Форматирование

**%-форматирование.** Самый популярный способ (Python 2), который перешел в Python из языка С. Передавать значения в строку можно через списки и кортежи , а также и с помощью словаря. Во втором случае значения помещаются не по позиции, а в соответствии с именами.

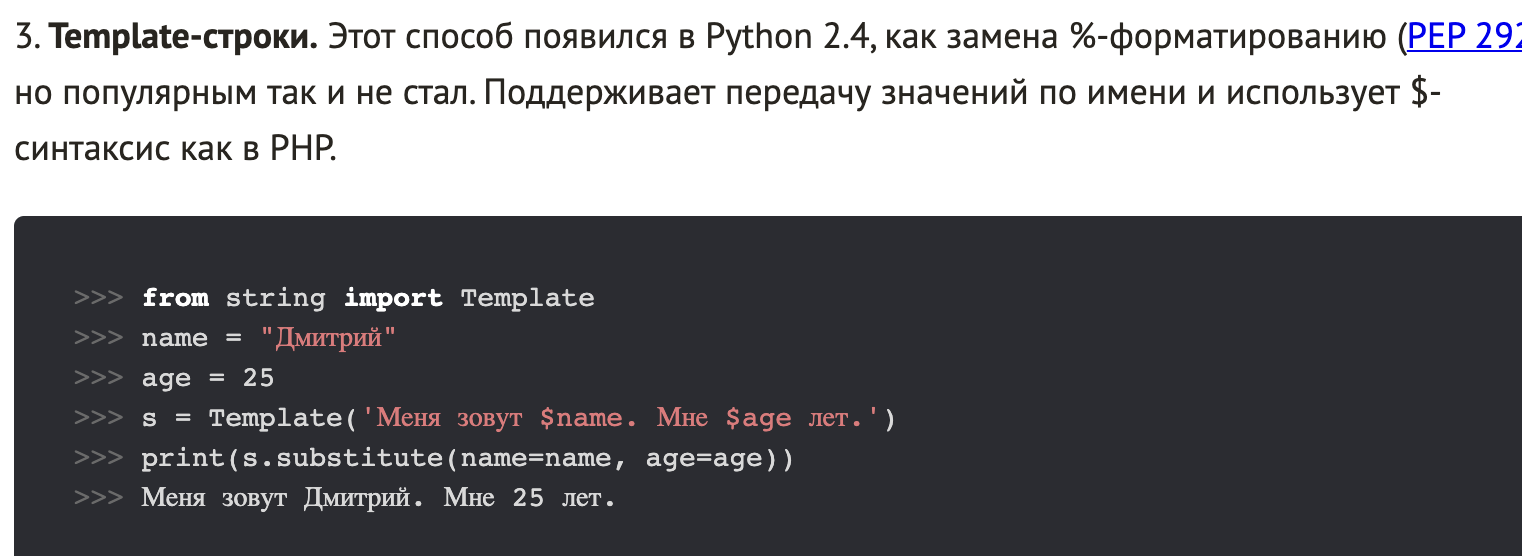
name = Den  
age = 25

print("Меня зовут %s. Мне %d лет." % (name, age))

print("Меня зовут %(name)s. Мне %(age)d лет." % {"name": name, "age": age})

**Добавленно >>>** К вопросу о том как и зачем использовать в %   
символы розширения форматирования

[**https://pythonworld.ru/osnovy/formatirovanie-strok-operator.html**](https://pythonworld.ru/osnovy/formatirovanie-strok-operator.html)

****

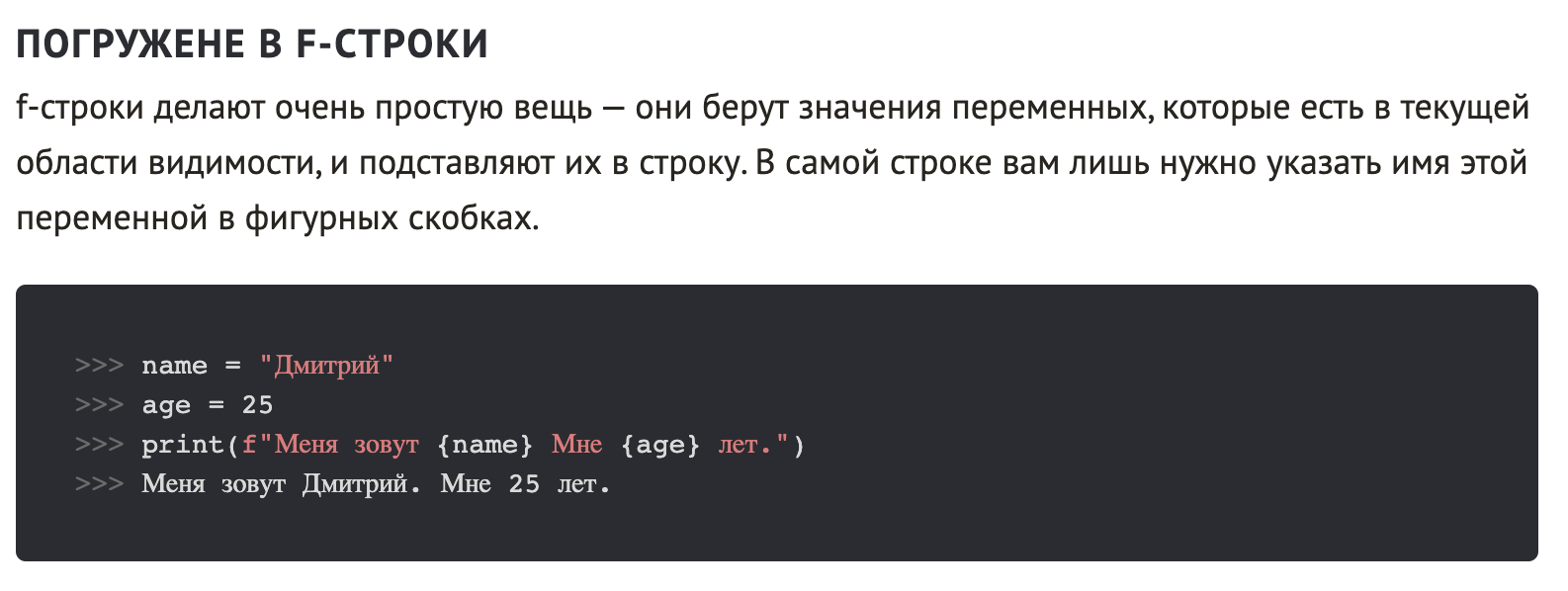
**Форматирование с помощью метода format().**   
  
<https://pythonworld.ru/osnovy/formatirovanie-strok-metod-format.html>  
  
Этот способ появился в Python 3 в качестве замены %-форматированию. Он также поддерживает передачу значений по позиции и по имени.

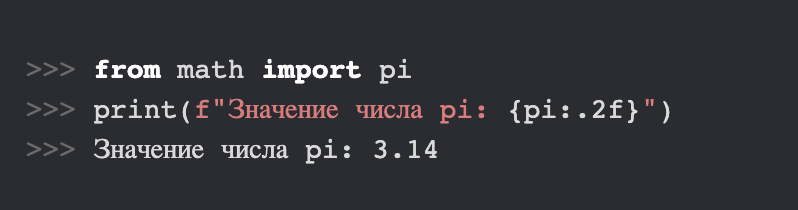
name = Den  
age = 25

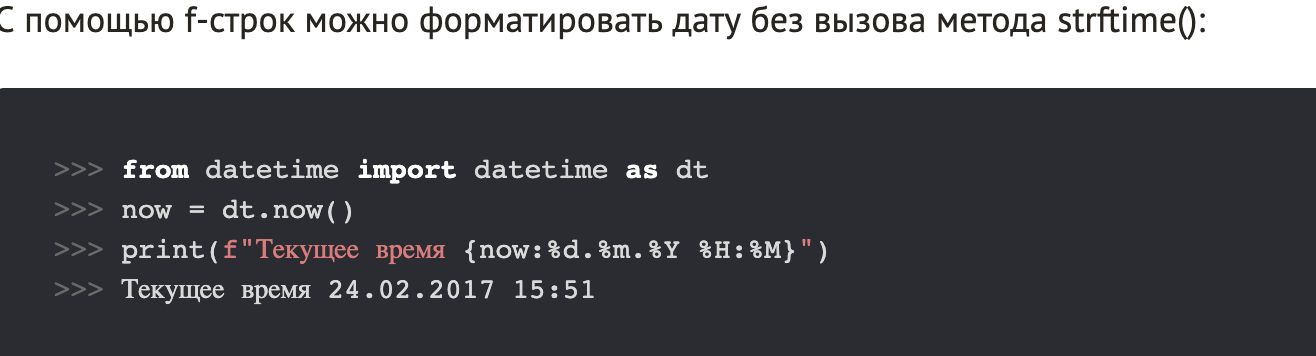
print("Меня зовут {}. Мне {} лет.".format(name, age)

print("Меня зовут {name} Мне {age} лет.".format(age=age, name=name)

**f-строки.** Форматирование, которое появилось в Python 3.6 ([PEP 498](https://www.python.org/dev/peps/pep-0498/)). Этот способ похож на форматирование с помощью метода format(), но гибче, читабельней и быстрей.



f-строки также поддерживают расширенное форматирование чисел:  




Они поддерживают базовые арифметические операции  
Позволяют обращаться к значениям списков по индексу,  
А также к элементам словаря по ключу  
Также вызывать функции  
  
Сравние скорости работы :  
  
  
