



# Истинность значения объекта Python



mango-office.ru

## Виртуальная АТС Расширенная

До 120 функций. Когда нужно распределять звонки по группам сотрудников и по типам...

Узнать больше

РЕКЛАМА

[Справочник по языку Python3.](#) / Истинность значения объекта Python

Любой объект может быть [проверен на значение истинности](#), для использования в [if/else](#) или [while](#) конструкциях или в качестве операнда [булевых операций](#).

По умолчанию объект считается истинным, если его класс НЕ определяет либо метод `__bool__()`, возвращающий `False`, либо метод `__len__()`, возвращающий ноль.

Вот большинство встроенных объектов, которые считаются ложными:

- константы, определенные как ложные: `None` и `False`.
- ноль любого числового типа: `0`, `0.0`, `0j`, `Decimal(0)`, `Fraction(0, 1)`
- пустые последовательности и коллекции: `''`, `()`, `[]`, `{}`, `set()`, `range(0)`

Операции и [встроенные функции](#), которые возвращают логический результат, для ложного значения всегда возвращают `0` либо `False` и для истинного значения `1` либо `True`, если не указано иное.

Истинность значения объекта может быть вычислена с помощью встроенной функции [bool\(\)](#).

Важное исключение: [логические операции `or` и `and`](#) всегда возвращают один из своих операндов!

## Примеры проверки истинности в Python.

- [Проверка истинности функцией `bool`](#);
- [Проверка в конструкции `if/else`](#);
- [Проверка истинности в цикле по условию `while`](#);
- [Пример с логическими операторами `or` и `and`](#).

## Проверка истинности функцией `bool`.

```
# ЧИСЛО
>>> a = 0
>>> bool(a)
# False
>>> a = 1
>>> bool(a)
# True
>>> a = 5
>>> bool(a)
# True

# СТРОКА
>>> a = ''
>>> bool(a)
# False
>>> a = 'строка'
>>> bool(a)
# True

# СПИСОК
>>> a = []
>>> bool(a)
# False
```

```
>>> a = [0, 1]
>>> bool(a)
# True
```

Смотрите подробнее о встроенной функции [bool](#).

## Проверка в конструкции if/else.

Проверка строки на истинность:

```
>>> a = ''
>>> if a: print('НЕ пустая строка')
>>> else: print('Пустая строка')
# Пустая строка

>>> a = 'строка'
>>> if a: print('НЕ пустая строка')
>>> else: print('Пустая строка')
# НЕ пустая строка
```

Проверка истинности последовательности на примере списка:

```
# список
>>> a = []
>>> if a: print('НЕ пустой список')
>>> else: print('Пустой список')
# Пустой список

>>> a = [0]
>>> if a: print('НЕ пустой список')
>>> else: print('Пустой список')
# НЕ пустой список
```

Смотрите подробнее о конструкции [if/else](#).

## Проверка истинности в цикле по условию while.

```
>>> a = 5
# цикл while будет выполняться
# пока `a` будет истинно
>>> while a:
...     print(f'a = {a}')
...     a = a - 1
... else:
...     print(f'Число a = {a}, а 0 подразумевает `False`')

# a = 5
# a = 4
# a = 3
# a = 2
# a = 1
# Число a = 0, а 0 подразумевает `False`
```

Смотрите подробнее о цикле по условию [while](#).

## Пример с логическими операторами or и and.

Логические операции [or и and](#) всегда возвращают один из своих операндов!

Пример с and:

```
a = 'a'
b = 'b'
>>> a and b
# 'b'
>>> '' and b
# ''
```

Пример с or:

```
a = 'a'
b = 'b'
>>> a or b
# 'a'
>>> '' or b
# 'b'
```

Более подробно в материале ["Логические операции and, or и not"](#).

ХОЧУ ПОМОЧЬ  
ПРОЕКТУ

РЕКЛАМА • 18+

practicum.yandex.ru

**Бесплатное  
занятие  
английским  
в Яндекс  
Практикуме**

Полноценное занятие  
с преподавателем, а не  
презентация курсов

Устный тест на уровень  
языка >

Практика английского >

Узнать больше