

# Программирование на Python

Презентация занятия

## **Изучение возможностей и синтаксиса Python: Ветвление и исключения.**

3 занятие



**инжинириум**<sup>®</sup>

МГТУ им. Н.Э. Баумана

2019

## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

В Python условный оператор имеет вид:

if <логическое выражение>:

<действия, выполняемые, когда логическое  
выражение принимает значение True >

else:

<действия, выполняемые, когда логическое  
выражение принимает значение False >

```
a = int(input('Введите первое число: '))
```

```
b = int(input('Введите второе число: '))
```

```
if a % b == 0: # Делится ли первое введённое число на второе нацело  
    print(" Yes")
```

```
else:
```

```
    print("No")
```



## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

Если нужно последовательно проверить несколько условий, используется форма с дополнительным оператором `elif`

`if` <логическое выражение>:

<действия, выполняемые, если логическое выражение принимает значение `True` >

`elif` <второе логическое выражение>:

<действия, выполняемые, если второе логическое выражение принимает значение `True` >

`elif` <третье логическое выражение>:

<действия, выполняемые, если третье логическое выражение принимает значение `True` >

...

`else`:

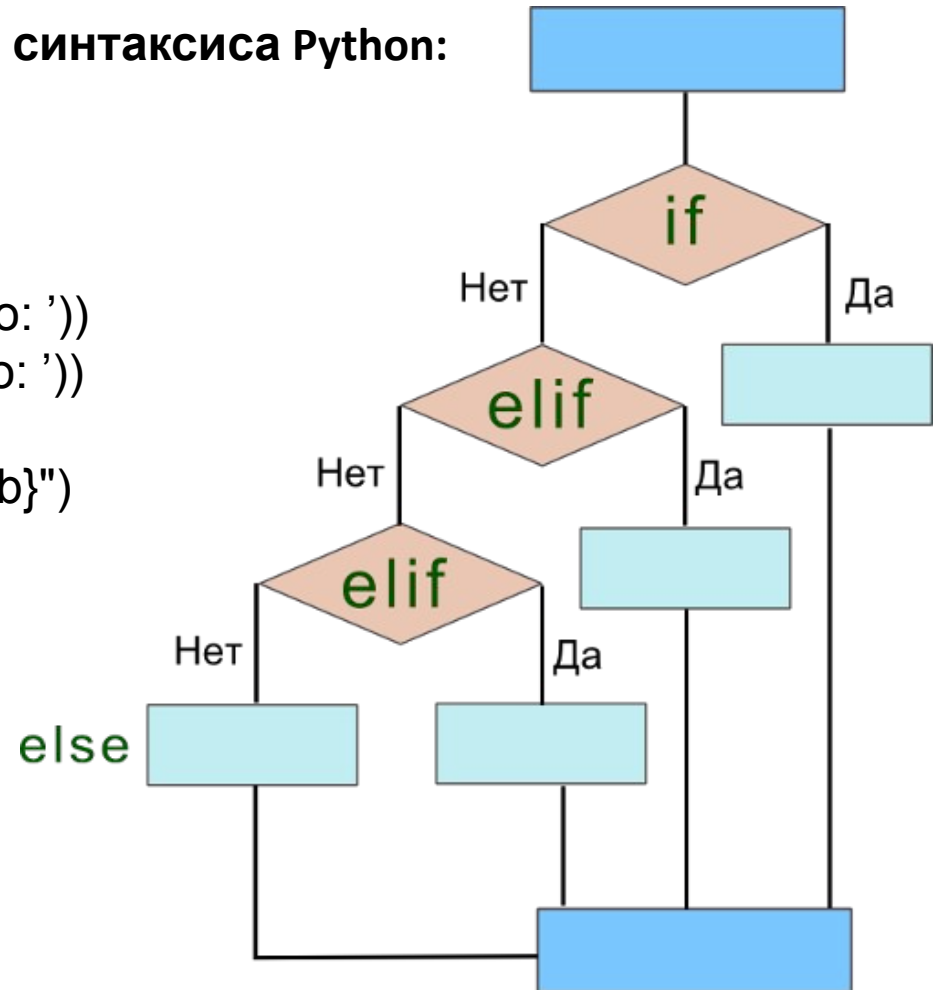
<действия, выполняемые, если ни одно из логических выражений не принимает значение `True` >



## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

Пример кода с if-elif-else

```
a = int(input('Введите первое число: '))
b = int(input('Введите второе число: '))
if a % b == 0:
    print(f"{a} делится нацело на {b}")
elif a % 2 == 0:
    print(f"{a} четное")
elif b % 2 == 0:
    print(f"{b} четное")
elif a % 2 == 1:
    print(f"{a} нечетное")
else:
    print(f"{b} нечетное")
```





## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

Блок — это группа команд

```
age = int(input('Введите сколько лет:'))  
if age > 20:  
    print('Как-то вы староваты!')  
    print('Что вы здесь делаете?')  
    print('Почему не стрижете газон  
или не перекладываете бумажки?')  
  
age = int(input('Введите сколько лет:'))  
if age > 20:  
    # 4 пробела  
    print('Как-то вы староваты!')  
    # 6 пробелов  
    print('Что вы здесь делаете?')
```

**SyntaxError: unexpected indent**

строка кода  
строка кода  
строка кода

Блок 1

строка кода  
строка кода  
строка кода

Блок 2

строка кода  
строка кода  
строка кода

Блок 3

строка кода

строка кода  
строка кода



## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

try:

<Сначала выполняются эти действия>

except <name1>:

<действия, выполняемые, если в блоке try возникло исключение  
name1>

except (name2, name3):

<действия, выполняемые, если в блоке try возникло любое из этих  
исключений>

except <name4> as <data>:

<действия, выполняемые, если в блоке try  
возникло исключение name4, и получает экземпляр исключения>

except:

<действия, выполняемые для остальных возникших исключений>

finally:

<действия, выполняемые после того, как все исключения будут  
обработаны>



## Тема: Изучение возможностей и синтаксиса Python: Ветвление и исключения.

Пример обработки исключений

```
total_count = 100_000
```

```
try:
```

```
    raw = input("введите число: ")
```

```
    number = int(raw)
```

```
    total_count = total_count / number
```

```
    print(total_count + ' - это число')
```

```
except ValueError:
```

```
    # Возникнет, если ввести не число
```

```
    print("некорректное значение!")
```

```
except (TypeError, ZeroDivisionError):
```

```
    # Возникнет, если ввести ноль и при достижении строки с print
```

```
    print("некорректное значение!")
```

```
finally:
```

```
    print("Обработали исключения")
```

