

OR			AND		
	True	False		True	False
True	True	True	True	True	False
False	True	False	False	False	False



ИЗУЧАЕМ:

- булеву логику и цикл while;
- break и continue.

ПОВТОРЯЕМ (ЗАКРЕПЛЯЕМ):

- преобразование строки в int;
- проверку на ввод числа;
- input и print;
- if, elif и else;
- Конкатенацию (сложение) строк;
- срезы строк.

False	False	False	False	False	False	False	False
False	True	True			True	False	False
False	True	True			True	True	False
False	False	True			True	True	False
False	False	True			True	True	False
False	True	True	True	True	True	True	False



True и False

Логический тип представлен двумя постоянными значениями False и True. Значения используются для представления истинности:

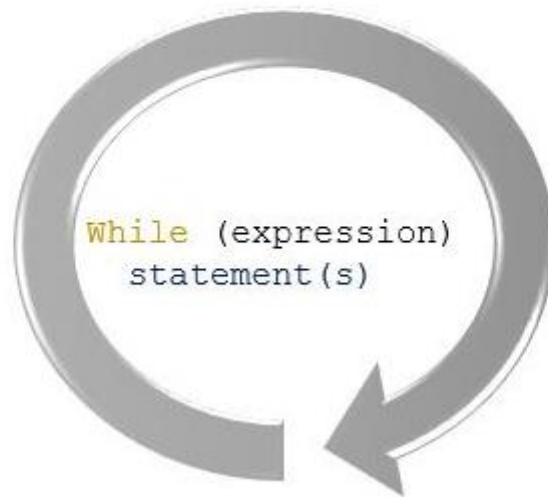
True (или 1): результат – «ИСТИНА»

False (или 0): результат – «ЛОЖЬ»

ПРИМЕР:

- `10 > 2 and 10 > 9 # True`
- `10 > 2 and 10 > 11 # False`

Python while loop



Цикл **While**

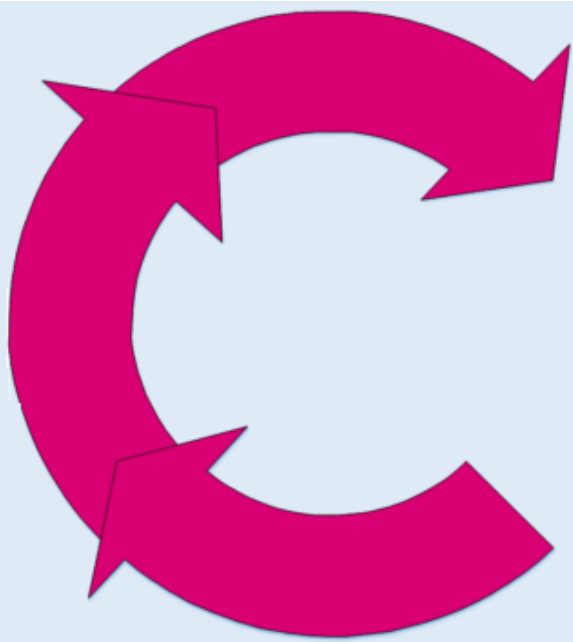
СИНТАКСИС:

```
while <условие>:  
    <тело цикла>
```

ПРИМЕР:

```
n = 5  
while n > 0:  
    n -= 1  
    print(n)
```

РЕЗУЛЬТАТ: 4 3 2 1 0



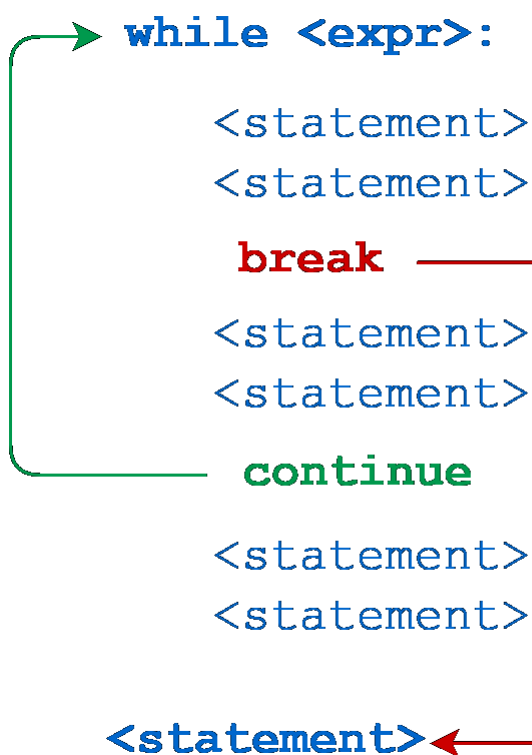
continue

+

break



Continue и break



```
n = 5
while n > 0:
    n -= 1
    if n == 2:
        break
    print(n)
print('Loop ended.')
```

Результат:

4
3
Loop ended.

```
n = 5
while n > 0:
    n -= 1
    if n == 2:
        continue
    print(n)
print('Loop ended.')
```

Результат:

4
3
1
0
Loop ended.

Прерываем или возвращаем в начало



ПРОГРАММА: счет в кафе

```
print ("Произведем подсчет расходов в кафе.")  
print ("Вводите стоимость без копеек:\n")  
  
bun = input ("Слоенная булка: ")  
tea = input ("Чай с сахаром: ")  
  
total = bun + tea  
  
print ("\nРасходы составили: ", total, " рублей.")
```

Разумеется, счет в кафе мы не получим.
Результатом будет набор введенных цифр.
Убедимся? Пишем код и тестируем.

Не удивительно!!!

Ведь программа объединила две строки



ПРОГРАММА: счет в кафе

Исправляем ошибку:

```
print ("Произведем подсчет расходов в кафе.")
print ("Вводите стоимость без копеек:\n")

bun = int(input ("Слоенная булка: "))
tea = int(input ("Чай с сахаром: "))

total = bun + tea

print ("\nРасходы составили: ", total, " рублей.")
```

Вроде работает! Но что, если пользователь ввел не целое число. Или вовсе не число? Вернуть пользователя назад поможет вечный цикл.

```
While True:
    #тело цикла
```

Он будет выполняться пока мы его не прервем директивой

break



ПРОГРАММА: счет в кафе

Внедряем цикл:

```
print ("Произведем подсчет расходов в кафе.")
print ("Вводите стоимость без копеек:\n")

total = 0

while True:
    bun = input ("Слоенная булка: ")
    tea = input ("Чай с сахаром: ")

    if (bun.isdigit() and tea.isdigit()):
        break
    elif not bun.isdigit():
        print («Булка за: " + bun + " руб.? Не верю. И без копеек.
        Начнем сначала...\n")
        continue
    elif not tea.isdigit():
        print («Чай за: " + tea + " руб.? Не верю. И без копеек.
        Начнем сначала...\n")
        continue

total = int(bun) + int(tea)
print ("\nРасходы составили: ", total, " рублей.")
```




ПРОГРАММА: счет в кафе

Если пользователь вводит число с «.» или «,»?

Например: 48,50 или 48.50.

Возможно принудительно отбросить копейки.

Для этого вводим проверку:

```
index1 = bun.find(".")
index2 = bun.find(",")
if index1 > 0:
    bun = bun [0:index1] # срез строки от 0 позиции до «.»
elif index2 > 0:
    bun = bun [0:index2] # срез строки от 0 позиции до «,»
```

Аналогично вводим для **tea**.

Полагаю, справитесь и найдете куда вставить эту проверку?

Увы, копейки отбрасываются без округления ☹️

Но мы ведь и просили без копеек 😊