Форматирование строк с помощью f-строк

В Python 3.6 добавился новый вариант форматирования строк - f-строки или интерполяция строк. F-строки позволяют не только подставлять какие-то значения в шаблон, но и позволяют выполнять вызовы функций, методов и т.п.
Во многих ситуациях f-строки удобней и проще использовать, чем format, кроме того, f-строки работают быстрее, чем format и другие методы форматирования строк.
Синтаксис
F-строки - это литерал строки с буквой f перед ним. Внутри f-строки в паре фигурных скобок указываются имена переменных, которые надо подставить:

```
ip = '10.1.1.1'
mask = 24

f"IP: {ip}, mask: {mask}"
```

Напечатается: IP: 10.1.1.1, mask: 24
Аналогичный результат с format можно получить так: "IP: {ip}, mask: {mask}".format(ip=ip, mask=mask).
Очень важное отличие f-строк от format: f-строки это выражение, которое выполняется, а не просто строка. То есть, в случае с ipython, как только мы написали выражение и нажали Enter, оно выполнилось и вместо выражений {ip} и {mask} подставились значения переменных.
Поэтому, например, нельзя сначала написать шаблон, а затем определить переменные, которые используются в шаблоне:

```
f"IP: {ip}, mask: {mask}"
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-e6f8e01ac9c4> in <module>()
----> 1 f"IP: {ip}, mask: {mask}"

NameError: name 'ip' is not defined
```

Кроме подстановки значений переменных, в фигурных скобках можно писать выражения:

```
octets = ['10', '1', '1', '1']
mask = 24

print(f"IP: {'.'.join(octets)}, mask: {mask}")
```

Напечатается: IP: 10.1.1.1, mask: 24
После двоеточия в f-строках можно указывать те же значения, что и при использовании format:

```
oct1, oct2, oct3, oct4 = [10, 1, 1, 1]

print(f'''
      IP address:
      {oct1:<8} {oct2:<8} {oct3:<8} {oct4:<8}
      {oct1:08b} {oct2:08b} {oct3:08b} {oct4:08b}''')
```

Напечатается:

```
IP address:
10       1        1        1
00001010 00000001 00000001 00000001
```

Особенности использования f-строк

1. При использовании f-строк нельзя сначала создать шаблон, а затем его использовать, как при использовании format.

2. F-строка сразу выполняется и в нее подставляются значения переменных, которые должны быть определены ранее:

3. Если необходимо подставить другие значения, надо создать новые переменные (с теми же именами) и снова написать f-строку
4. При использовании f-строк в циклах, f-строку надо писать в теле цикла, чтобы она "подхватывала" новые значения переменных на каждой итерации:

```
ip_list = ['10.1.1.1/24', '10.2.2.2/24', '10.3.3.3/24']

for ip_address in ip_list:
    ip, mask = ip_address.split('/')
    print(f"IP: {ip}, mask: {mask}")
```

Напечатается:
IP: 10.1.1.1, mask: 24
IP: 10.2.2.2, mask: 24
IP: 10.3.3.3, mask: 24

Примеры f-строк

Базовая подстановка переменных:

```
intf_type = 'Gi'
intf_name = '0/3'

print(f'interface {intf_type}/{intf_name}')
```

Напечатается:
interface Gi0/3