



Главная / Инструкции / Python / Логические выражения в Python 3

Бесплатная миграция IT-инфраструктуры в облако

Оставить заявку

Логические выражения и операторы в Python 3



Команда Timeweb Cloud

Наши инженеры, технические писатели, редакторы и маркетологи

Логические выражения

Сравнение строк

Сравнение других типов значений

Преобразование значений в тип bool

Логические операторы

Другие виды логических операторов

Заключение

Если пользователь на сайте, мы запрашиваем у него логин и пароль. Если логин существует в базе, то мы авторизуем пользователя. Иначе продолжаем запрашивать у пользователя логин. Ниже мы сравним значения одного типа друг с другом на языке программирования Python.

Сравнение в Python:

Популярные темы

Python Linux Облако Серверы

Ubuntu Сети и безопасность

Все темы >

Похожие инструкции

Цикл while в Python: руководство

Деплой Python-приложения на сервер

Индексация и разделение строк в Python 3

Использование функций в Python

Введите пароль: 123

Пароль неверный

Введите пароль: qwerty

Вы авторизованы

Логическое сравнение или его результат используется в конструкциях if-else и циклах while.

Логические выражения

Логическое выражение в самом простом смысле - это математическое сравнение. В Python вопрос "Что больше: 3 или 5?" будет записан в следующем виде: $3 > 5$ и звучать как: "Три больше пяти. Правда? - Нет."

```
print(3 > 5) #False
```

Или такой вариант: $4 \neq 0$ (4 не равно 0. Правда?)

```
print(4 != 0) #True
```

Как записываются операторы сравнения в Python:

- $a == b$ - А равно Б
- $a > b$ - А больше Б
- $a \geq b$ - А больше или равно Б



• `a <= b` - А меньше или равно Б

Бесплатная миграция IT-инфраструктуры в облако

• `a != b` - А не равно Б

```
a = 5
```



Логические выражения

Сравнение строк

Сравнение других типов значений

Преобразование значений в тип bool

Логические операторы

Другие виды логических операторов

Заключение

Значение можно присваивать в переменную, как и любое другое значение:

Популярные темы



Python Linux Облако Серверы

Ubuntu Сети и безопасность

Все темы >

Похожие инструкции

Цикл while в Python: руководство

Сделайте все по порядку:

Деплой Python-приложения на сервер



Индексация и разделение строк в Python 3

Использование функций в Python

```
expr = 0 < a < 10
```

```
print(expr) #False
```

В итоге в первом значении мы получили True, а во втором Python выдал False.

Сравнение строк



Помимо чисел, в Python можно сравнивать друг с другом *строки*, *списки* и другие типы данных.

Вот, например, как выглядит сравнение двух строк в Python:

```
s1 = 'python'
s2 = 'python'
s3 = 'Python' # Поменяем одну букву на заглавную
```

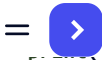


```
print(s1 == s2) #True
print(s1 == s3) #False
```

Во втором случае результатом оказалась ложь, т.к. сравнение чувствительно к регистру символов.

Также строки можно проверять между собой на большинство/меньшинство.





print('abc' > 'x') #False

Бесплатная миграция IT-инфраструктуры в облако

- В первом случае всё просто - строки равны.
- Во втором: происходит посимвольное сравнение строк. Первые два символа равны, сравниваются следующие.
алфавите по порядку после 'a'. Поэтому всё выражение истинно. Чтобы проще было понять, цковые номера в алфавите: 123 > 111
- Сравнение строк
г длины строк, поэтому в третьем случае выражение ложно: 'a' > 'x'
- Сравнение других типов значений
- Преобразование значений в тип bool
- Логические операторы
- Другие виды логических операторов ическим: [wikipedia](#).



Заклучение

Облачные серверы

Популярные темы

Масштабируемые вычислительные ресурсы по всему миру с почасовой оплатой

Python Linux Облако Серверы

[Создать аккаунт](#) юпасность

Все темы >

Похожие инструкции

Цикл while в Python: руководство

Деплой Python-приложения на сервер

Индексация и разделение строк в Python 3

Использование функций в Python

Сравнение значений



ю строк. Например, когда условие в Python не равно. Каждый элемент списка по очереди



Значения разных типов можно также сравнивать между собой:

```
expr1 = '1' == 1
expr2 = '' == 0

print(expr1) #False
print(expr2) #False
```



Результат ложный, т.к. типы данных не равны. (Строка и число).

Проверка на большинство/меньшинство разных типов выдаст ошибку:

```
print('a' > 1)
TypeError: '>' not supported between instances of 'str' and 'int'

print((1, 1, 1) < [1, 2, 3])
TypeError: '<' not supported between instances of 'tuple' and 'list'
```



Преобразование значений в тип bool



Значения False или True можно получить непосредственно из самих значений, без операции сравнения. Например, строка 'a' будет True, а 0 - False. Для этого нужно вызвать функцию bool() и передать ей параметр, который мы хотим преобразовать:



Бесплатная миграция IT-инфраструктуры в облако

```
print(var1)    #True
print(var2)    #False
```

А что такое значения False:

Логические выражения

Сравнение строк

Сравнение других типов значений

Преобразование значений в тип bool

Логические операторы

Другие виды логических операторов

Заключение

строки равен нулю, то переменная bool в Python будет равна False .

L

Популярные темы

Python Linux Облако Серверы

Ubuntu Сети и безопасность

ли с длиной больше 0 будут True.

нужны, когда проверяется наличие какой-либо информации в переменной, например, поля в форме регистрации или оставил их пустыми.

Все темы >

У,

Похожие инструкции

Цикл while в Python: руководство

Деплой Python-приложения на сервер

Индексация и разделение строк в Python 3

Использование функций в Python

Это обязательное поле

Введите Ваше имя: Иван

Добро пожаловать, Иван!

i: '')

.name}!')

Логические операторы



Если мы хотим проверить на истинность сразу несколько логических выражений или значений, то нам нужно использовать в **Python 3** **логические операторы**.

Например ввел ли пользователь своё имя и больше ли 12 лет его возраст:

```
name = 'Иван'
age = 10

expr = name and age > 12

print(expr)    #False
```

Мы использовали логический оператор and. Он возвращает истинное значение, только тогда, когда по обе стороны от него находятся истинные значения.

Есть ещё or и not.

- or - Возвращает истинное значение, когда хотя бы с одной стороны от него находится *истинное* значение.
- not - Меняет значение на противоположное.



Бесплатная миграция IT-инфраструктуры в облако

```
print(True and False) # False
print(True or False) # True
print(not True) # False
print(not False) # True
```



Логические выражения

ставятся между собой в одном выражении:

Сравнение строк

Сравнение других типов значений



Преобразование значений в тип bool

Логические операторы

Другие виды логических операторов

Если нужно изменить его, надо использовать скобки, как в математике.

Заключение

Популярные темы



Python Linux Облако Серверы

Ubuntu Сети и безопасность

Все темы >

Сначала считается выражение внутри скобок (False), а затем происходит его инверсия. В Python операторы and и or. Они не обязательно возвращают только True и False. Они возвращают последнее остановившееся вычисление.

Похожие инструкции

Цикл while в Python: руководство

Деплой Python-приложения на сервер



Индексация и разделение строк в Python 3

Использование функций в Python

) и

- В первом случае вернулся 0, потому что расчёты остановились на переменной age. Так как все операторы являются and, то если одно из значений ложно, то и всё выражение будет ложно, следовательно, дальше считать нет смысла.
- Во втором случае обратная ситуация. Все операторы являются or, поэтому вычисления останавливаются на первом истинном значении. Собственно, оно нам и вернулось - значение переменной name1.

Другие виды логических операторов



Существуют также другие логические операции, которые используются реже и в специфических ситуациях.

Например, есть логические побитовые операции, которые применяются в шифровании данных.

Одна из таких - XOR: исключающее ИЛИ. В Python этой операции соответствует символ ^ .

Пример шифрования числа. Вместо чисел могут быть любые бинарные данные.

```
phrase = 2022
key = 101
```



```
# Шифрование используя ключ
encrypt = phrase ^ key
print(encrypt)
```

```
# Расшифровка сообщения используя ключ
decrypt = encrypt ^ key
```



1923
Бесплатная миграция IT-инфраструктуры в облако
2022

Список всех побитовых операторов в Python:

Логические выражения		
Сравнение строк		
Сравнение других типов значений		
Преобразование значений в тип bool		
Логические операторы		
Другие виды логических операторов	b	
Заключение		
Популярные темы		
Python	Linux	Облако
	Серверы	а на b
Ubuntu	Сети и безопасность	я а на b

Все темы >
Операторы, отсутствующие в питоне. Например, в питоне нет логической операции импликация.

- Похожие инструкции
- Цикл while в Python: руководство
- Деплой Python-приложения на сервер
- Индексация и разделение строк в Python 3
- Cloud-15
Использование функций в Python
300 ₽/мес

выгодные тарифы на облачные серверы

emium 3.3 ГГц Standard 2.8 ГГц High CPU 5 ГГц

 Санкт-Петербург 

Процессор	1 x 3.3 ГГц
Память	1 Гб
NVMe	15 Гб
Канал	1 Гбит/с
Публичный IP	

Заказать

Cloud-30
550 ₽/мес

Процессор	1 x 3.3 ГГц
Память	2 Гб
NVMe	30 Гб
Канал	1 Гбит/с
Публичный IP	

Заказать

[Показать еще 6 тарифов](#)

Заключение





Бесплатная миграция IT-инфраструктуры в облако

Кстати, в [официальном канале](#) Timeweb Cloud собрали комьюнити из специалистов, которые говорят про IT-тренды, делятся полезными инструкциями и даже приглашают к себе работать.

Хотите внести свой вклад?

Логические выражения

Участвуйте в нашей контент-программе за

вознаграждение или запросите нужную вам инструкцию

Сравнение строк

Операторы в значений

Написать инструкцию

Преобразование значений в тип bool

[Запросить инструкцию](#)

Другие виды логических операторов

Заключение



Популярные темы

Python Linux Облако Серверы

Ubuntu Сети и безопасность

Все темы >

Ваш комментарий

Оставьте свой комментарий

Войти

Поддержка и рекомендации экспертов

Python 3

Поддерживается разметка Markdown

Использование функций в Python

Пока нет комментариев



< Мы в социальных сетях

[Подписаться на рассылку](#)

timeweb > cloud

Сервисы



Отдел продаж



8 (800) 700-32-92



twcloud

Облачные серверы

VDS и VPS

Выделенные серверы

Облачные базы данных

Kubernetes

Балансировщики

Объектное хранилище S3

Защита от DDoS

Решения

Соответствие 152-ФЗ

Администрирование под запрос

Полное администрирование

Корпоративная почта

Инструменты

API

Бесплатная миграция ИТ-инфраструктуры в облако

Cloud-init

Партнерам

Программа реселлинга

Логические выражения

а

Сравнение строк

Сравнение других типов значений

Преобразование значений в тип bool

Логические операторы

,

Другие виды логических операторов

Заключение

Популярные темы

Python

Linux

Облако

Серверы

Ubuntu

Сети и безопасность

Все темы

- Похожие инструкции

Цикл while в Python: руководство

Деплой Python-приложения на сервер

Индексация и разделение строк в Python 3

Использование функций в Python
- Участник

CNews Analytics

Рейтинг провайдеров

IaaS 2024: 6 место Лучший SLA IaaS 2024: 10 место Рейтинг провайдеров

DBaaS 2023: 6 место

© 2024 ООО «ТАЙМВЭБ.КЛАУД». Все права защищены.

Политика ООО «ТАЙМВЭБ.КЛАУД» в отношении обработки персональных данных.

ПО "Платформа "ТАЙМВЭБ.КЛАУД" включено в Единый реестр российских программ для электронных вычислительных машин и баз данных. Запись в реестре №15725 от 05.12.2022 г.