

# بولین (bool)

صفحه اصلی / دانش‌نامه / پایتون / انواع داده

آخرین به‌روزرسانی: ۱۴۰۵ خرداد

نوع داده بولین (bool) در پایتون نمایانگر مقادیر منطقی (Boolean) است و تنها دو مقدار ممکن دارد: True و False. یکی از مهم‌ترین ویژگی‌های bool در پایتون این است که یک زیرنوع از عدد صحیح به شمار می‌رود؛ یعنی True دقیقاً معادل عدد صحیح 1 و False دقیقاً معادل عدد صحیح 0 است. این طراحی به برنامه‌نویس امکان می‌دهد مقادیر **نوع داده** بولین را مستقیماً در محاسبات عددی استفاده کند. البته مفهوم ارزیابی صحت در پایتون بسیار فراتر از نوع bool است؛ هر شیء پایتون می‌تواند در یک زمینه بولین ارزیابی شود و پایتون قوانین مشخصی تعریف کرده که کدام اشیاء False و کدام True هستند.

## فهرست مطالب:

- ایجاد بولین در پایتون
- بولین به عنوان زیرنوع عدد صحیح
- ارزیابی درستی در پایتون
- عملگرهای بولین در پایتون
- اولویت عملگرهای بولین در پایتون
- عملگرهای مقایسه در پایتون
- عملگرهای هویت در پایتون
- عملگرهای بیتی روی bool در پایتون
- مثال واقعی از کاربرد بولین
- سوالات متداول

## ایجاد بولین در پایتون

داده‌هایی از نوع بولین در پایتون را می‌توان به روش‌های مختلفی ساخت.

(۱) نوشتار مستقیم:

```
>>> x = True
>>> y = False
>>> type(x)
<class 'bool'>
```

۲) نتیجه عملگرهای مقایسه:

```
>>> 5 > 3
True
>>> 5 == 3
False
>>> 10 != 10
False
```

۳) تابع bool: سازنده نوع داده بولین در پایتون، تابع bool است:

```
bool(object=False)
```

مقدار بازگشتی تابع bool همیشه یکی از دو مقدار True یا False است.

## راهنمای جامع تابع bool

در مورد تابع bool که سازنده یا Constructor برای بولین‌ها در پایتون است بیشتر بدانید: [آشنایی با تابع bool در پایتون](#)

**نکته:** True و False کلیدواژه‌های از پیش تعریف شده در پایتون هستند (با حرف بزرگ) و نه رشته متنی. نوشتن true یا false با حرف کوچک خطای NameError می‌دهد.

## بولین به عنوان زیرنوع عدد صحیح

نوع داده bool در پایتون، یک زیرکلاس مستقیم از نوع [عدد صحیح \(int\)](#) است؛ یعنی True دقیقاً معادل عدد صحیح 1 و False دقیقاً معادل عدد صحیح 0 است. این رابطه یکی از جالب‌ترین طراحی‌های پایتون است:

```
>>> isinstance(True, int)
```

```
True
```

```
>>> int(True)
```

```
1
```

```
>>> int(False)
```

```
0
```

```
>>> True + True
```

```
2
```

```
>>> True + False
```

```
1
```

```
>>> True * 5
```

```
5
```

## ارزیابی درستی در پایتون

یکی از قدرتمندترین ویژگی‌های پایتون این است که هر شیء می‌تواند در یک زمینه بولین (مثل شرط `if` یا `while`) ارزیابی شود. پایتون متد `__bool__` و سپس `__len__` شیء را بررسی می‌کند. پایتون هر آبجکتی را `True` می‌داند مگر آنکه یکی از مقادیر `falsy` باشد. این مقادیر عبارتند از:

- ثابت‌های `None` و `False`
- صفر در هر نوع عددی: `0`، `0.0`، `0j`
- دنباله‌ها و مجموعه‌های خالی: `""`، `[]`، `()`، `{}`، `set()`
- آبجکت‌هایی که متد `__bool__` آن‌ها `False` یا متد `__len__` آن‌ها `0` برمی‌گرداند

تمام مقادیر دیگر، از نظر پایتون `True` هستند.

## عملگرهای بولین در پایتون

پایتون سه عملگر بولین دارد. نکته‌ی کلیدی این است که `and` و `or` همیشه یکی از عملوندها را برمی‌گردانند؛ نه لزوماً `True` یا `False`. در ادامه، تعدادی مثال از عملیات روی اعداد صحیح را با هم مرور خواهیم کرد.

**عملگر `not`:** تنها عملگر یک عملوندی (unary) بولین در پایتون است که مقدار بولین معکوس را برمی‌گرداند و همیشه `True` یا `False` است:

```
>>> not False
True
>>> not 0
True
>>> not ""
True
>>> not "hello"
False
>>> not []
True
```

اولویت not از and و or بیشتر است اما از عملگرهای مقایسه کمتر:

```
>>> not 3 > 2          # not (3 > 2)
False
>>> not 3 == 3        # not (3 == 3)
False
```

**عملگر and:** معادل عملگر «و» منطقی در پایتون است؛ اگر عملوند اول False باشد، عملوند دوم هرگز ارزیابی نمی‌شود. مقدار اولین عملوند False یا آخرین عملوند را برمی‌گرداند:

```
>>> True and True
True
>>> True and False
False
>>> False and True
False
>>> False and False
False
>>> 0 and "hello"
0
>>> "a" and "b"
'b'
```

**عملگر or:** معادل عملگر «یا» منطقی در پایتون است؛ اگر عملوند اول True باشد، عملوند دوم هرگز ارزیابی نمی‌شود. مقدار اولین عملوند True یا آخرین عملوند را برمی‌گرداند:

```
>>> True or True
True
>>> True or False
True
>>> False or True
True
>>> False or False
False
>>> 0 or "hello"
'hello'
>>> "" or "default"
'default'
```

## اولویت عملگرهای بولین در پایتون

در پایتون، اولویت از بالا به پایین به صورت `or > and > not` است. یعنی ابتدا `not`، سپس `and` و در نهایت `or` ارزیابی می‌شود.

```
>>> True or False and False           # True or (False and False)
True
>>> (True or False) and False
False

>>> not True or True                   # (not True) or True
True
>>> not (True or True)
False
```

## عملگرهای مقایسه در پایتون

پایتون ۶ عملگر مقایسه دارد که همگی مقدار `bool` برمی‌گردانند:

```
>>> 5 > 3
True
>>> 5 < 3
False
>>> 5 >= 5
True
>>> 5 <= 4
False
>>> 5 == 5
True
>>> 5 != 4
True
```

پایتون از زنجیره‌بندی عملگرهای مقایسه پشتیبانی می‌کند که در ریاضیات رایج است:

```
>>> x = 5
>>> 1 < x < 10          # 1 < x and x < 10
True
>>> 1 < x < 3
False
>>> 0 <= x <= 10
True
>>> 1 < 2 < 3 < 4
True
```

## عملگرهای هویت در پایتون

عملگر `is` در پایتون هویت شیء را بررسی می‌کند؛ دقت کنید که هنگام استفاده از `is` پایتون بررسی می‌کند که آیا دو متغیر به همان شیء در حافظه اشاره می‌کنند یا خیر. این بررسی صرفاً بر اساس مقادیر متغیرها نیست:

```
>>> a = [1, 2, 3]
>>> b = [1, 2, 3]
>>> a == b
True
>>> a is b
False
```

در مثال بالا مشاهده می‌کنید که مقادیر ۲ متغیر با هم برابرند ولی به دو آبجکت متفاوت در حافظه اشاره دارند. عملگر `is not` دقیقاً رفتاری برعکس عملگر `is` دارد.

**نکته طلایی:** برای مقایسه با `None`، همیشه از عملگرهای `is` یا `is not` استفاده کنید، نه از `==`!

## عملگرهای بیتی روی `bool` در پایتون

چون `bool` در پایتون یک زیرنوع از `int` است، تمام عملگرهای بیتی روی آن کار می‌کنند:

```
>>> True & True          # bitwise AND
True
>>> True | False       # bitwise OR
True
>>> True ^ True        # bitwise XOR
False
>>> ~True              # bitwise NOT
-2
>>> ~False
-1
```

**تذکر مهم:** از عملگرهای بیتی `&` و `|` به جای `and` و `or` در شرطهای بولین استفاده نکنید. عملگرهای `&` و `|` اتصال کوتاه ندارند و هر دو طرف را همیشه ارزیابی می‌کنند. این عملگرها برای عملیات بیتی روی اعداد طراحی شده‌اند، نه منطق بولین.

## مثال واقعی از کاربرد بولین

در یک سناریوی فرضی، شما سیستم اعتبارسنجی فرم ثبت‌نام کاربر می‌نویسید. این سیستم از ارزیابی صحت پایتون، عملگرهای بولین و متد `__bool__` برای کنترل جریان استفاده می‌کند. در این مثال از `__bool__` برای

تعریف صحت یک شیء سفارشی، عملگر and با اتصال کوتاه برای ترکیب نتایج، عملگر not برای بررسی ساده شکست اعتبارسنجی و از ترکیب تابع any با regex برای بررسی ویژگی‌های رمز عبور استفاده شد.

کدهای این مثال به همراه خروجی نهایی را می‌توانید از [اینجا](#) دانلود کنید.

وبسایت شخصی رضا قلعه‌خانی

## 1. چرا ("False") مقدار True برمی‌گرداند؟

تابع bool رشته‌های متنی را بر اساس خالی یا غیرخالی بودن ارزیابی می‌کند، نه محتوای آن‌ها. استرینگ "False" پنج کاراکتر دارد پس غیرخالی است و True برمی‌گردد.

## 2. تفاوت عملگرهای and/or با & | چیست؟

عملگرهای and و or عملگرهای منطقی هستند: اتصال کوتاه دارند (ممکن است عملوند دوم هرگز ارزیابی نشود)، یکی از عملوندها را برمی‌گرداند (نه لزوماً True یا False) و با هر شیء پایتون کار می‌کنند. اما عملگرهای & و | عملگرهای بیتی هستند: هیچ‌وقت اتصال کوتاه ندارند، هر دو طرف همیشه ارزیابی می‌شوند و روی int عملیات بیتی انجام می‌دهند.

## 3. آیا می‌توانم bool را زیرکلاس کنم؟

از نظر فنی بله، اما به شدت توصیه نمی‌شود. bool یک کلاس نهایی نیست ولی طراحی پایتون فرض می‌کند که تنها دو نمونه بولین وجود دارند: True و False. ساخت زیرکلاس می‌تواند رفتارهای غیرمنتظره ایجاد کند. به جای زیرکلاس کردن bool، متد `__bool__` را در کلاس خود تعریف کنید.

جهت کسب اطلاعات بیشتر می‌توانید به [مستندات رسمی پایتون برای نوع داده بولین \(bool\)](#) مراجعه کنید.

دسته: انواع داده - پایتون - دانش‌نامه برنامه‌نویسی