

Для написания кода можно воспользоваться онлайн-компилятором по ссылке:

https://www.onlinegdb.com/online_python_compiler

Необходимо ознакомиться с теорией и выполнить задачи в компиляторе. После выполнения делать скриншот выполненной задачи.

Диапазоны от и до

Зачастую возникает необходимость создать и обработать последовательность целых чисел в определённом диапазоне. Например, когда нужно узнать сумму всех целых чисел от 3 до 5 или просто перечислить все номера вагонов поезда.

Можно вручную создать список из чисел, либо выбрать способ проще: в Python есть функция `range()`, в которую передаются два целых числа: начало и конец диапазона. В результате будет создана последовательность, включающая все целые числа в указанном диапазоне.

Особенность этой последовательности в том, что **в неё не включается последнее число диапазона**:

```
three = range(0, 3)
```

Последовательность three будет включать в себя числа 0, 1 и 2.

Тройка в эту последовательность не войдёт:

последнее число в последовательность не включается.

Числа могут быть и отрицательными. Важно, чтобы первое число было меньше второго.

```
around_zero = range(-3, 3)
```

Последовательность around_zero будет включать в себя числа -3, -2, -1, 0, 1 и 2.

Приятная особенность последовательности `range()` в том, что её можно обрабатывать в цикле, как обычный список.

Например, можно сделать так:

```
around_zero = range(-3, 3)
```

Вместо списка в цикл передаётся переменная around_zero,

в ней хранится range() от -3 до 3

```
for i in around_zero:
```

Перебрать все числа в диапазоне от -3 до 3 и напечатать их:

```
    print(i)
```

Будет напечатано

-3

-2

-1

0

1

2

Внутреннюю переменную цикла, в которую передаются числа из последовательности, традиционно называют `i`, но её можно назвать как угодно.

Задать диапазон можно прямо в условии цикла, без промежуточной переменной `around_zero`.

Цикл переберёт все числа в диапазоне от -3 до 3 и напечатает их:

```
for i in range(-3, 3):
```

```
    print(i)
```

Результат будет тот же

Список вагонов через range()

Запустите код и посмотрите, что он напечатает. Замените в `range()` параметры на любые случайные — и вновь запустите код.

Будьте внимательны: второе число в `range()` — это значение, перед которым останавливается счётчик. Это число не включается в последовательность. Например, чтобы обработать числа от 1 до 100 включительно — диапазон должен быть таким: `range(1, 101)`.

Код	<pre>for i in range(1, 5): print("Вагон №" + str(i))</pre>
-----	--

Как и в ситуации с обычным списком, цикл будет по очереди передавать каждое число из диапазона `range` в переменную `i`. А в теле цикла эта переменная будет печататься.

Задача 1

В этом уроке четыре задачи. Напечатайте магическую мантру, которая поможет справиться с ними:

Я справлюсь с задачей 1

Я справлюсь с задачей 2

Я справлюсь с задачей 3

Я справлюсь с задачей 4

Я всех победю!

Вместо многоточий вставьте необходимый код — имя переменной и диапазон.

Внутреннюю переменную цикла можно назвать `i`: так традиционно называют переменную, принимающую числа из последовательного числового ряда.

Код	<pre>for ... in range(...): print('Я справлюсь с задачей', ...)</pre>
Подсказка В условии цикла после <code>for</code> задайте имя переменной, в которую будут передаваться числа из диапазона <code>range</code> . Обычно такую	<pre>print('Я всех победю!')</pre>

переменную-счётчик называют `i`, но можно назвать её как угодно.
Диапазон чисел в `range()` указывается двумя числами через запятую: `range(от, до)`. Последнее число в диапазон не включается.
Если в функцию `print()` передать несколько аргументов через запятую (вот так: `print('раз', 2)`), то они будут напечатаны в одну строку через пробел: `раз 2`.

Задача 2

Если бы Джек, который построил дом, возводил его из чисел, получилось бы так:

```
Это первый этаж.  
А это 2 этаж, он на один выше, чем этаж 1  
А это 3 этаж, он на один выше, чем этаж 2  
...
```

Постройте с помощью цикла десятиэтажный дом. Последней строкой цикл должен вывести:

```
А это 10 этаж, он на один выше, чем этаж 9
```

Код

Подсказка

Начать строительство надо со второго этажа, а закончить — на десятом. Значит, в `range()` нужно указать диапазон чисел от 2 до 11.

В переменную `i` по очереди передаются все числа из диапазона, указанного в `range()`.

Значит, текущий этаж — это `i`, а предыдущий этаж — это `i-1`.

Числа и фрагменты текста можно напечатать в одну строку, перечислив их в функции `print()` через запятую; при этом они будут выведены через пробел.

Например, `print('Дом', 'в', 100, 'этажей')` напечатает Дом в 100 этажей.

```
print('Это первый этаж.')  
# Первый этаж построен, начинайте  
строить со второго  
for i in ...  
    # Здесь вместо многоточий  
    # вставьте номер текущего этажа,  
    # вычислите и вставьте номер  
    предыдущего этажа.  
    print('А это', ..., 'этаж, он на один выше,  
        чем этаж', ...)
```

Список задом наперёд

Если нужно пройти по списку или по диапазону чисел в обратном порядке — вызывайте функцию **reversed()**: она переворачивает и список, и диапазон; чтение любой последовательности начнётся с конца.

Для примера напишем симулятор новогодних курантов.

Функция `range()` не включает в последовательность чисел второй аргумент, поэтому для `reversed(range(1, 13))` отсчёт начнётся с 12.

```
1 for i in reversed(range(1, 13)):
2     print(i, 'бoмм!')
3
4 print('С новым годом!')
```

С новым счастьем!

Функция `reversed()` может развернуть в обратном направлении любую последовательность:

Можно обратить вспять обычный список:

```
seasons = ['зима', 'весна', 'лето', 'осень']
```

```
for season in reversed(seasons):
```

Переменную цикла, в которую будут передаваться элементы «перевернутого» списка seasons,

назовём season

```
print(season)
```

Задача 3

Разверните год наоборот: напечатайте месяцы в обратном порядке.

Код	<pre>months = ['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль', 'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь']</pre>
Подсказка	<pre>for ... in ...: print(...)</pre>
<ul style="list-style-type: none">• Обратить список можно функцией <code>reversed(имя_списка)</code>.• После <code>for</code> вместо многоточия поставьте название переменной цикла. Обычно это название образуют от имени списка, который обрабатывается в цикле.• Чтобы перебрать список в цикле — нужно передать его в условие цикла. Обратить список можно прямо в условии: <code>for month in reversed(имя_списка)</code>.• После условия цикла должно стоять двоеточие.	

Полезная переменная снаружи цикла

Написать циклом стихотворение про зайчика:

1, 2, 3, 4, 5, вышел зайчик погулять!

Здесь числа и фрагмент текста объединены в одну строку. Это означает, что задачу не удастся решить, печатая числа прямо из цикла (в этом случае каждое число окажется на отдельной строке).

Сначала все части строки нужно записать в дополнительную переменную, и потом напечатать её одним вызовом `print()`.

Эту переменную назовём `bunny_counter` и объявим снаружи цикла. На каждой итерации цикла будем дописывать в неё очередное число из `range()`.

Выведем на печать эту переменную и фразу «вышел зайчик погулять!» — и готово.

Числа из `range()` надо преобразовать в строку, иначе не получится объединить их с текстовой строкой.

Код

```
1 bunny_counter = '' # Создали переменную bunny_counter, её значение - пустая строка.
2
3 for i in range(1, 6):
4     # На каждой итерации цикла
5     # к переменной bunny_counter будет дописываться
6     # очередная цифра, запятая и пробел (чтобы числа в строке не слиплись).
7     # Получившееся значение будет присвоено той же переменной bunny_counter
8     bunny_counter = bunny_counter + str(i) + ', '
9
10 print(bunny_counter + 'вышел зайчик погулять!')
```

Результат

1, 2, 3, 4, 5, вышел зайчик погулять!

Задача 4

Настало время великих стартов: Tesla улетела на гелиоцентрическую орбиту за Марсом, а вам предстоит отправить ракету с питоном на Сатурн.

Сгенерируйте строку с обратным предстартовым отсчётом.

Она должна выглядеть так:

```
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, поехали!
```

Это задание очень похоже на пример со считалкой про зайца

Код	<code>countdown_str = "</code>
Подсказка	<code>for ...</code> <code>countdown_str = countdown_str + ...</code>

В условии цикла укажите переменную `i` и диапазон чисел в качестве обрабатываемого списка: `range(от, до)`. Предстартовый отсчёт ведётся от большего к меньшему, значит, список надо развернуть задом наперёд функцией `reversed()`. На каждой итерации в переменную `countdown_str` дописывайте:

- Очередное число из списка (не забудьте преобразовать его в `str()`).
- Строку `' '`, чтобы отделить в тексте последующее число от предыдущего. Обращайте внимание на пробел после запятой.

```
countdown_str = countdown_str + 'поехали!'
print(countdown_str)
```