

Step 1

Step on Stepik: <https://stepik.org/lesson/2414/step/1>

Рассмотрим следующую задачу. Мы хотим проверить число на четность, если число четное выводим слово "Четное", если нечетное выводим "Нечетное". Как можно проверить числа четность?

Напомню, что четные числа это числа - 2 4 6 8, то есть которые делятся на 2, а нечетные - 1 3 5 7, которые на 2 не делятся. Значит нам нужно проверить число на делимость на 2. Для этого мы можем использовать операцию взятия остатка. Мы берем остаток от деления нашего числа на 2, и если он равен 0, значит наше число четное. В противном случае число нечетное. Но нам не достаточно определить четность числа, необходимо еще выполнить разные действия, в зависимости от результата (в одном случае вывести слово "Четное", в другом "Нечетное"). Для такого рода ситуаций нам нужны условные конструкции. Условные конструкции в Python реализуются с помощью конструкции:

```
if условие:
    инструкция для случая, если условие истинно
else:
    инструкция для случая, если условие ложно
```

Применительно к нашей задаче это выглядит так:

Условия

```
if x % 2 == 0:  
    print('Четное')  
else:  
    print('Нечетное')
```



мы пишем ключевое слово **if**, далее задаем наше логическое условие (вычисляем остаток деления числа на 2 и сравниваем результат с 0). После двоеточия запишем инструкцию для случая, когда результат логического выражения (всего целиком) оказался **True**. Применительно к нашей программе - функция `print` выведет "Четное". А после **else:** описываем действия, которые нужно выполнить, если логическое условие вернуло **False**. Стоит обратить внимание на следующий момент, в обоих случаях блоки кода, описывающие действия (следующие после двоеточий) выделяются отступами. Если нам необходимо выполнить несколько команд, их необходимо записывать с одинаковыми отступами. Для отступов рекомендуется использовать 4 пробела. В IPython автоматические отступы делаются нажатием на клавишу **<Tab>**.