



9-11 классы

Программирование на Python

Презентация занятия

ЦИКЛЫ В PYTHON. WHILE & FOR.

4 занятие



Минцифры
России



20.35
УНИВЕРСИТЕТ

Программирование
на Python

Теоретическая часть

**ЦИКЛЫ В PYTHON.
WHILE & FOR.**

4 занятие



инжинириум®

МГТУ им. Н.Э. Баумана

2022

Тема: Циклы в Python. While & for.

```
if <логическое выражение>:  
    <действия, которые будут>  
    <выполнены, если лог. выраж ист.>  
else:  
    <действия, которые будут>  
    <выполнены, если лог. выраж ложно>
```

```
1 state = input('How are you?')  
2 if state == 'sleepy':  
3     print('Sleep in')  
4 else:  
5     print('Start your day!')
```

True and True -> True
True and False -> False
False and True -> False
False and False -> False

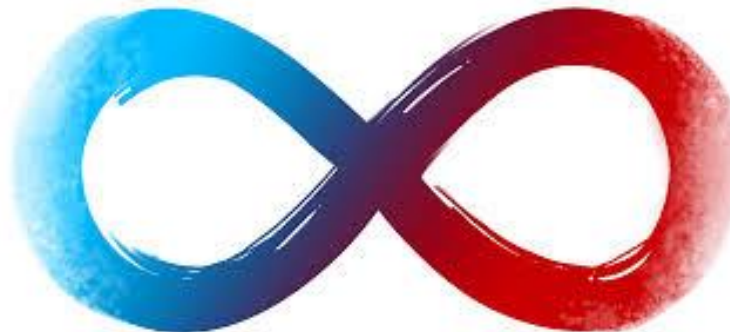
True or True -> True
True or False -> True
False or True -> True
False or False -> False



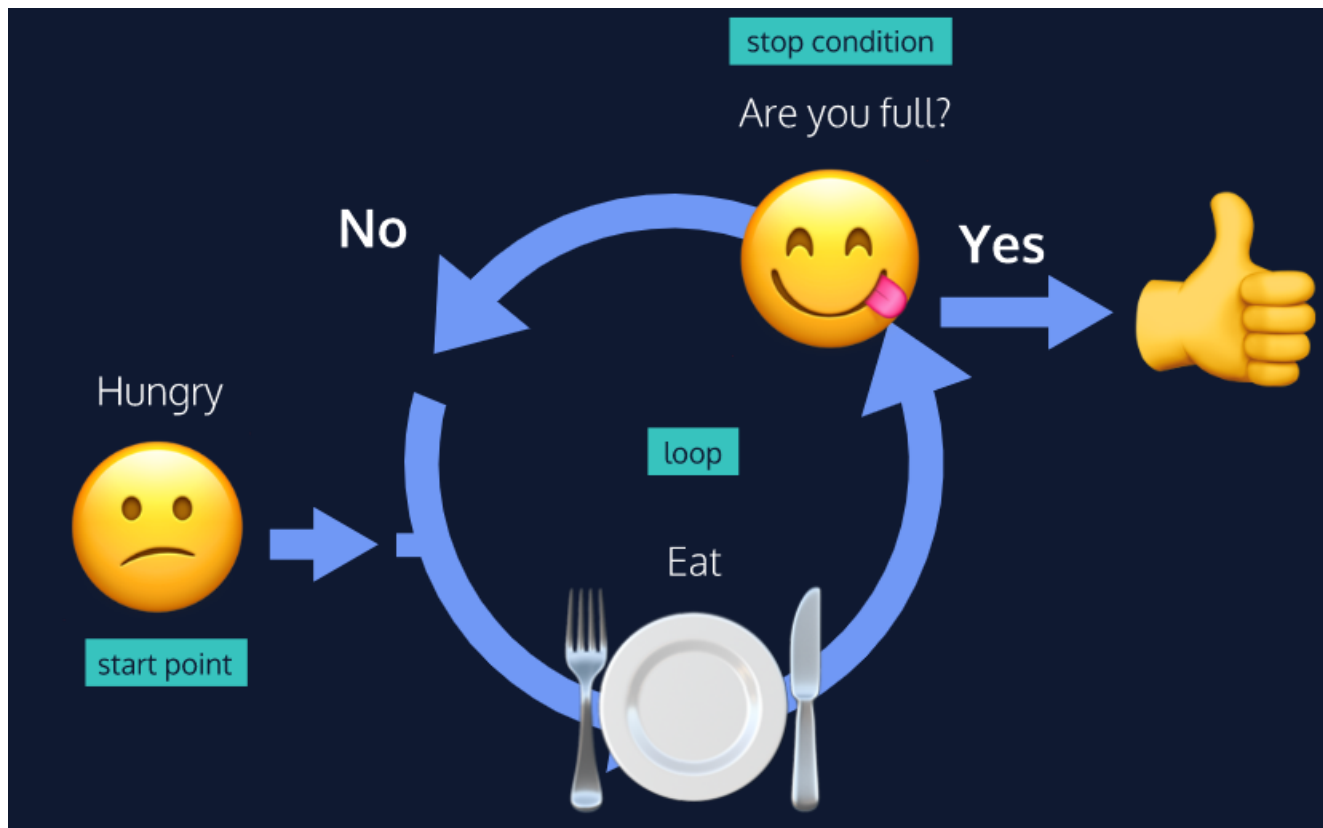
Тема: Циклы в Python. While & for.

```
1  c = 0
2  if c != 10:
3  |      print(c)
```

```
1  c = 0
2  while c != 10:
3  |      print(c)
```

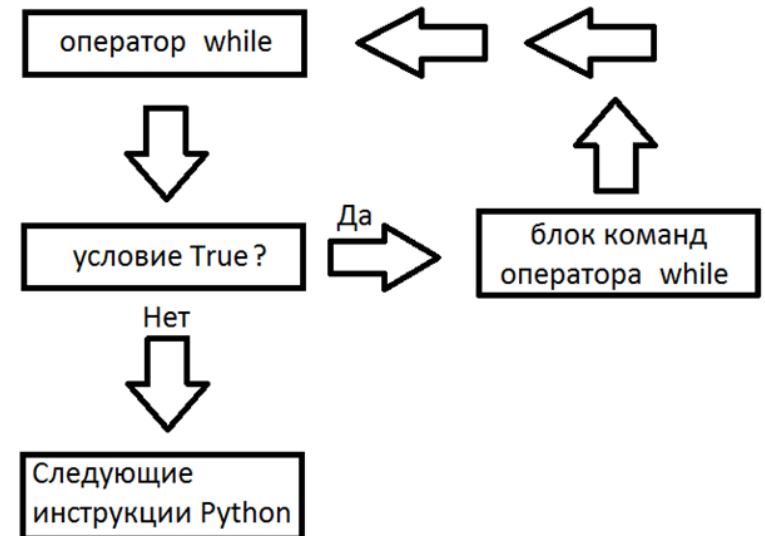
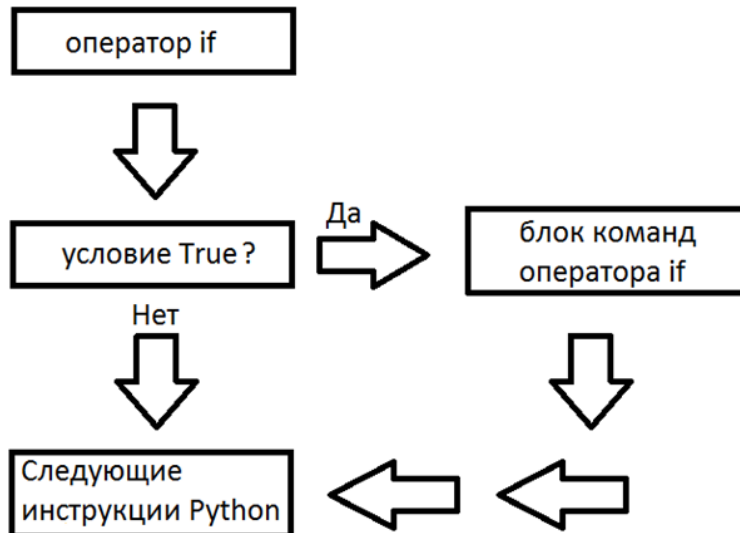


Тема: Циклы в Python. While & for.





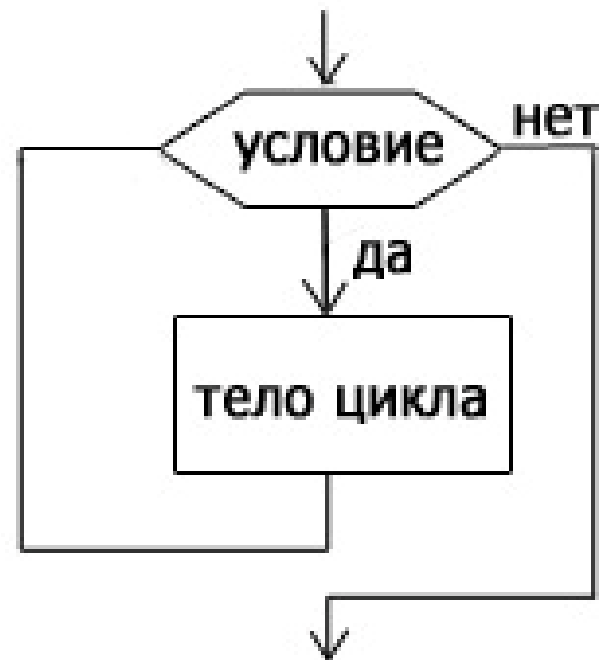
Тема: Циклы в Python. While & for.





Тема: Циклы в Python. While & for.

```
1 print('Start')
2 c = 0 # counter - счётчик
3 while c != 10:
4     print(c)
5     c += 1
6 print('Stop')
```





Тема: Циклы в Python. While & for.

Обратный отсчёт начнётся с 10

10

9

8

7

6

5

4

3

2

1

Поехали!





Тема: Циклы в Python. While & for.

```
1  c = int(input('Обратный отсчёт начнётся с '))
2  while c != 0:
3      print(c)
4      c -= 1
5  print('Поехали!')
```





Тема: Циклы в Python. While & for.

default = 0
Start

```
1  for i in range(0, 10, 1):  
2      print(i)
```

Stop

default = 1
Step



Программирование
на Python

Практическая часть

**ЦИКЛЫ В PYTHON.
WHILE & FOR.**

4 занятие



инжинириум®

МГТУ им. Н.Э. Баумана

2022



Тема: Циклы в Python. While & for.

Задание 1

Реализуйте программу
“обратный отсчёт” с
помощью цикла for

Обратный отсчёт начнётся с 10

10

9

8

7

6

5

4

3

2

1

Поехали!





Тема: Циклы в Python. While & for.

Задание 2

Напечатайте столбец таблицы умножения

(номер столбца по вариантам)

Задание 3

Напечатайте столбец таблицы умножения, где номер столбца выбирает пользователь.

```
1 * 7 = 7
2 * 7 = 14
3 * 7 = 21
4 * 7 = 28
5 * 7 = 35
6 * 7 = 42
7 * 7 = 49
8 * 7 = 56
9 * 7 = 63
```

Какой столбец? 5

```
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
```





Тема: Циклы в Python. While & for.

2 * 8 = 16
2 * 9 = 18

Задание 4

Напечатайте всю таблицу
умножения в один столбце

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

4 * 1 = 4
4 * 2 = 8
4 * 3 = 12

Задание 5

Напечатайте всю таблицу
умножения в два столбца

3 * 7 = 21
3 * 8 = 24
3 * 9 = 27

4 * 7 = 28
4 * 8 = 32
4 * 9 = 36

5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45

6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54

7 * 1 = 7
7 * 2 = 14
7 * 3 = 21

8 * 1 = 8
8 * 2 = 16
8 * 3 = 24



Тема: Циклы в Python. While & for.

Задание 6

Сделаем нашу таблицу красивой). В таблице должно быть 2 колонки

```
+#####+#####+
#   1 * 1 = 1   #   2 * 1 = 2   #
#   1 * 2 = 2   #   2 * 2 = 4   #
#   1 * 3 = 3   #   2 * 3 = 6   #
#   1 * 4 = 4   #   2 * 4 = 8   #
#   1 * 5 = 5   #   2 * 5 = 10  #
#   1 * 6 = 6   #   2 * 6 = 12  #
#   1 * 7 = 7   #   2 * 7 = 14  #
#   1 * 8 = 8   #   2 * 8 = 16  #
#   1 * 9 = 9   #   2 * 9 = 18  #
+#####+#####+
#   3 * 1 = 3   #   4 * 1 = 4   #
#   3 * 2 = 6   #   4 * 2 = 8   #
#   3 * 3 = 9   #   4 * 3 = 12  #
#   3 * 4 = 12  #   4 * 4 = 16  #
#   3 * 5 = 15  #   4 * 5 = 20  #
```



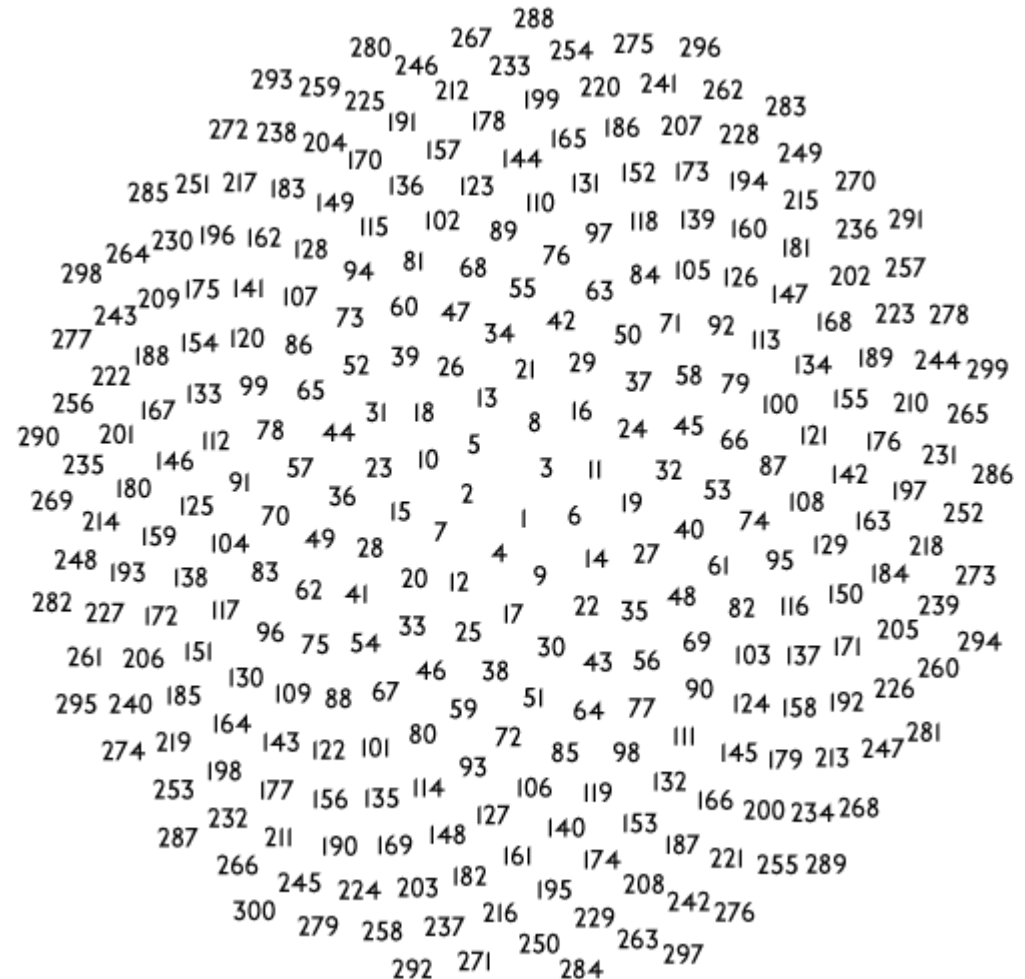
Тема: Циклы в Python. While & for.

Задание 7

Напечатать все чётные числа от 100 до 1.

*

Напечатать все нечётные числа от 100 до 1, среди которых точно не будет чисел от 40 до 60





Тема: Циклы в Python. While & for.

Задание 8

Во многих задачах, связанных с компьютерами, особенно близких к аппаратной части, важную роль играют числа, являющиеся степенями двойки: 1, 2, 4, 8 и так далее. Напишите программу, которая проверяет, является ли введённое натуральное число степенью двойки. Если да, то выводится сама эта степень; если нет, выводится «НЕТ».

1 267 650 600 228 229 401 496 703 205 376

