

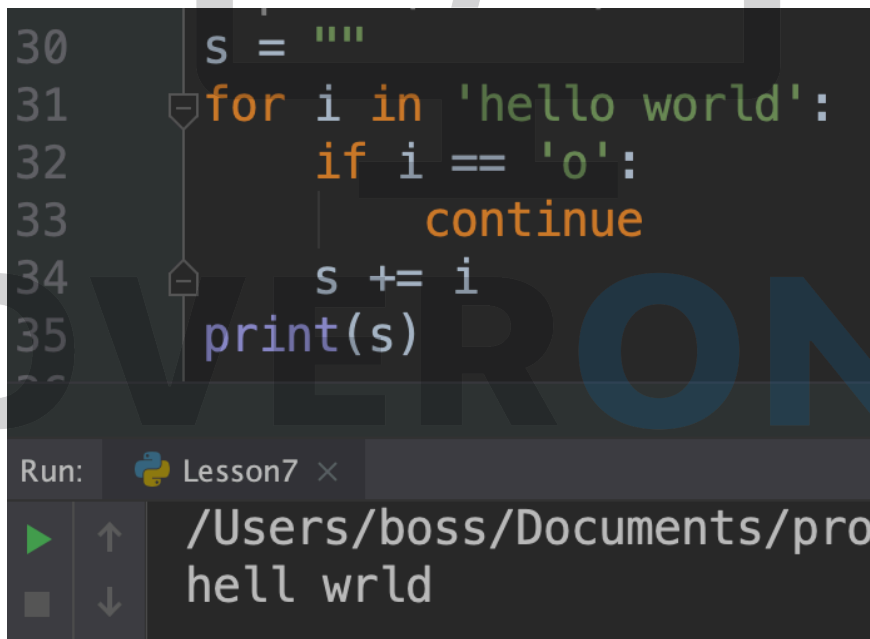
Тема 5. Оператор continue. Оператор break. Ключевое слово else.

Сегодня мы закончим изучать тему циклов и научимся использовать операторы continue и break, а также узнаем в каких случаях ставится else и для чего он нужен в циклах.

Оператор continue

Оператор **continue** **начинает следующий проход цикла, минуя оставшееся тело цикла** (for или while), то есть когда встречается данный оператор текущая итерация цикла прерывается и начинается следующая. Как правило, continue служит, когда нам необходимо в каких-то случаях пропустить выполнение тех или иных команд.

Например мы хотим из строки 'hello world' убрать все o, тогда программа будет выглядеть так:



```
30 s = ""
31 for i in 'hello world':
32     if i == 'o':
33         continue
34     s += i
35 print(s)
```

The screenshot shows a code editor with the above Python code. Below the editor, a terminal window titled 'Run: Lesson7' shows the output of the program: 'hell wrld'. The code uses a for loop to iterate over the string 'hello world'. When it encounters the character 'o', it uses the 'continue' statement to skip the rest of the loop body and move to the next iteration. This results in the output 'hell wrld', where the 'o' has been removed.

Оператор break

Данный оператор, в отличие от continue, не просто прерывает текущую итерацию, а **полностью прекращает выполнения цикла**. Он используется, когда нам необходимо прервать работу чтобы не возникло какой-нибудь ошибки.

Например мы хотим прервать работу цикла когда встретим символ 'r'. Наша программа может выглядеть так:

```
30 s = ""
31 for i in 'hello world':
32     if i == 'r':
33         break
34     s += i
35 print(s)
```

Run: Lesson7 x

/Users/boss/Documents/program
hello wo

Ключевое слово else

Слово else, примененное в цикле for или while, проверяет, был ли произведен выход из цикла инструкцией break, или же "естественным" образом. Блок команд внутри else выполнится только в том случае, если выход из цикла произошел без помощи break.

```
30 s = ""
31 for i in 'hello Python':
32     if i == 'r':
33         break
34     s += i
35 else:
36     print("Символ 'r' не встретился")
37 print(s)
```

Run: Lesson7 x

/Users/boss/Documents/programming
Символ 'r' не встретился
hello Python

В данной программе мы прерываем досрочно цикл, когда нам попадаетея символ 'r', так как в строке 'hello Python' данного символа нет, то после завершения цикла мы попали в блок else, а там прописана лишь одна команда, которая выводит на экран строку "Символ 'r' не встретился".

Давайте заменим строку 'hello Python' на 'hello World!!!'

```
30 s = ""
31 for i in 'hello World!!!':
32     if i == 'r':
33         break
34     s += i
35 else:
36     print("Символ 'r' не встретился")
37 print(s)
```

Run: Lesson7 x

/Users/boss/Documents/programming,
hello Wo

Как видите, в этот раз в блок else мы не попали, так как наш цикл прервался досрочно из-за оператора break.

Практическое задание:

- 1) Написать программу, которая выводит числа от 0 до 100 на экран, пропуская числа кратные 7

Примечание: Напоминаю, есть оператор % - остаток от деления, если число делится на цело то остаток от деления будет 0, например

$6 \% 2 = 0$,

$12 \% 4 = 0$

$12 \% 5 = 2$, так как ближайшее число, которое делится на 5 - это 10, следовательно $12 - 10 = 2$

- 2) Пользователь вводит строку, программа должна проверить наличие символа 'q' в строке и в случае его обнаружения прервать выполнение программы, если символа нет в строке - вывести данную фразу «Программа успешно отработала, символ q отсутствует»