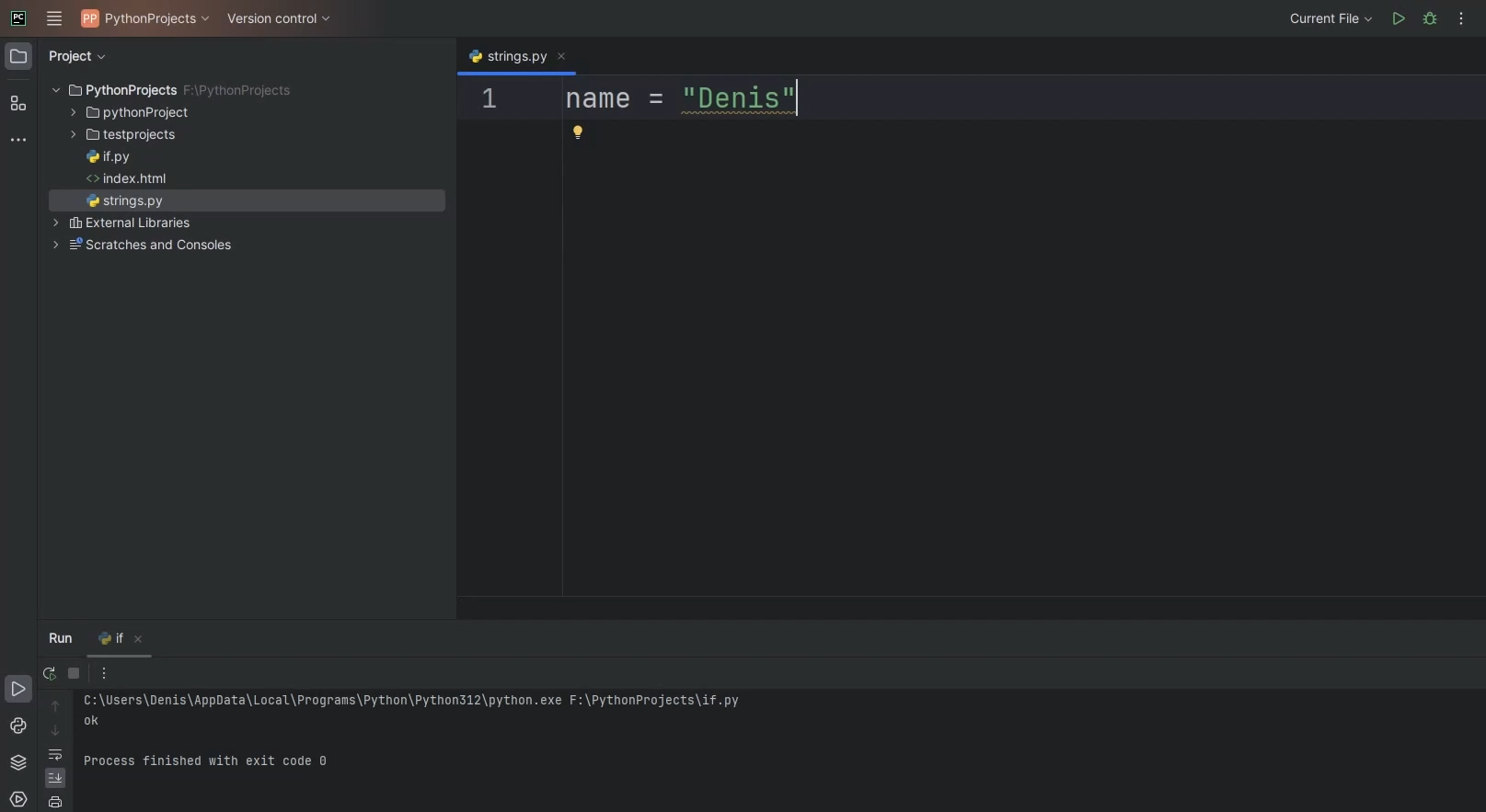
Сегодня мы подробнее поговорим о строках и о том, что они из себя представляют. Давайте начнем!

**Что такое строки в Python?**

Строки очень удобный тип данных. Вся информация, поступающая на вход от пользователя, или информация с полей для ввода, например на сайтах, будет являться строкой.  
  
**Строка – это последовательность символов кодировки Unicode**. Строки всегда заключаются в кавычки. В Python мы можем использовать, как одинарные, так и двойные кавычки.  
  
Например, у нас есть переменная «name», внутри которой находится строка «Denis», как мы видим, оба варианта кавычек допустимы (Рис.1, 2).



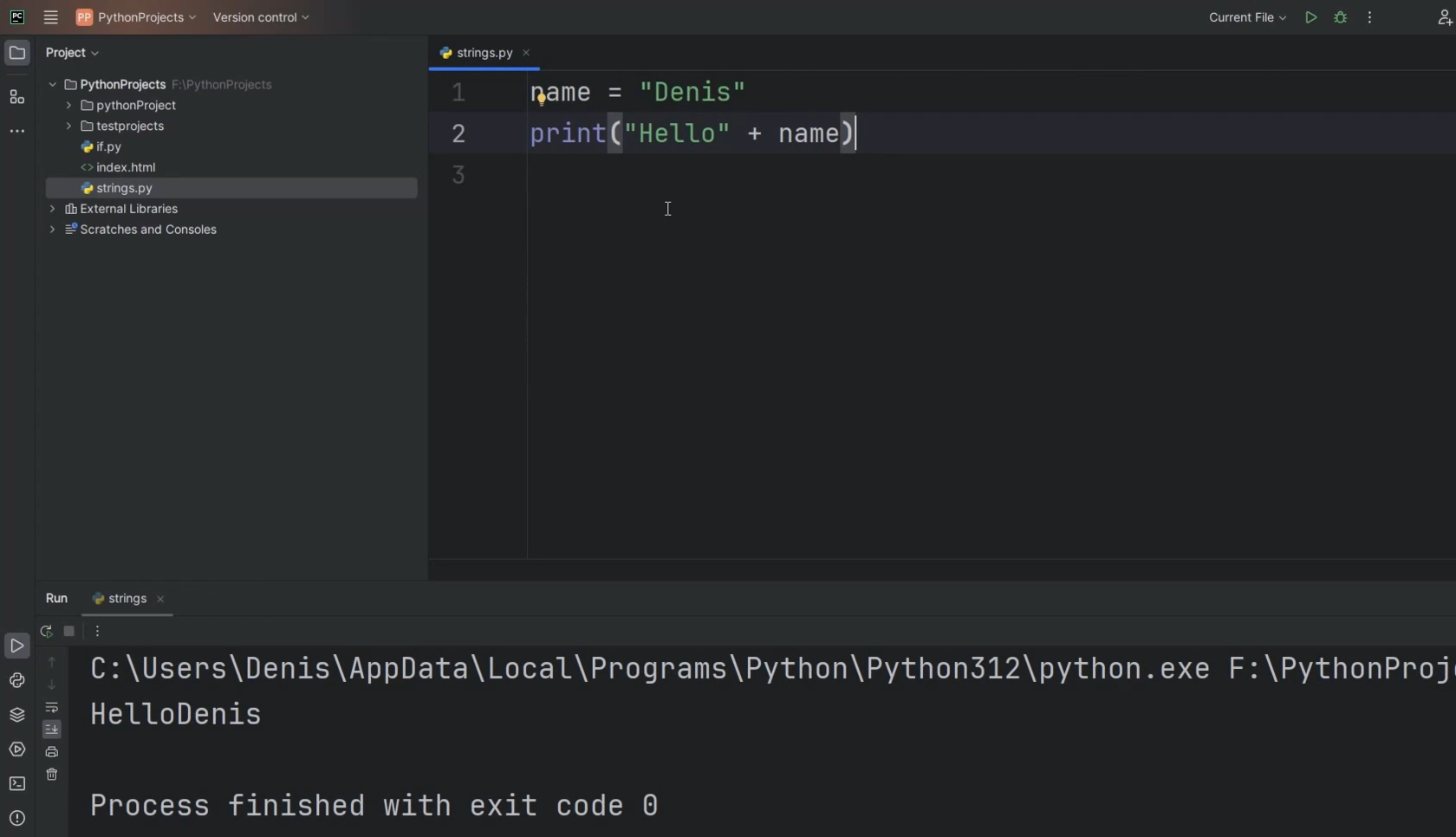
(Рис.1)



(Рис.2)

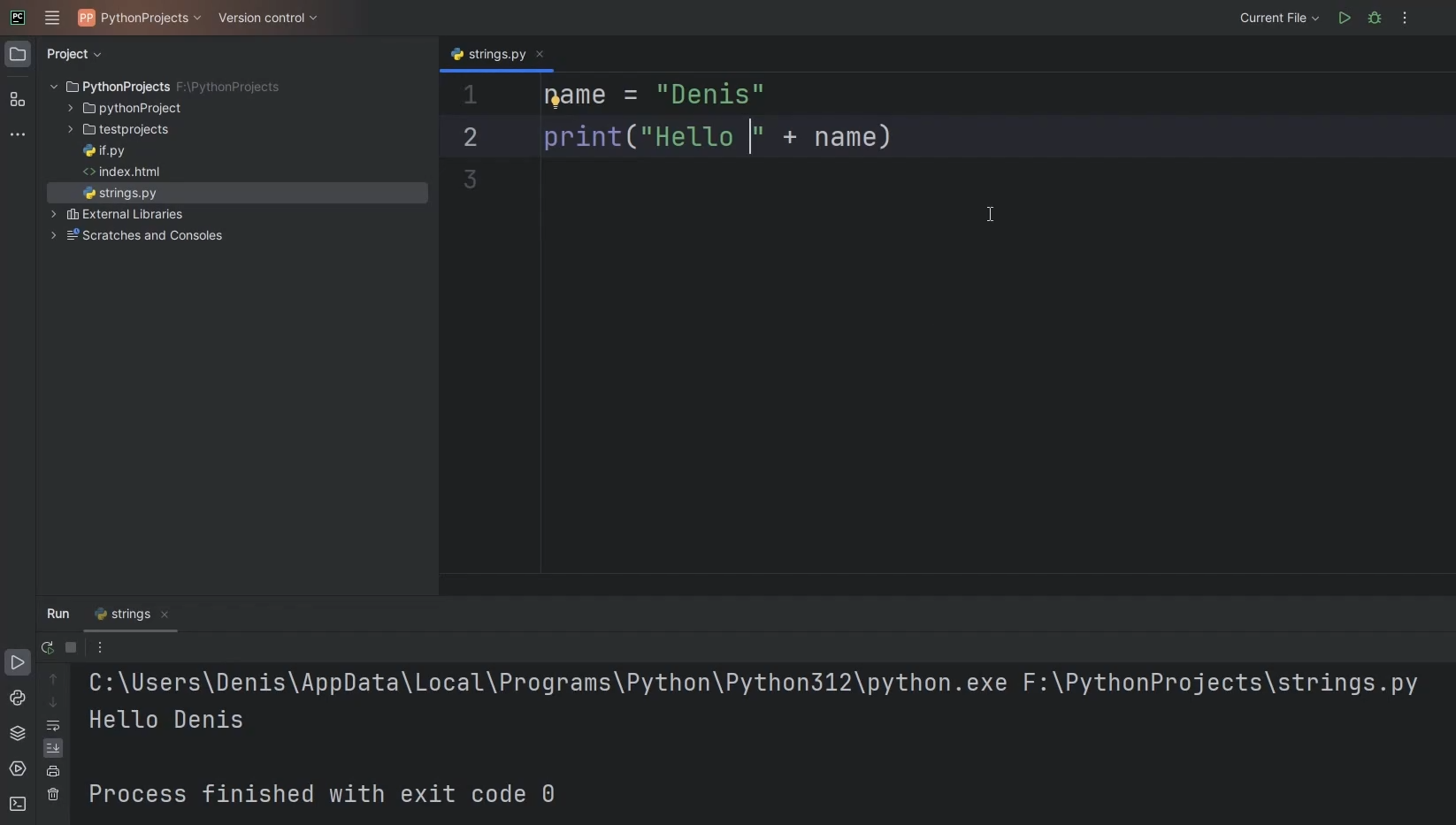
**Операции со строками**

Помимо их создания, мы конечно же можем совершать со строками определенные действия. Одна из поддерживаемых строками операций – это *конкатенация*, или же сложение. Допустим, у нас есть все та же переменная «name» со значением «Denis». Сложим нашу переменную со словом «Hello» и выведем на экран (Рис.3). Получим новую строку «HelloDenis».



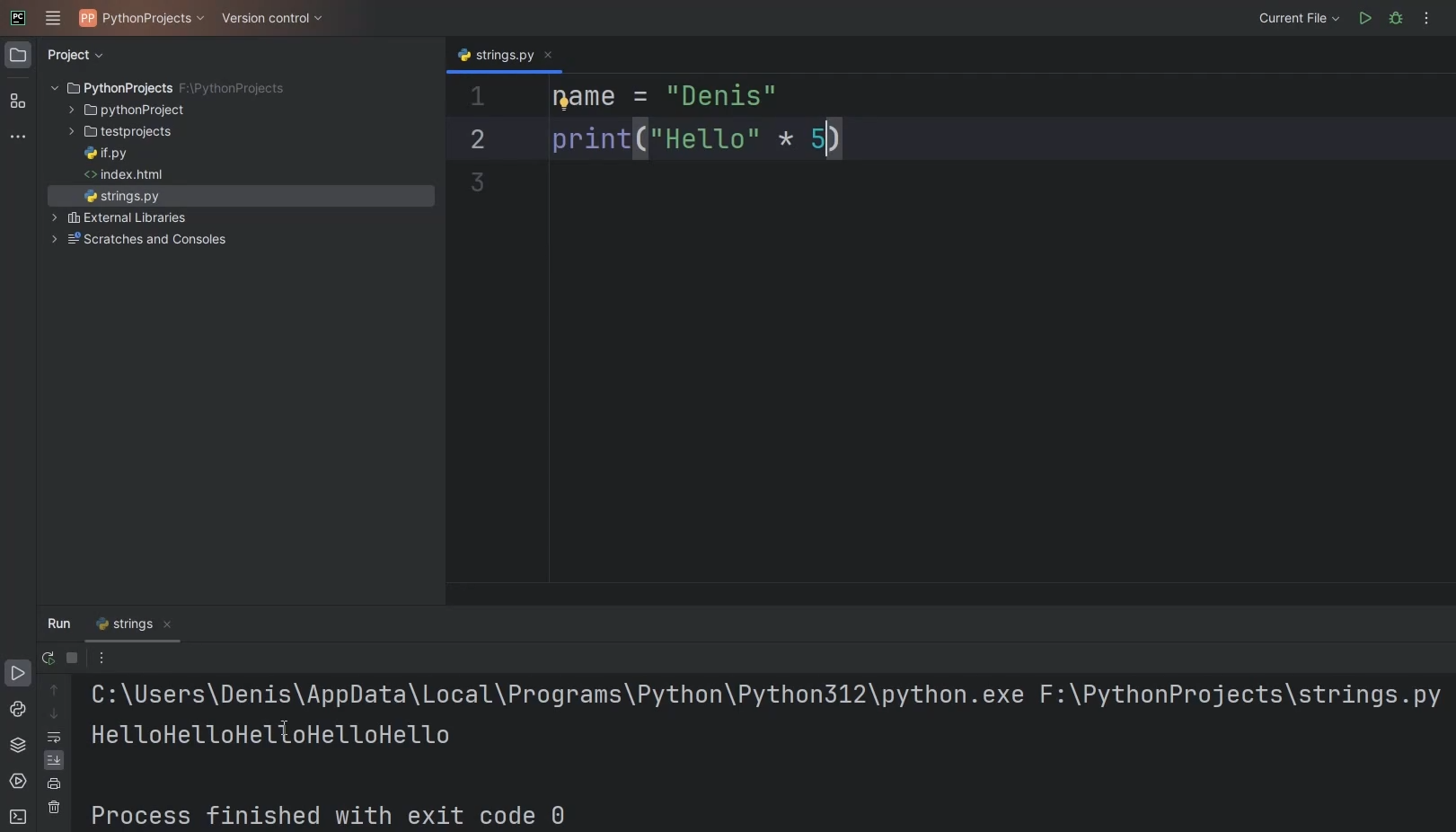
(Рис.3)

Чтобы выглядело приятнее глазу добавим пробел после «Hello» и получим уже строку «Hello Denis» (Рис.4). Компьютер воспринимает пробел как символ, поэтому он занимает такое же место внутри строки.



(Рис.4)

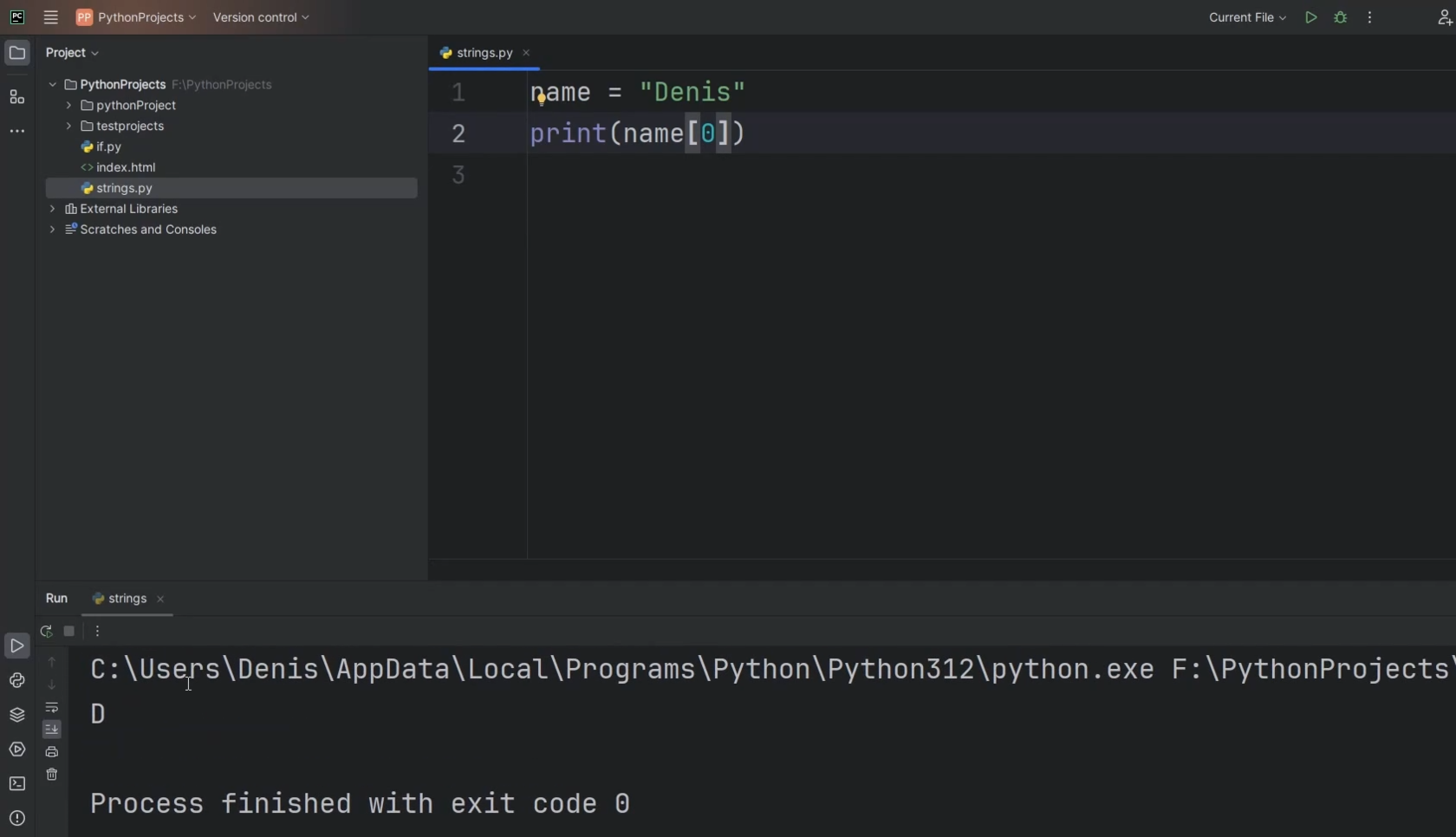
Помимо этого, мы можем дублировать строки с помощью умножения. Например, умножим строку «Hello» на 5 (Рис.5).



(Рис.5)

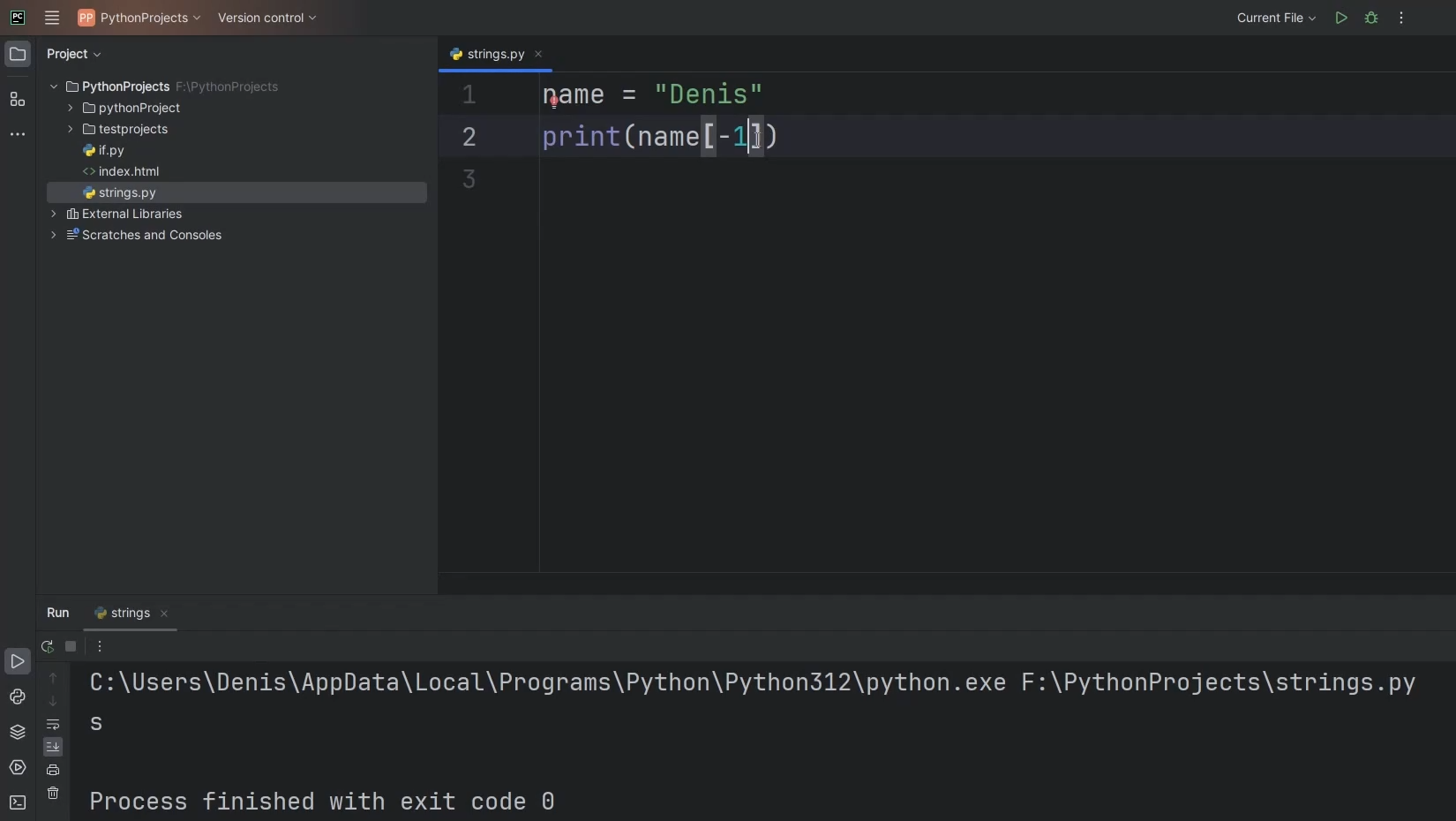
**Индексация строк**

Строки хороши тем, что представляют собой набор символов и мы можем пройтись по каждому из них. Здесь нам и приходит на помощь индексация. **Индексация подразумевает наличие порядкового номера у каждого элемента строки**. Мы можем использовать ее для вывода частей или отдельного символа строки. Счет в строке начинается с нуля, соответственно первый символ строки будет иметь индекс [0].  
  
Рассмотрим, как это работает на примере. Выведем самый первый элемент нашей строки, для этого пишем нашу переменную и в квадратных скобках записываем порядковый номер нужного нам элемента, в нашем случае [0] (Рис.6).



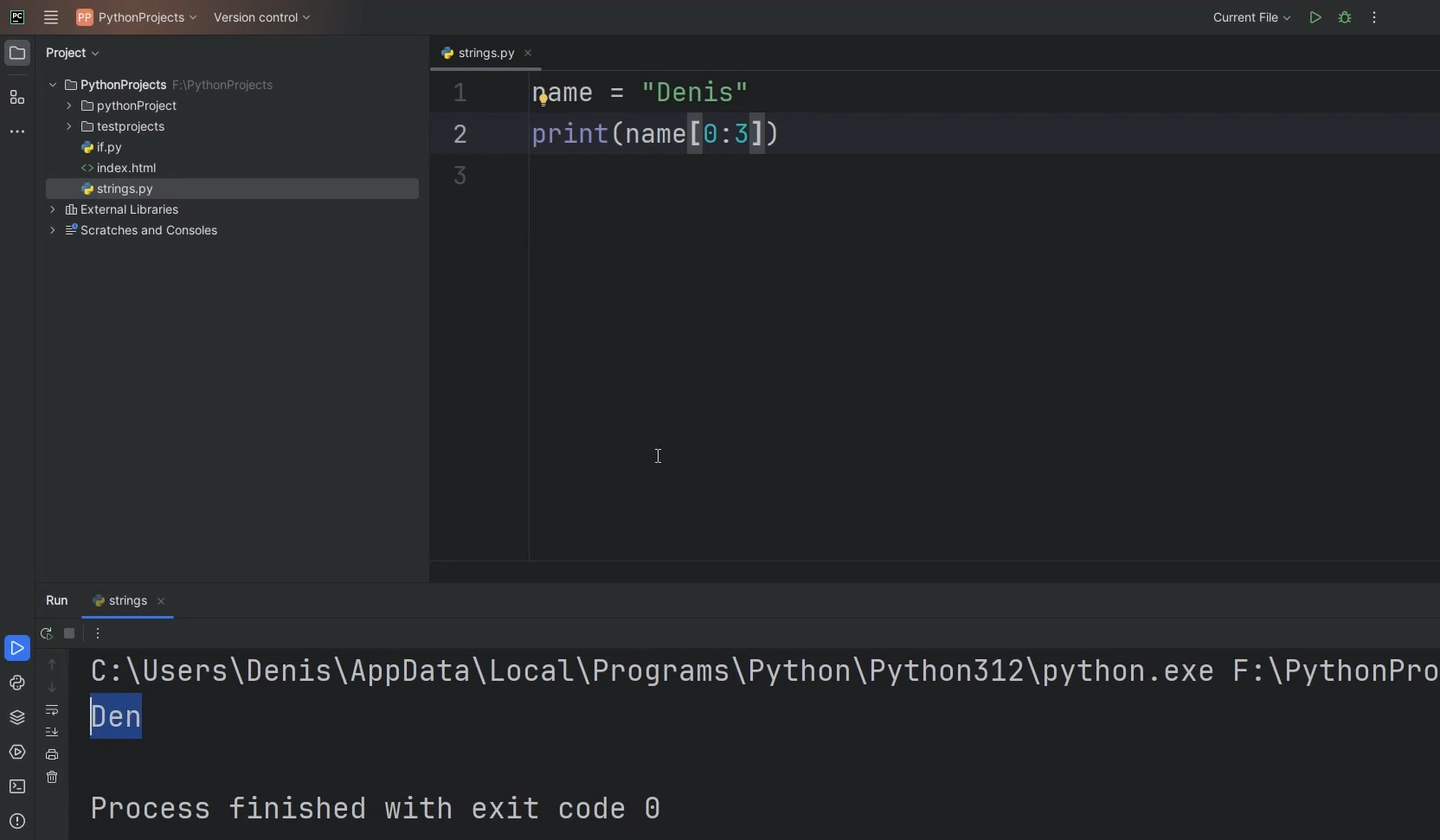
(Рис.6)

Точно так же можем вывести последний символ. Индексация поддерживает и отрицательные значения. При записи отрицательного индекса счет будет начинаться с конца. Например, нам нужен последний символ, поэтому пишем «print(name[-1])» и получаем в ответе «s» (Рис.7).



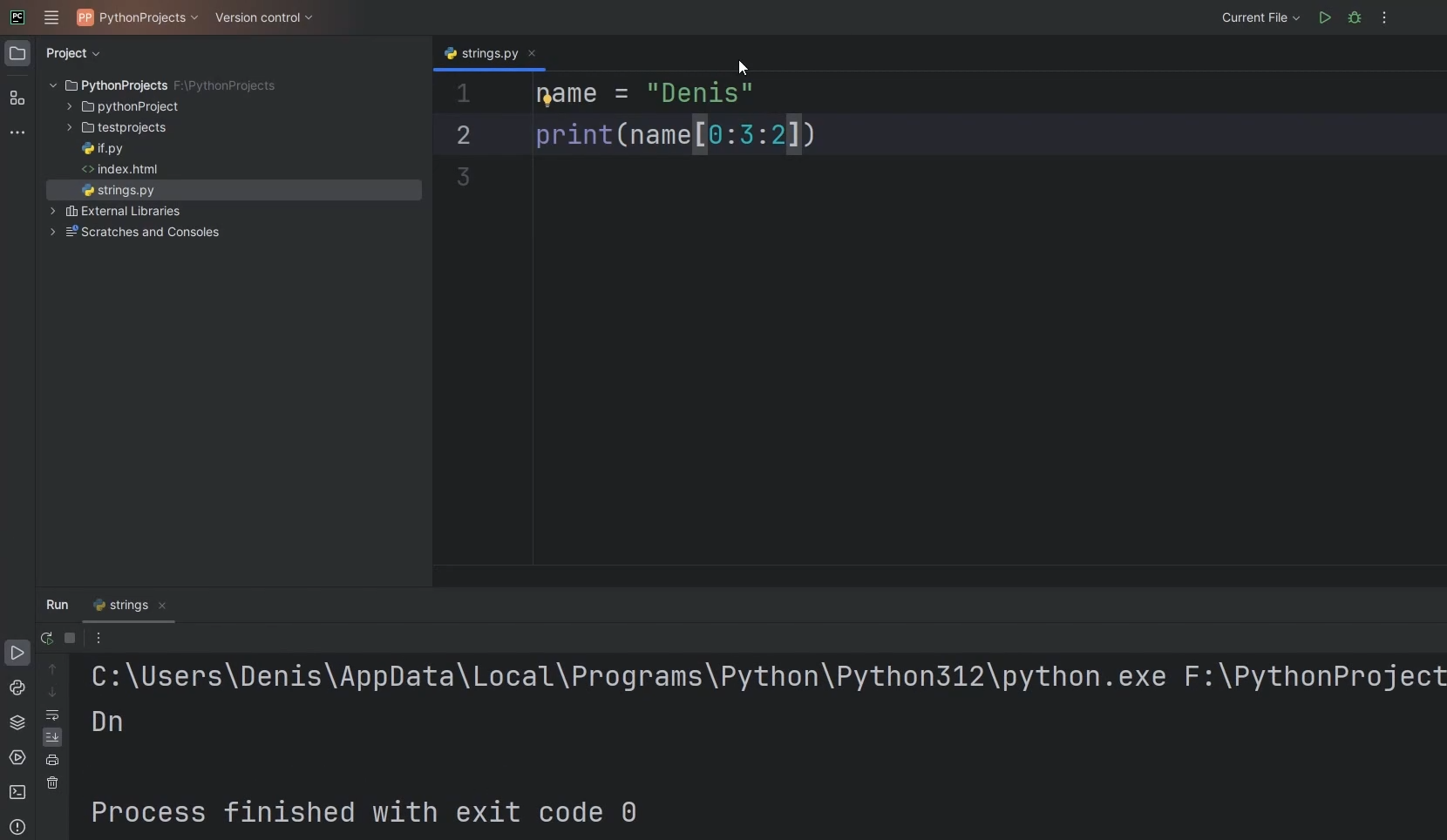
(Рис.7)

Мы можем выводить не только один символ, но и некую последовательность символов, своего рода подстроку. В квадратных скобках запишем [0:3] (Рис.8), с помощью двоеточия мы обозначаем начало среза, то есть его первый элемент, и конец среза, однако обозначающий его элемент в расчет уже не берется. Соответственно в ответе мы видим первые три символа «Den» с индексами [0], [1], [2]. Как мы видим, символ с индексом [3] не отображается.



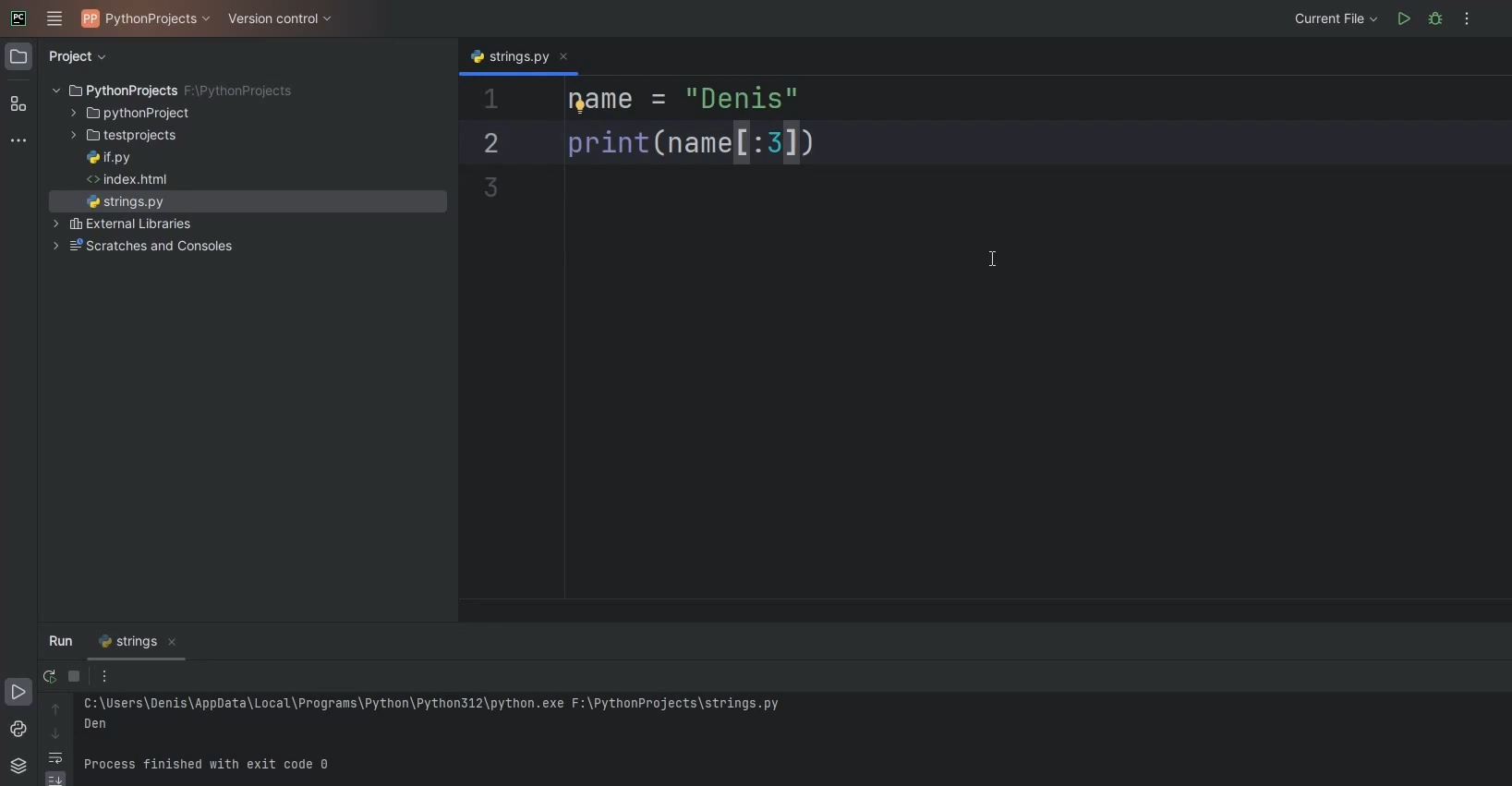
(Рис.8)

Кроме начала и конца среза мы можем указать еще и шаг. Для этого поставим еще одно двоеточие. Например, укажем, что нам нужны элементы с [0] по [3], но с шагом в 2 (Рис.9). Ответом в данном случае будет «Dn». Символ «e» был пропущен, т. к. был указан интервал 2.



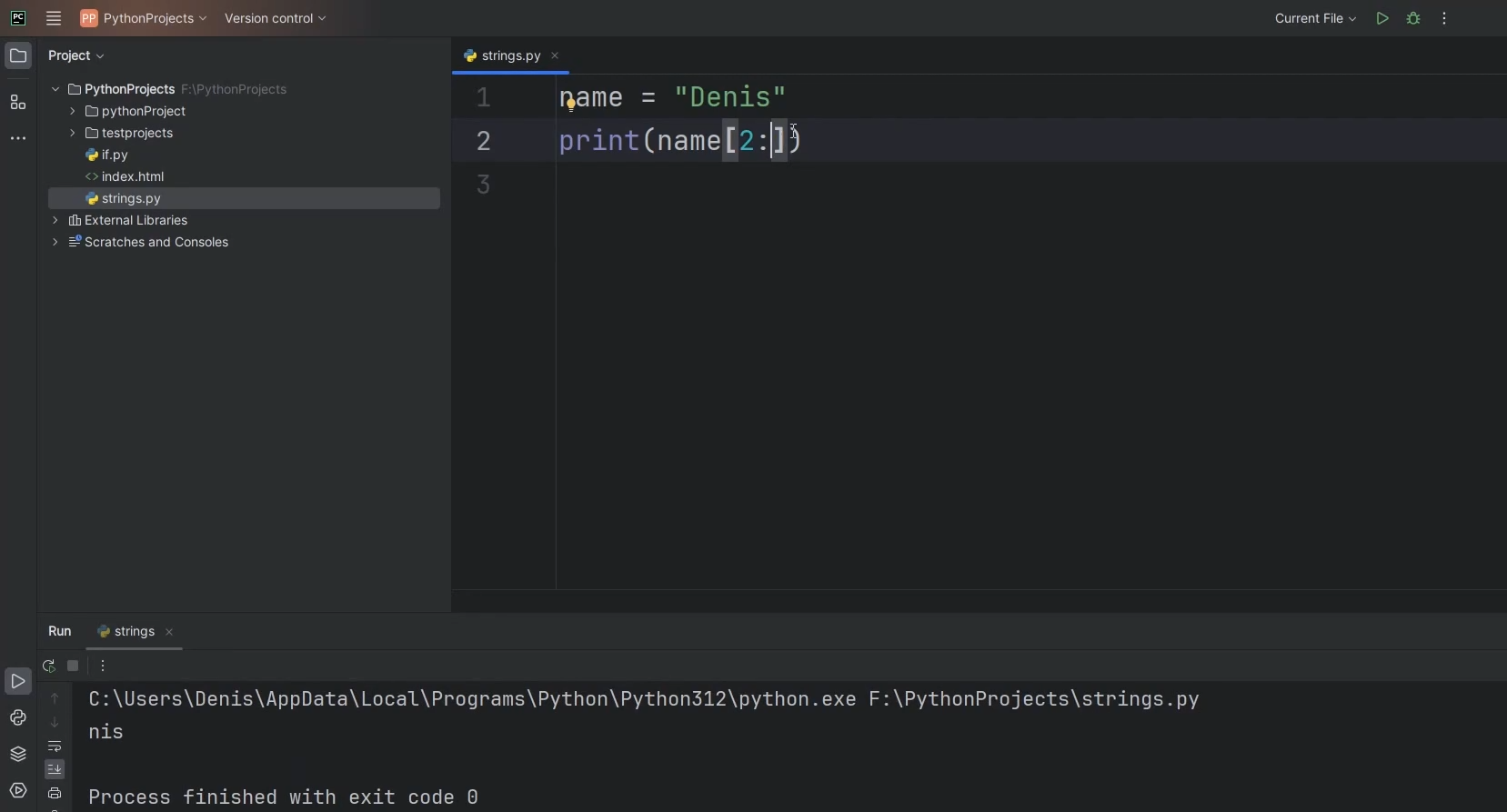
(Рис.9)

Итого, в квадратных скобках всего могут находиться три значения, однако мы можем также и пропускать начало и конец.  
  
Допусти мы напишем, что нам нужен срез [:3], в таком случае это будет равноценно срезу от 0 до 3, то есть отсчет автоматически начинается сначала (Рис.10).



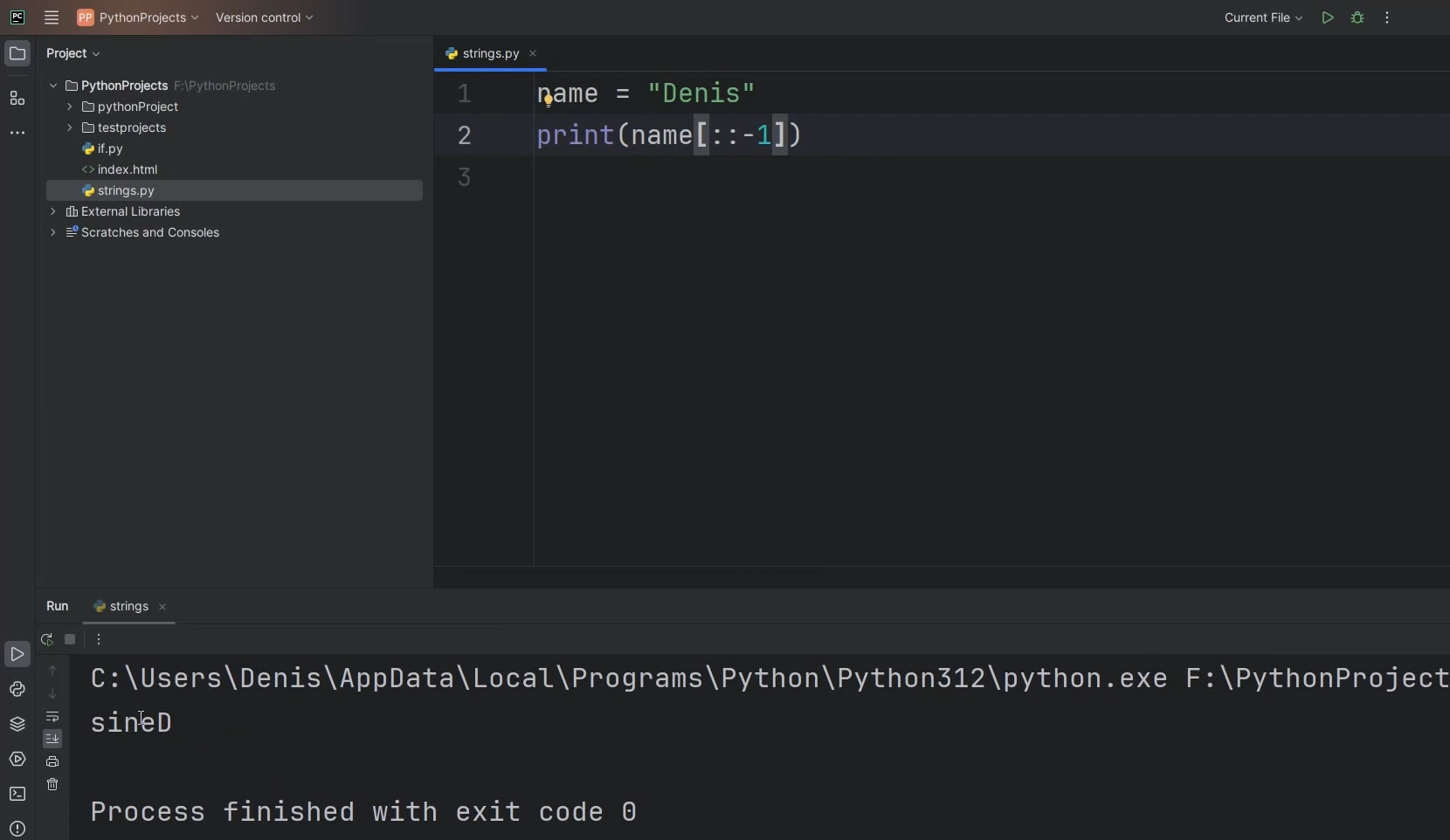
(Рис.10)

Или же напишем, что нам нужен срез [2:] (Рис.11), в таком случае мы не обозначаем конечную границу и срез будет происходить со второго элемента и до конечного, поэтому мы получаем ответ «nis».



(Рис.11)

По такому же принципу мы можем проставить два двоеточия и оставив пустыми начало и конец, указать только шаг, например [-1] (Рис.12). Нам будут выведены все символы, поскольку мы не указали границ, но уже в обратном порядке.



(Рис.12)

У строк существует обилие методов для работы, включая перевод символов в верхний или нижний регистр, поиск символов в строке и т. д., но о них мы поговорим уже позднее.

**Интересные ссылки**

<https://habr.com/ru/companies/ruvds/articles/500296/> - 41 вопрос о работе со строками в Python