**2023/10/05 00:00|Лекция. Способы вызова функции по умолчанию**

## Способы вызова функции по умолчанию

Сегодня мы рассмотрим какими способами мы можем вызвать функцию и какие ошибки мы можем при этом встретить.

Для начала создадим простую функцию, которая будет принимать три параметра, например «a», «b» и «c», и выводить их (Рис.1).

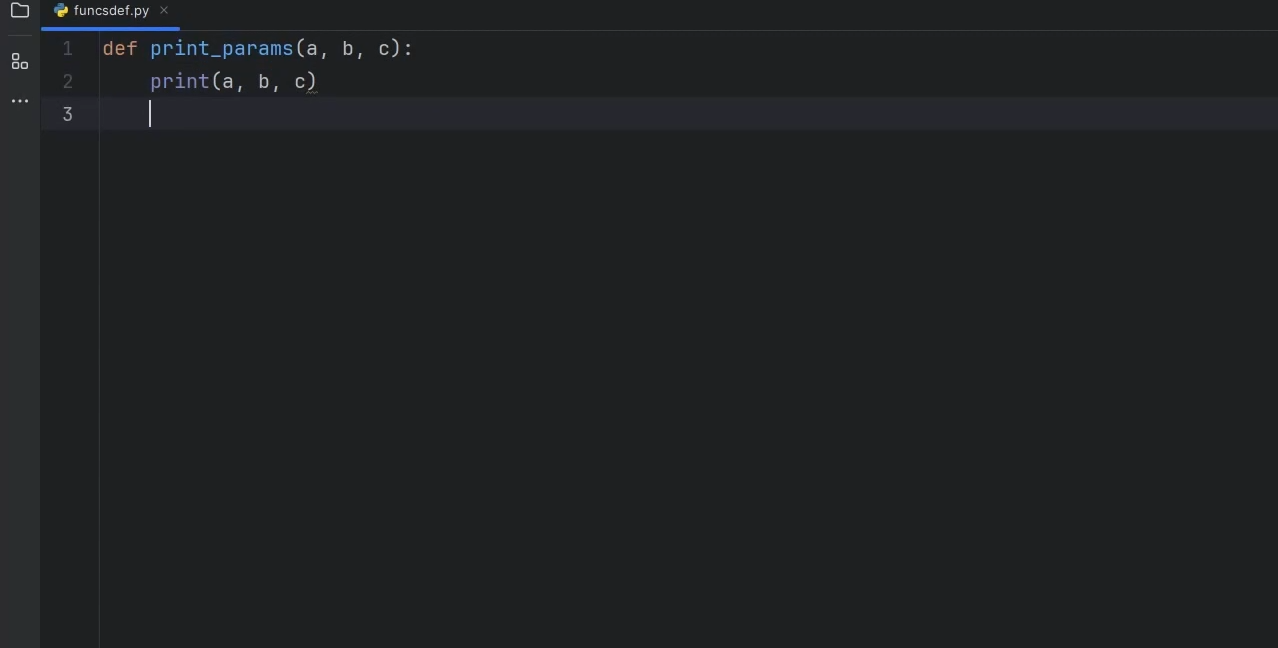


Рис.1  
  
Как мы помним, для вызова функции нам нужно написать ее имя, открыть и закрыть скобки.  
  
Наша функция является принимающей и принимает три параметра. Соответственно, если мы ее сейчас запустим, то увидим такую ошибку (Рис.2). Она сообщает, что у нас пропущено три обязательных позиционных аргумента «a», «b» и «c».

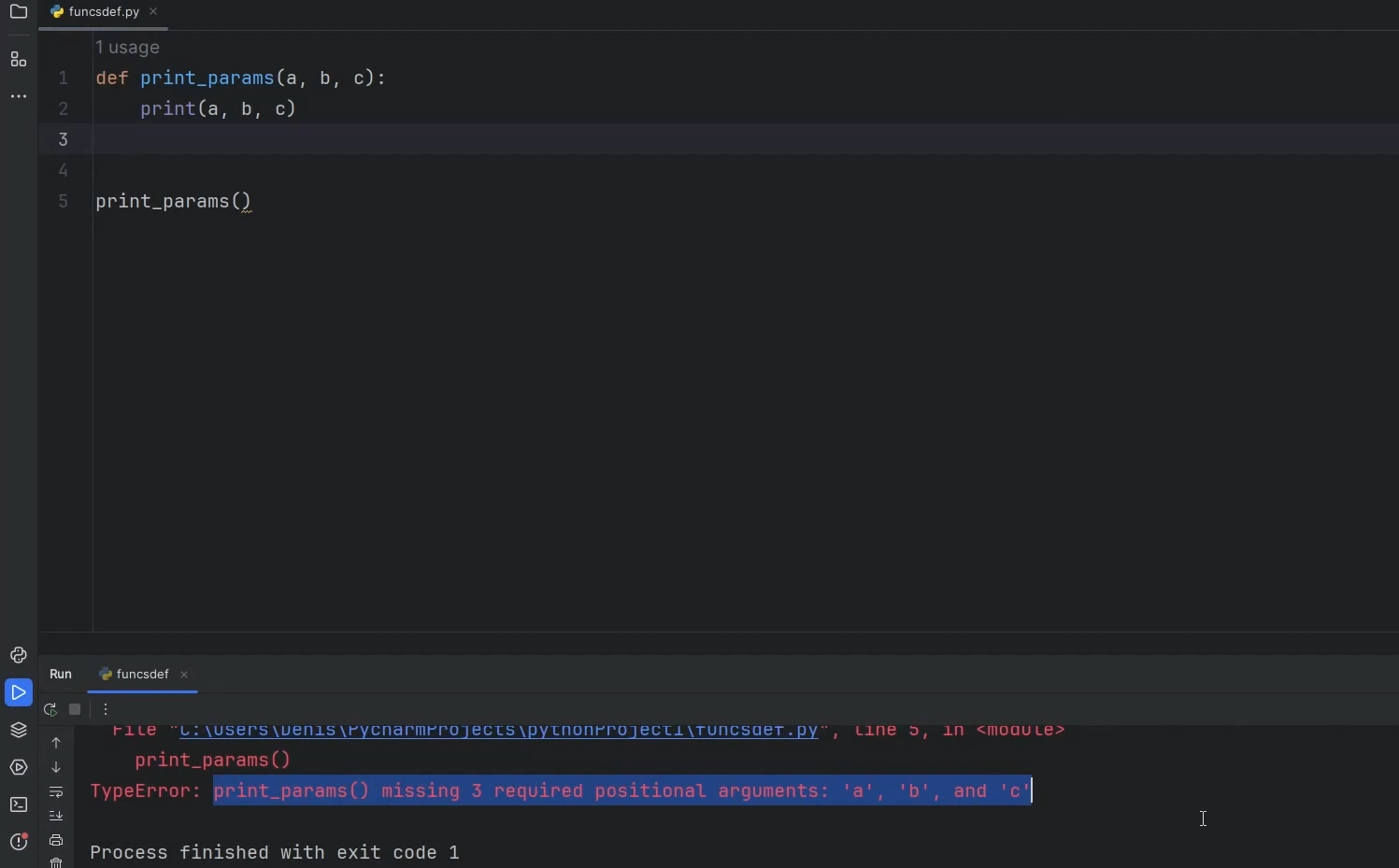


Рис.2  
  
То есть нам все же необходимо передать ей три каких-то значения, например 1, 2 и 3 (Рис.3). PyCharm здесь сам подсвечивает параметры, соответствующие введенным значениям.

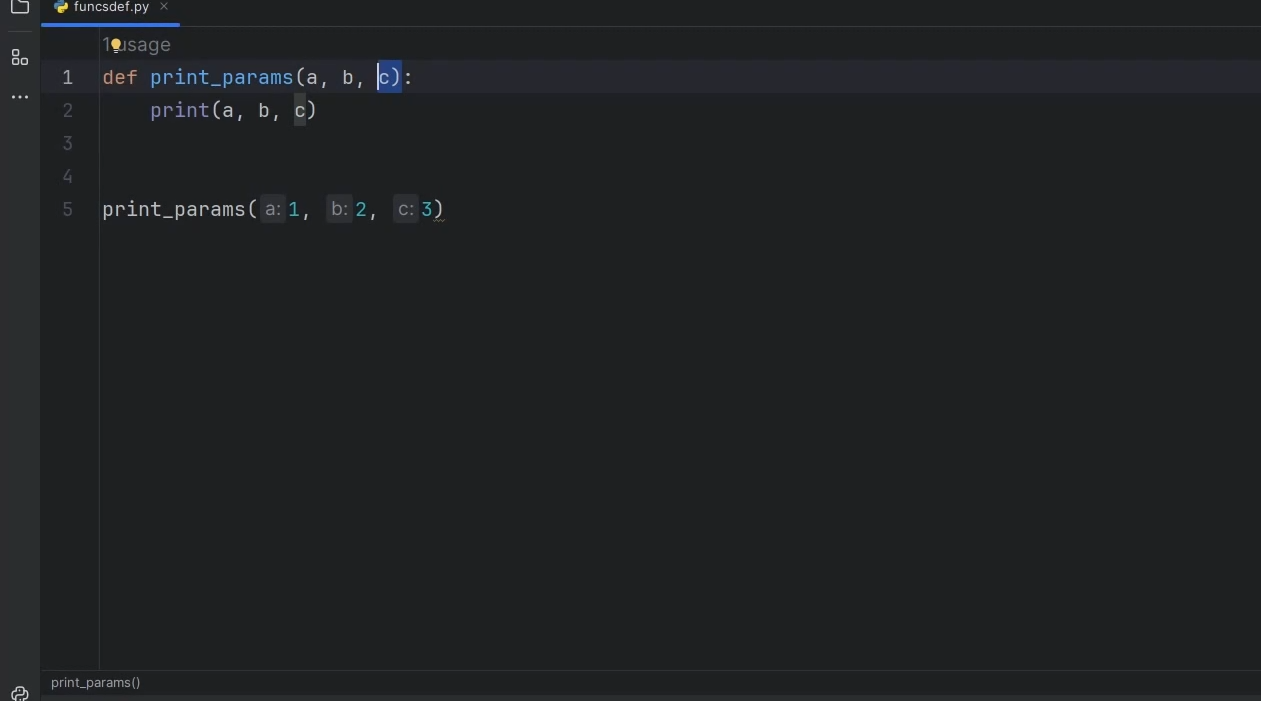


Рис.3  
  
Запустив программу, мы видим, что нам вывело наши значения (Рис.4).

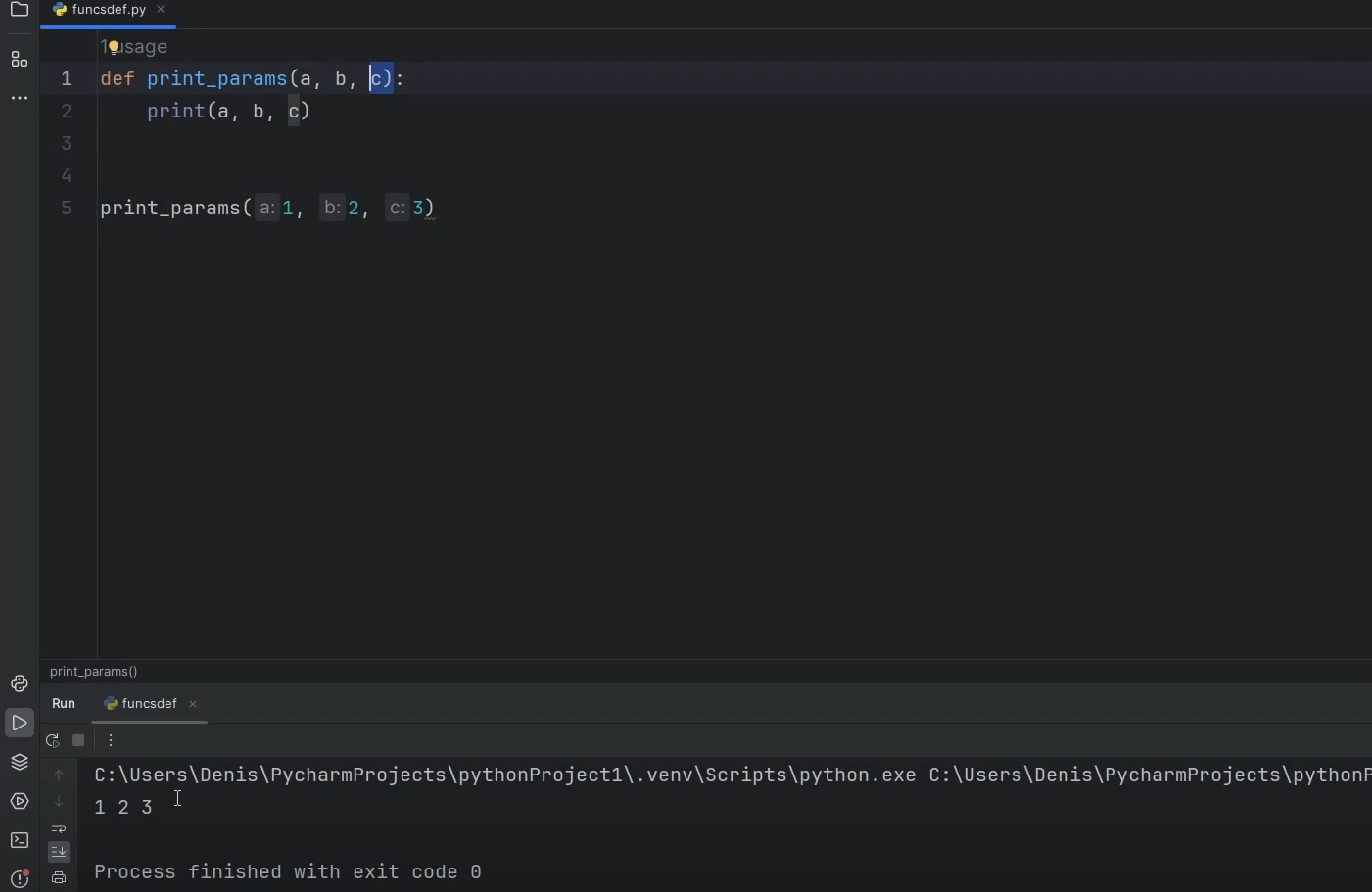


Рис.4  
  
При этом эти параметры могут быть вообще не связаны между собой. Они могут быть совершенно разных типов (Рис.5).

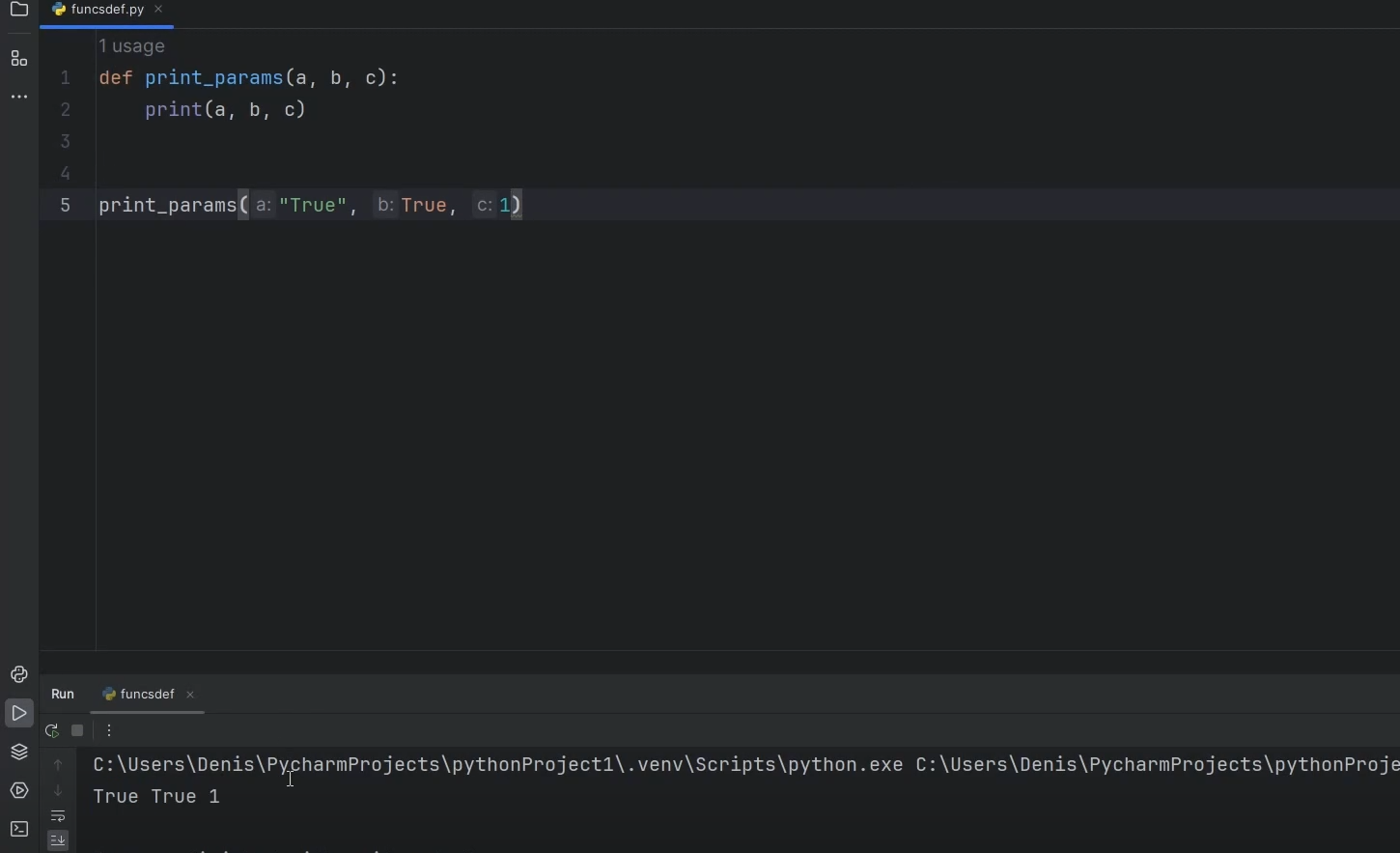


Рис.5  
  
Но важно заметить, что если в коде будет выстроено какое-то взаимодействие, например сложение «а» и «с», которые принадлежат разным типам, то мы получим ошибку (Рис.6).

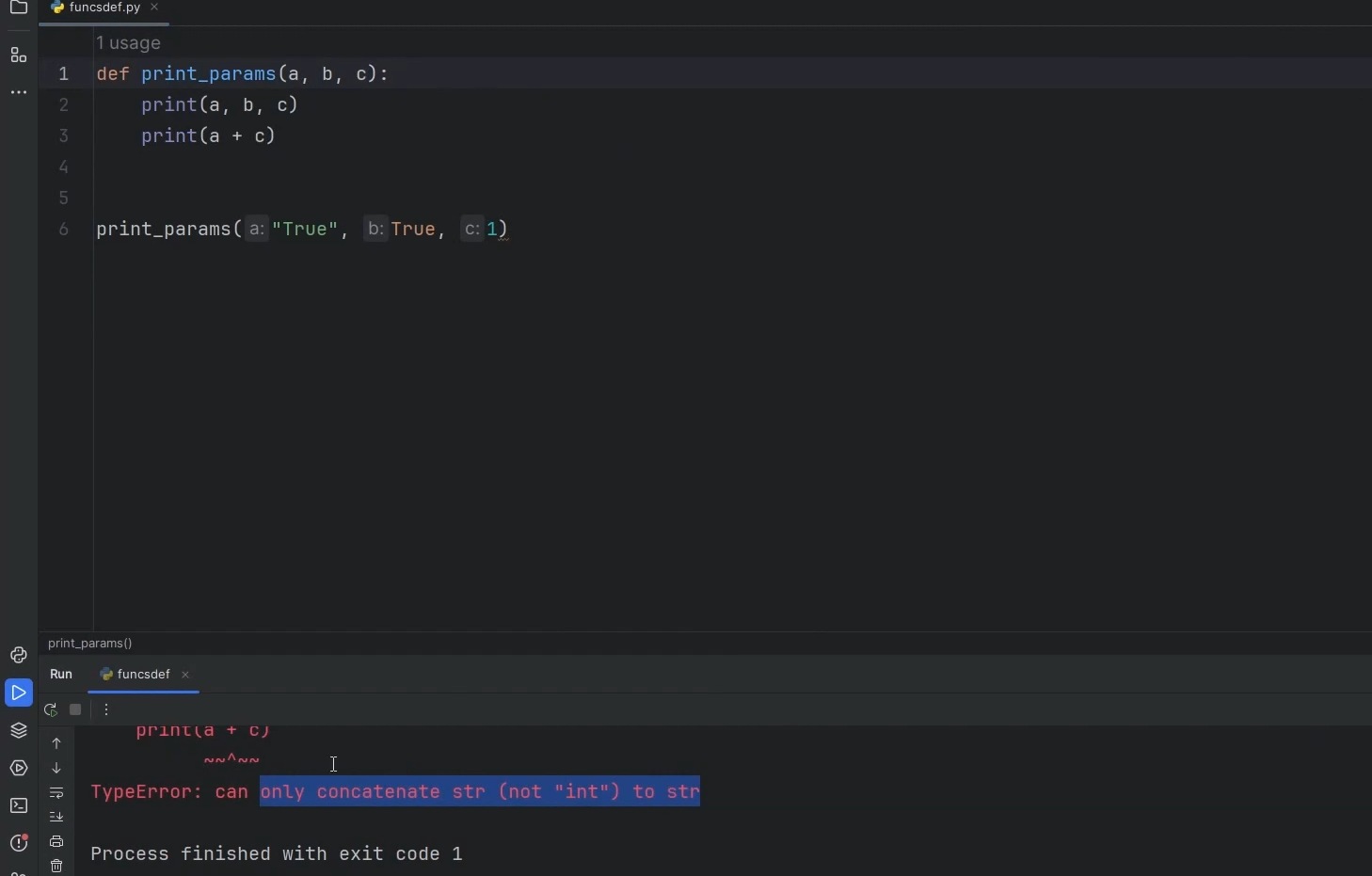


Рис.6  
  
Другое дело, если мы зададим параметру «а» значение 2 типа int, как и у параметра «с». Тогда мы уже получим их сумму – 3 (Рис.7).

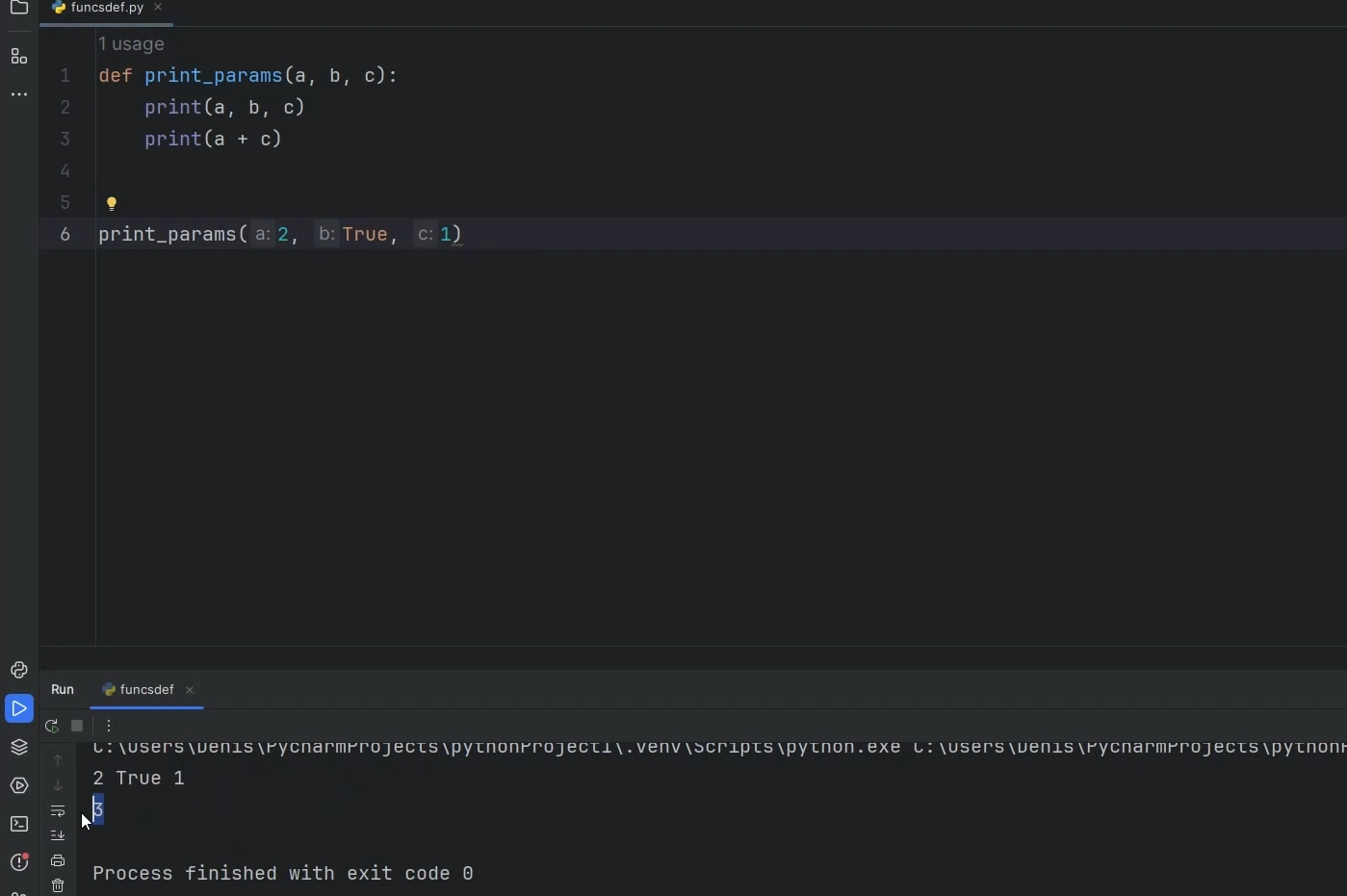


Рис.7  
  
Таким образом, если мы указываем, что функция принимает три параметра, мы должны передать ей три значения. Если передадим меньше, то получим ошибку (Рис.8) . И если передадим больше, тоже (Рис.9).

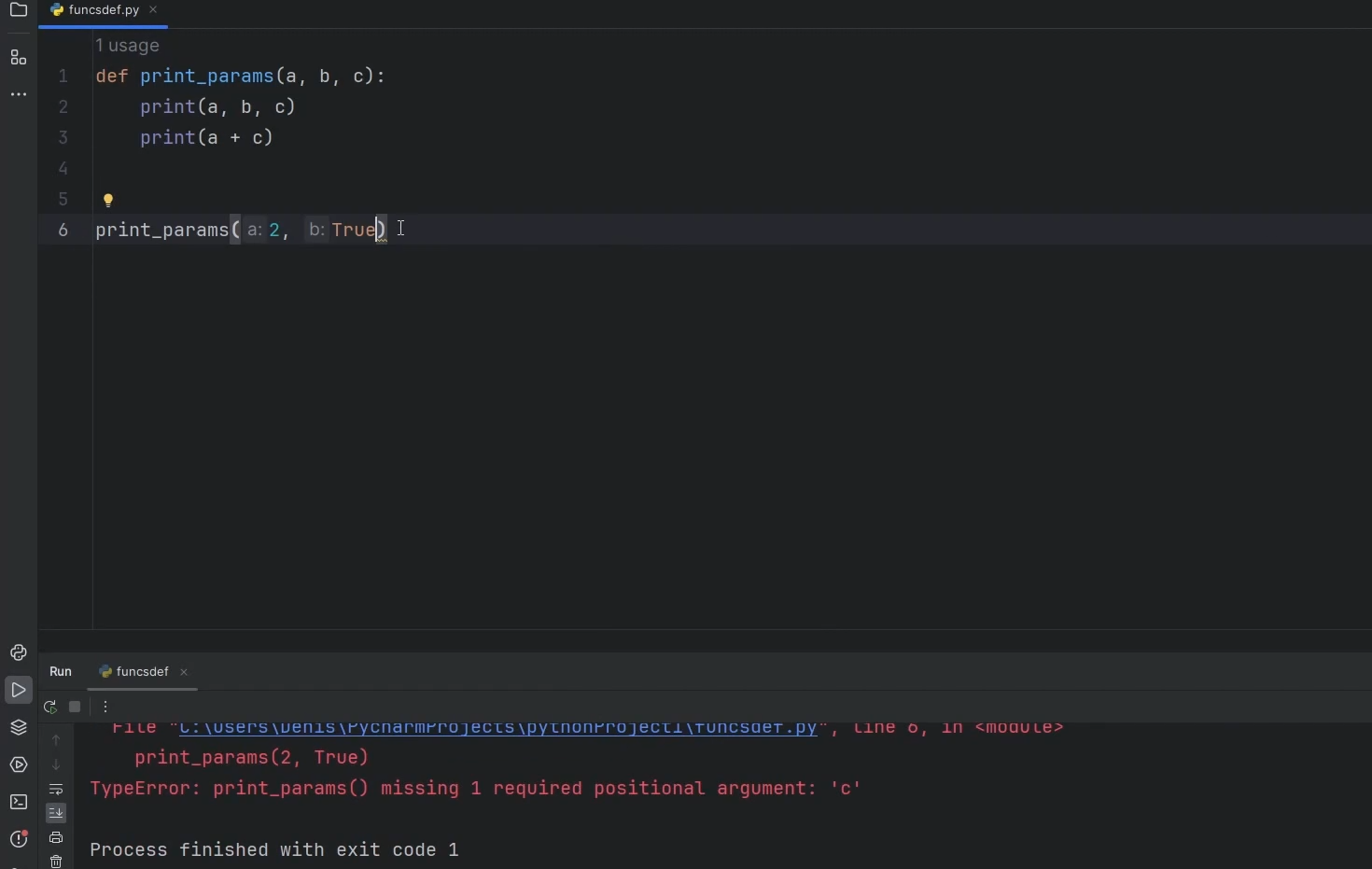


Рис.8

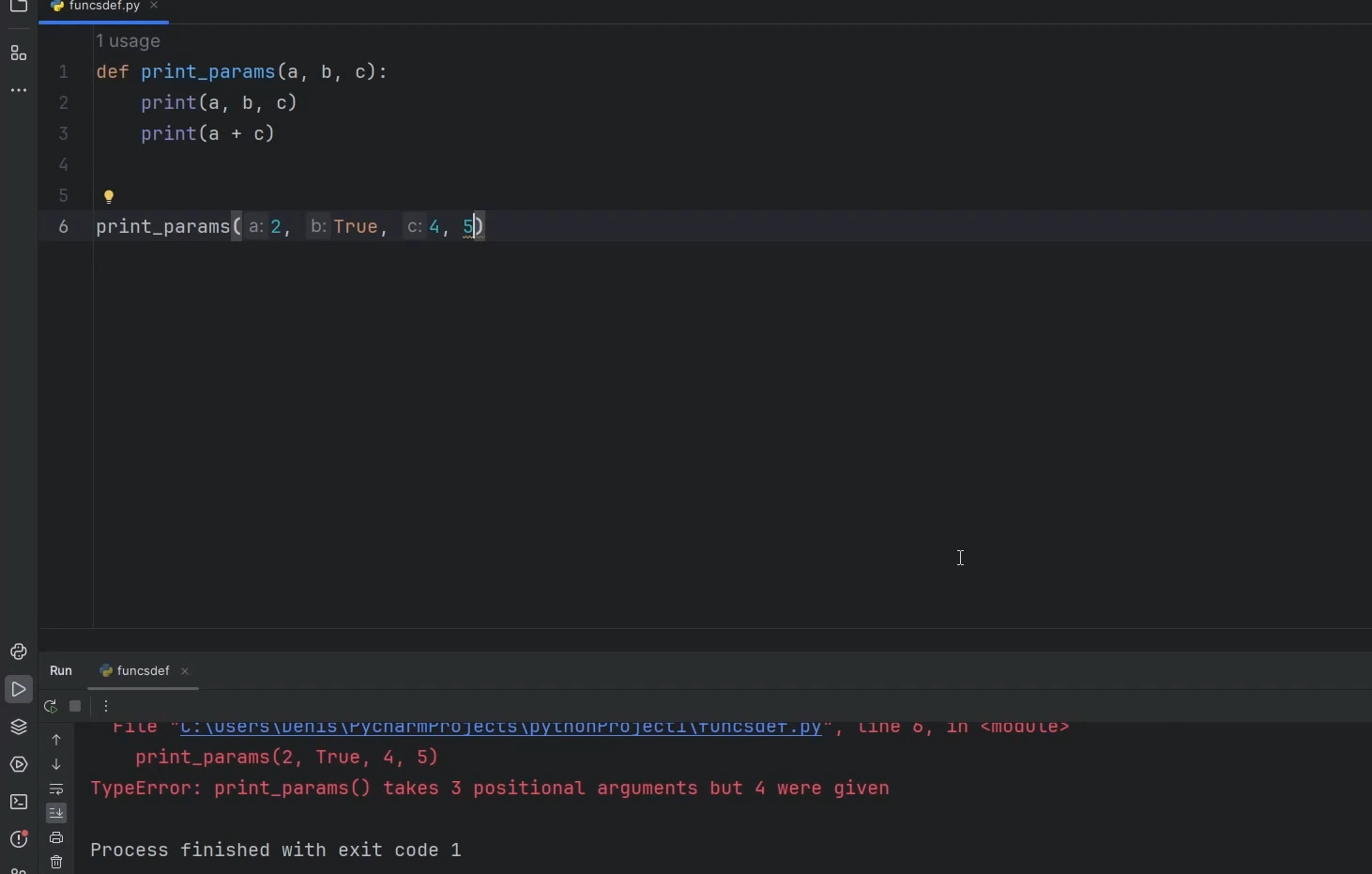


Рис.9  
  
Но есть ситуации, когда нам нет нужды передавать параметры. Когда они объявлены по умолчанию. Например, если мы сразу зададим, что «а» = 1, «b» = 2 и «с» = 3 (Рис.10).

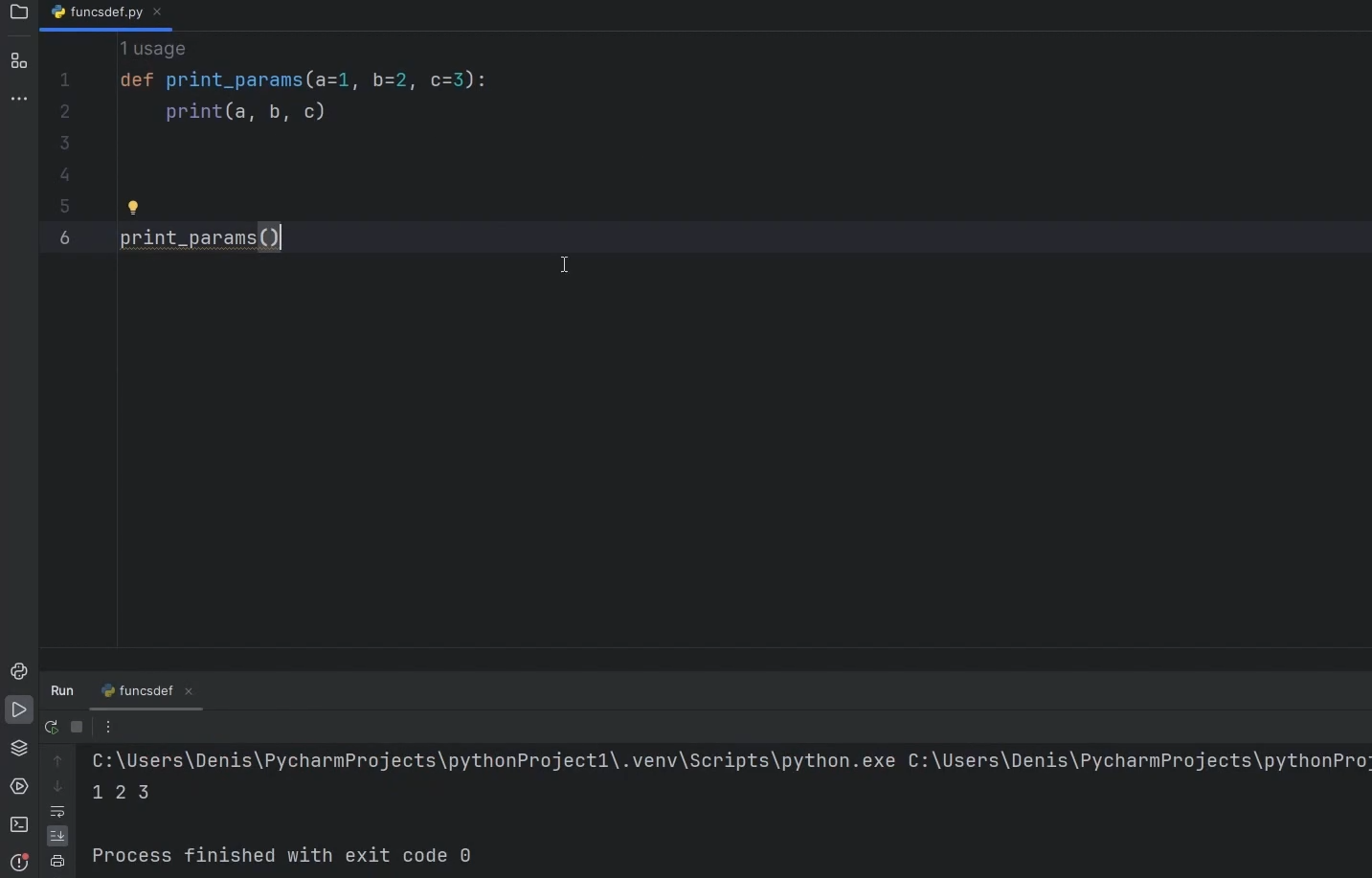


Рис.10  
  
В таком случае, даже если мы ничего не укажем в скобках, то увидим в ответе заданные нами раннее значения.  
  
Тем не менее, если нам необходимо, то мы можем вписать в скобки значения, которые могут не соответствовать заданным по умолчанию (Рис.11). В таком случае в ответе мы увидим их, то есть таким образом мы можем переопределить параметры.

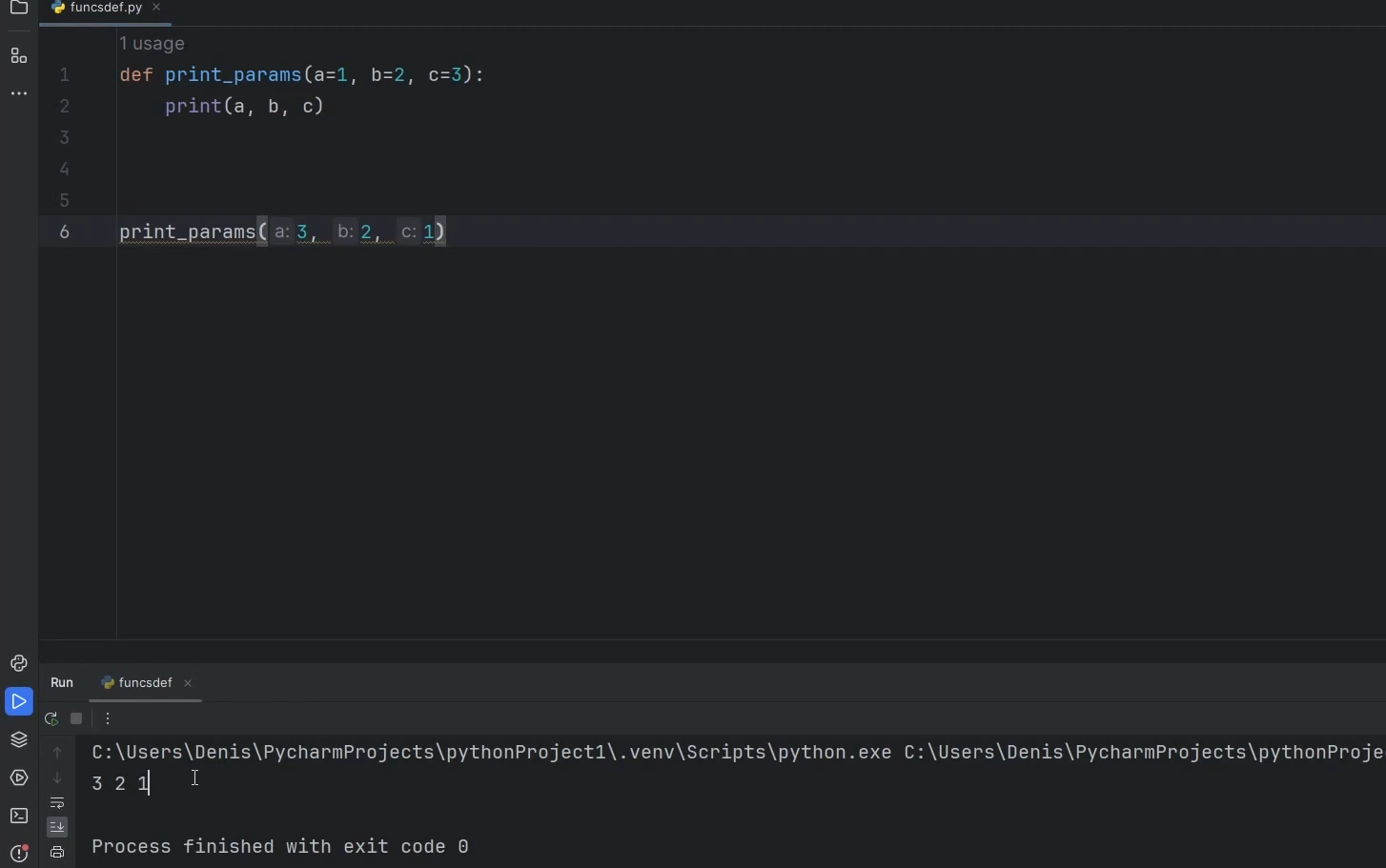


Рис.11  
  
Если мы запишем их по порядку, то они будут читаться позиционно. Но мы можем указать, что хотим изменить конкретный параметр, к примеру параметр «с» сделаем строкой (Рис.12). Запустим и увидим, что теперь «с» у нас строка.

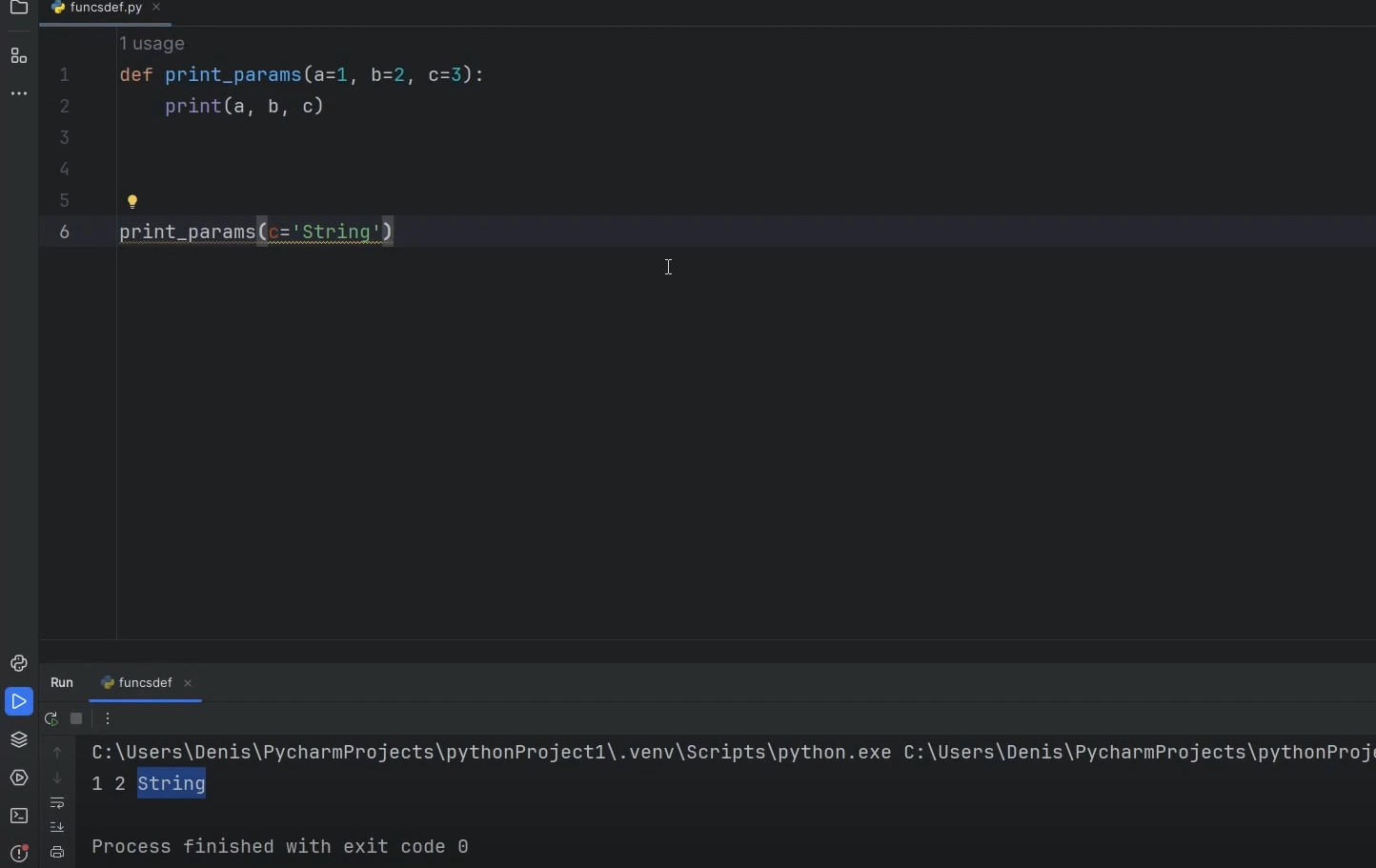


Рис.12  
  
Например, зададим также другие значения, которые автоматически позиционно передадутся параметрам «а» и «b» (Рис.13).



Рис.13  
  
Таким образом, мы можем задать значения одновременно позиционным и именованным параметрам. Однако, важно, чтобы позиционные параметры шли перед именованными.  
  
Если мы напишем какие-то значения после значения именованного параметра, то увидим ошибку (Рис.14).

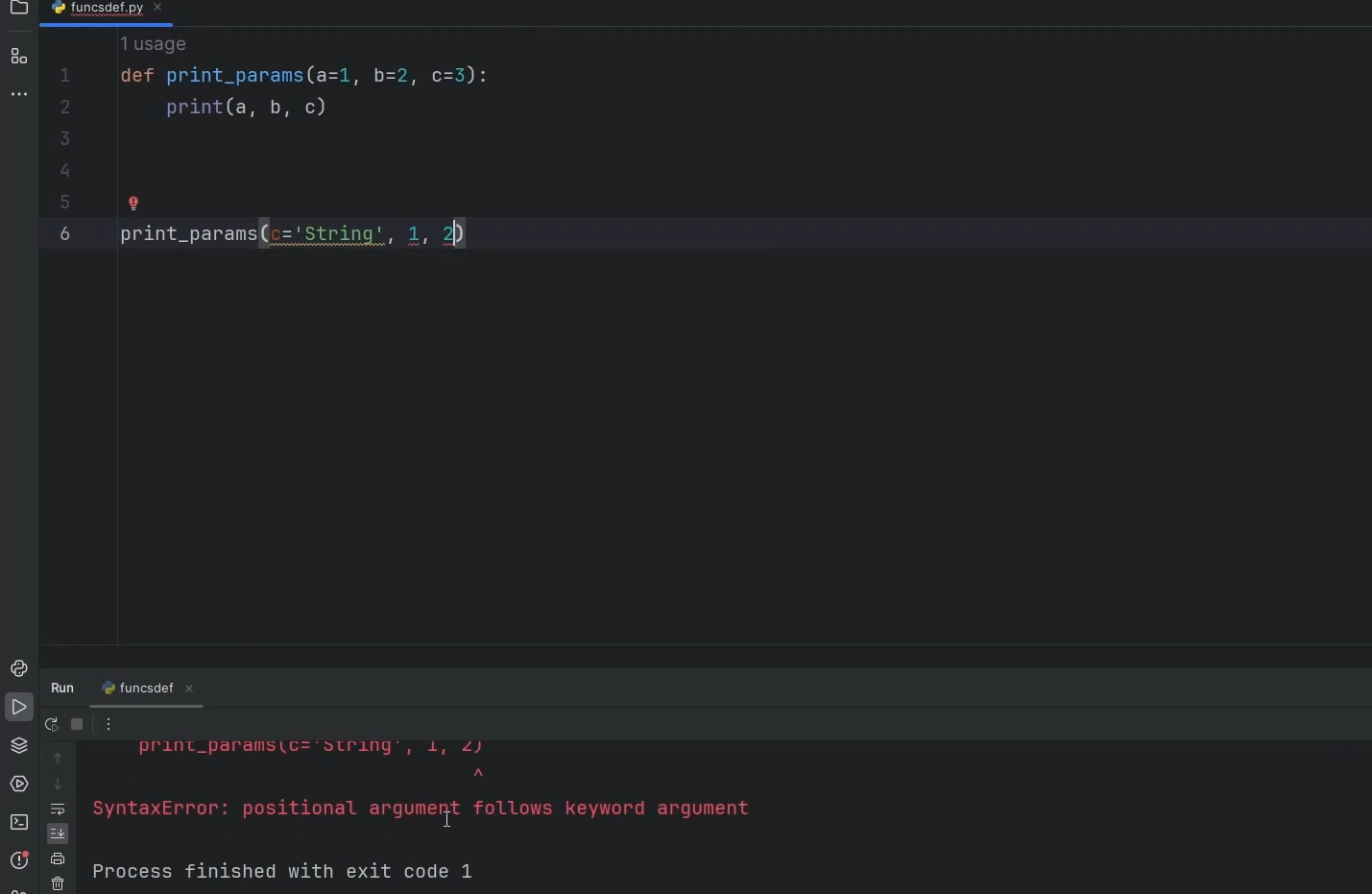


Рис.14  
  
Тем не менее, если мы передаем все именованные параметры, то они могут идти не по порядку (Рис.15).

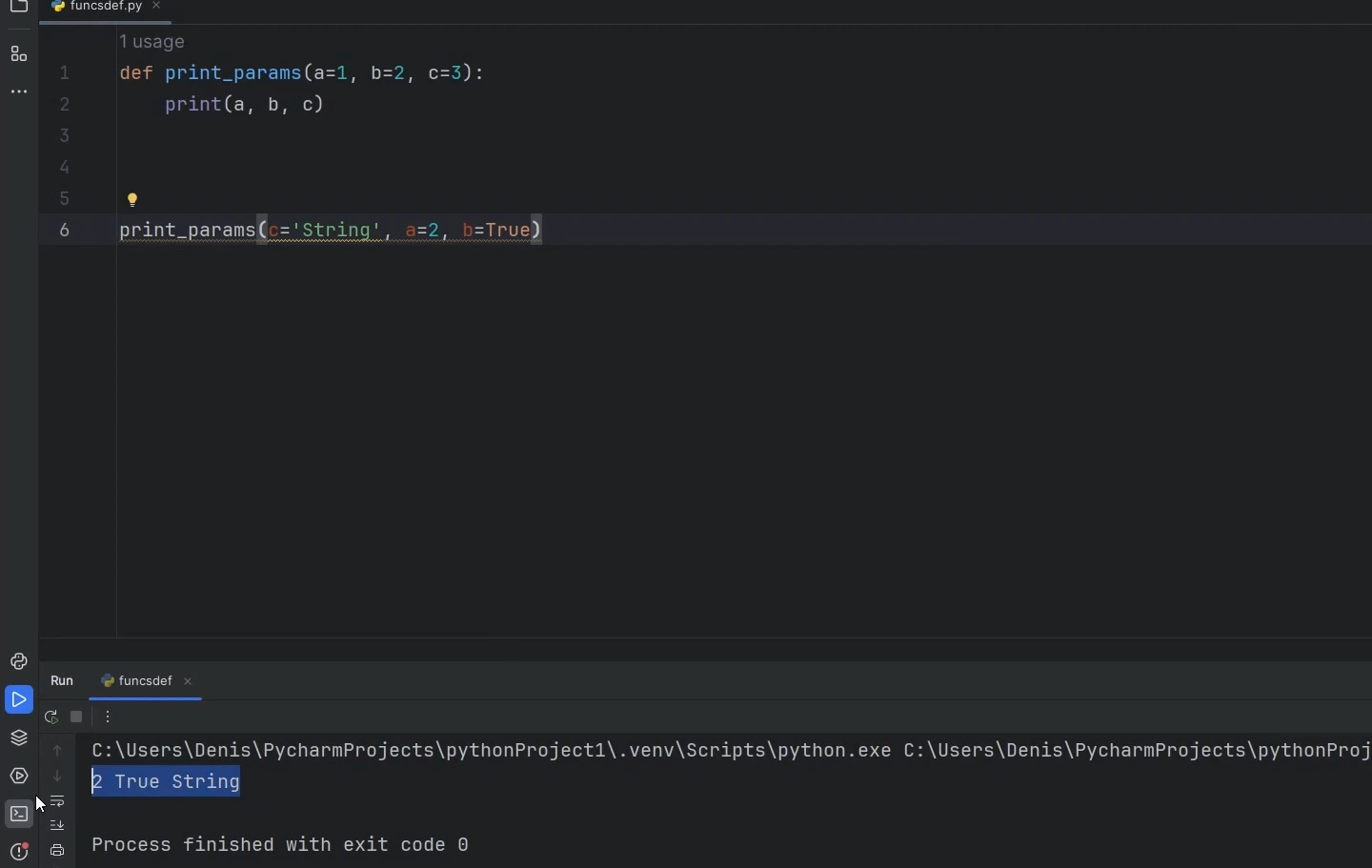


Рис.15  
  
Существует также способ конкретно указать, что параметр будет именованным. Нужно поставить знак «\*» перед этим параметром.  
  
Например, поставим «\*» перед параметром «b» (Рис.16). В таком случае первый параметр «а» у нас может быть позиционным, а параметры «b» и «с» будут именованными.

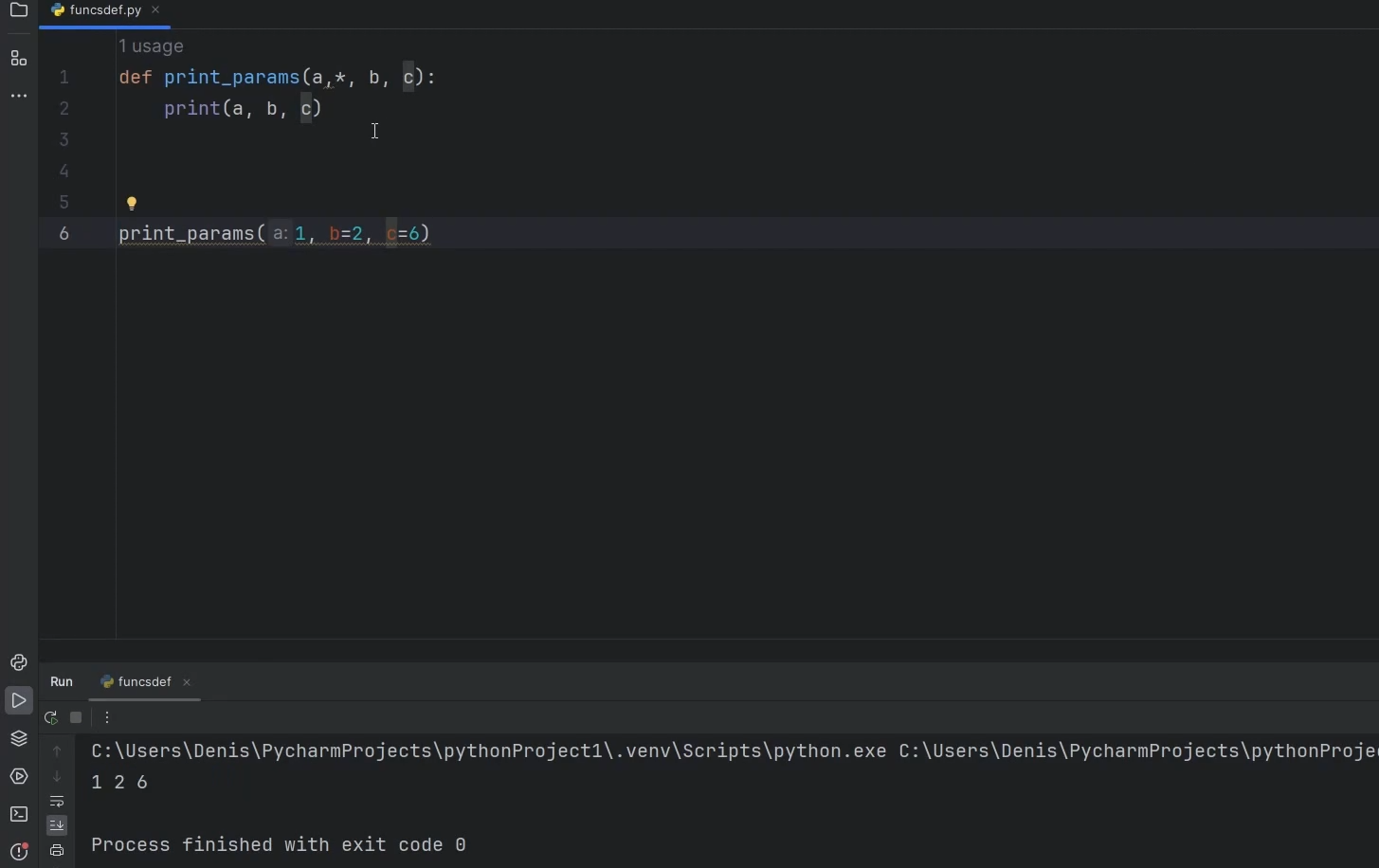


Рис.16  
  
То есть в данном случае мы не можем написать, что 2 будет значением позиционного параметра, мы просто получим ошибку (Рис.17). Все потому, что мы поставили «\*» перед двумя последними параметрами, все что идет после этого знака, мы должны конкретно указать.

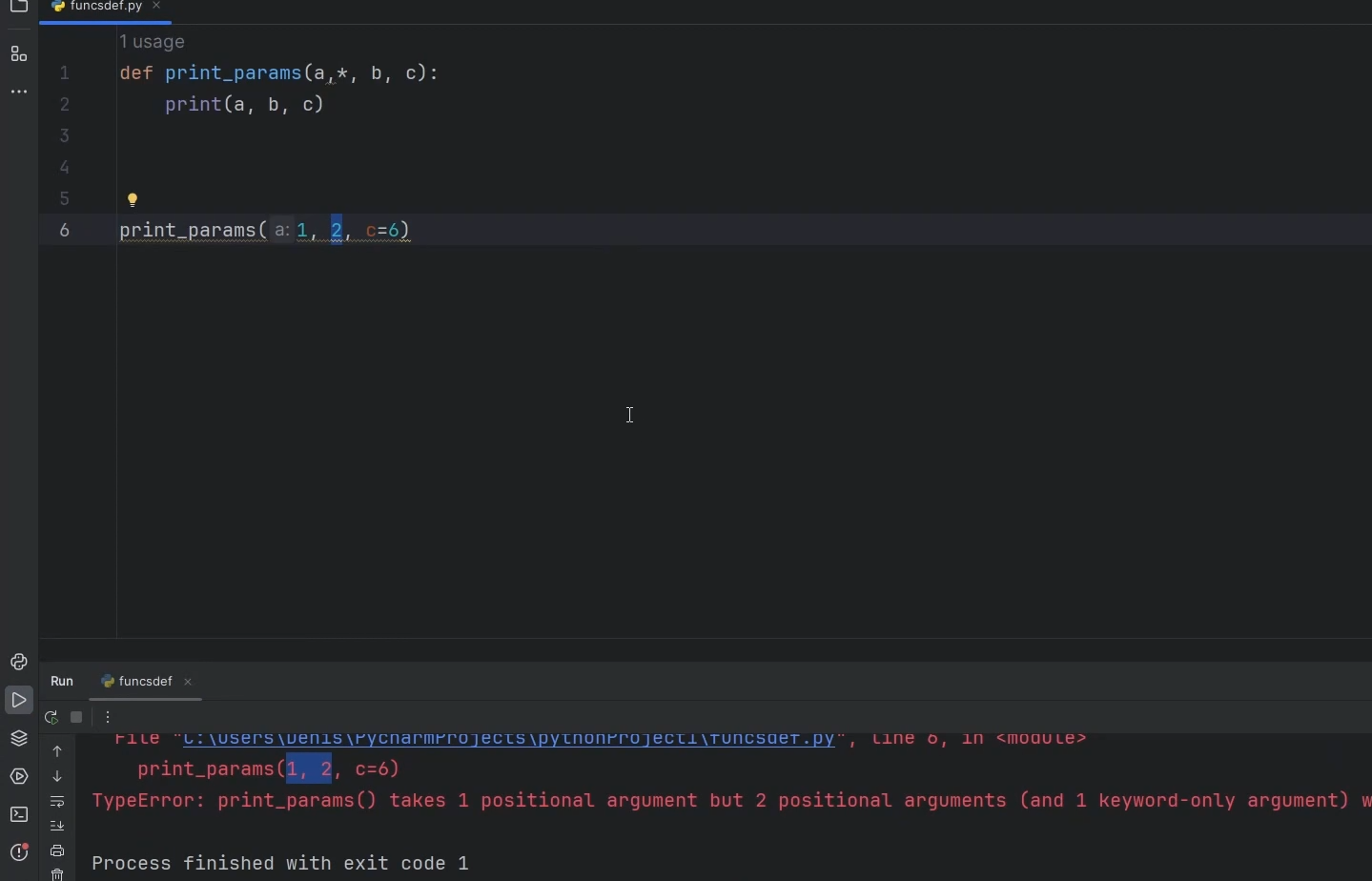


Рис.17  
  
Это правило работает и в том случае, если «\*» стоит в начале перед всеми параметрами (Рис.18).

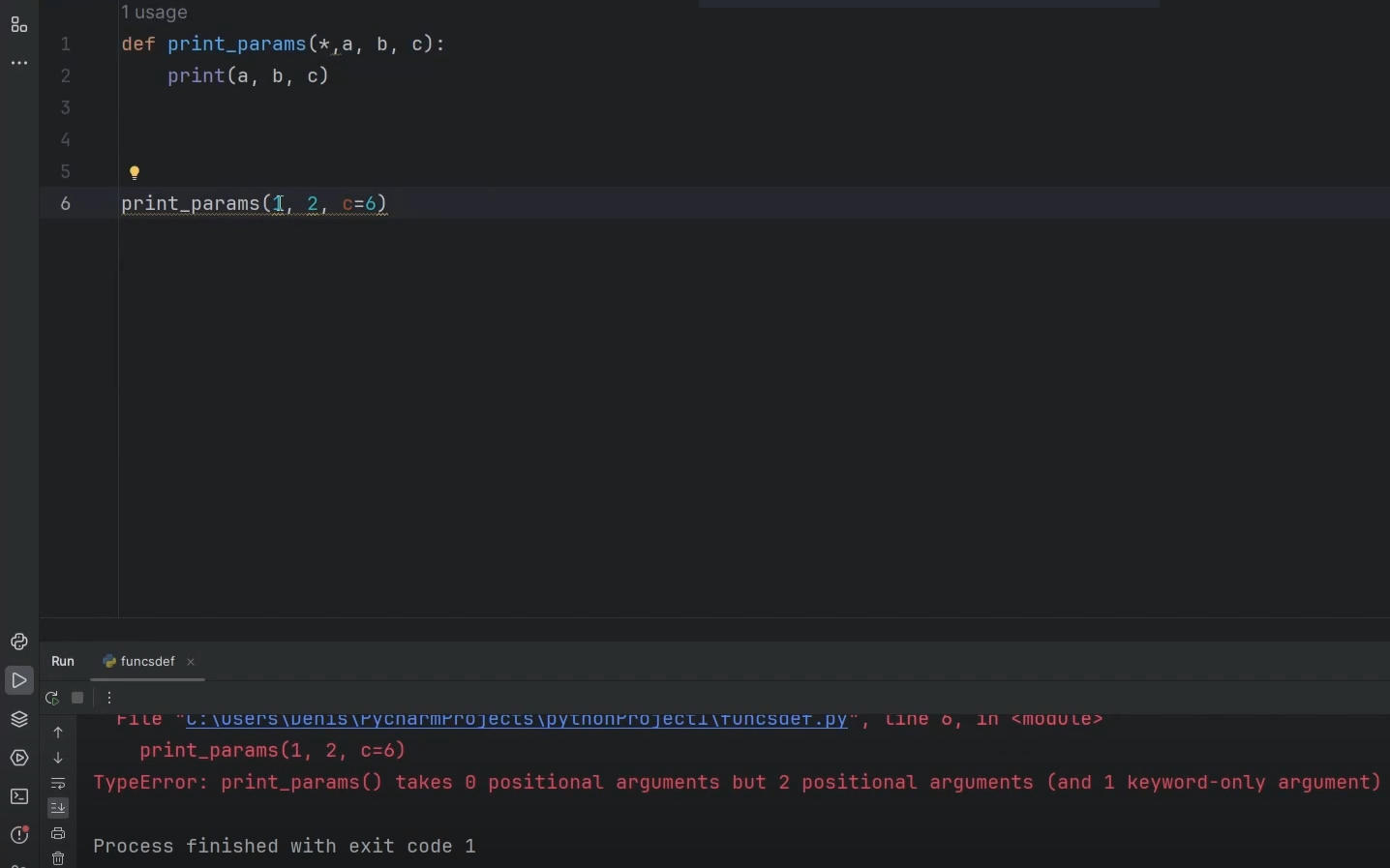


Рис.18  
  
Если мы конкретно укажем их значения, то увидим ответ (Рис.19).

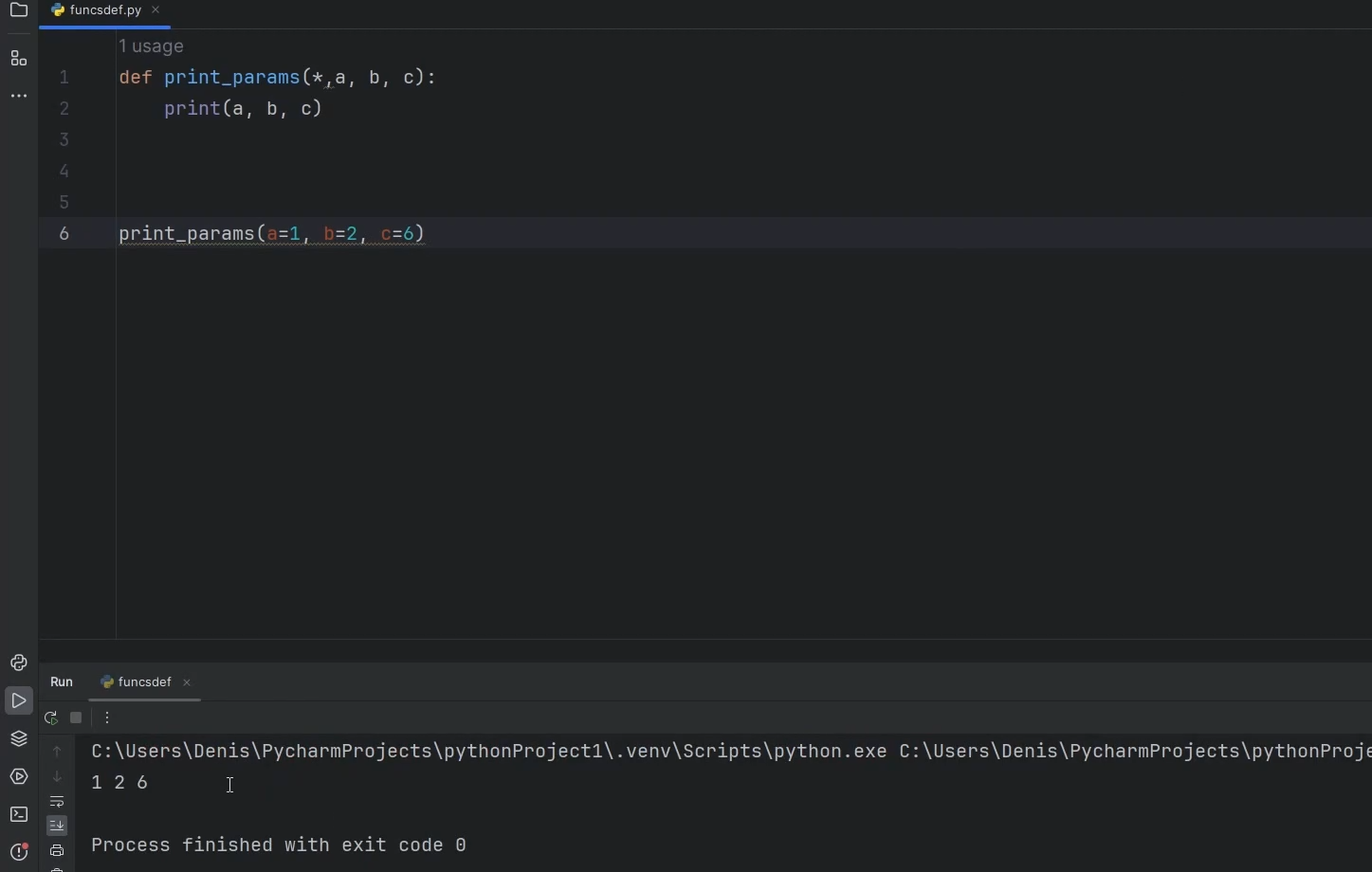


Рис.19