تابع int

آخرین به‌روزرسانی: ۲۶ اردیبهشت ۱۴۰۵

تابع int در پایتون یک تابع پیش‌ساخته (built-in) است که یک عدد یا استرینگ را به عدد صحیح تبدیل می‌کند. این تابع یکی از پرکاربردترین توابع پایتون در پردازش و تبدیل داده‌هاست و امکان تبدیل اعداد اعشاری، استرینگ‌های عددی و حتی اعداد در مبناهای مختلف (مانند مبنای ۲، ۸ و ۱۶) به عدد صحیح پایه ۱۰ را فراهم می‌کند.

فهرست مطالب:

سینتکس تابع int

سینتکس کلی تابع int به دو صورت زیر است:

```
int(x)
int(x, base=10)
```

پارامترها یا آرگومان‌های این سینتکس عبارتند از:

- آرگومان x مقداری است که می‌خواهیم به عدد صحیح تبدیل شود. می‌تواند یک عدد اعشاری، یک عدد صحیح دیگر، یا یک استرینگ باشد. اگر آرگومانی وارد نشود، مقدار پیش‌فرض 0 برگردانده می‌شود.

- آرگومان اختیاری `base` مبنای عددی استرینگ ورودی را مشخص می‌کند. مقدار پیش‌فرض آن 10 (مبنای ده) است. این آرگومان فقط زمانی معتبر است که x یک استرینگ یا از نوع bytes یا bytearray باشد. مقادیر قابل قبول برای این پارامتر بین ۲ تا ۳۶ هستند، علاوه بر مقدار ویژه 0 وجود دارد که مبنا را از پیشوند استرینگ تشخیص می‌دهد.

مقدار بازگشتی تابع `int` همیشه یک عدد صحیح از نوع `int` است.

راهنمای جامع عدد صحیح (int)

در مورد نوع داده عدد صحیح یا همان `int` در پایتون بیشتر بدانید: **عدد صحیح (int)**

کاربردهای تابع int

ساده‌ترین مثال از کاربرد این تابع می‌تواند حالت زیر باشد:

```
>>> int(123.45)
123
>>> int('123')
123
>>> int()
0
```

یا می‌توان استرینگ‌هایی در مبناهای مختلف را به عدد صحیح مبنای ۱۰ تبدیل کرد:

```
>>> int('FACE', 16)
64206
>>> int('0xface', 0)
64206
>>> int('01110011', base=2)
115
```

در صورتی که base=0 باشد، پایتون مبنا را از پیشوند استرینگ تشخیص می‌دهد: 0x برای مبنای ۱۶، 0o برای مبنای ۸ و 0b برای مبنای ۲.

```
# convert base 2 to base 10
>>> int('1010', 2)
10
>>> int('0b1010', 0)
10

# convert base 8 to base 10
>>> int('17', 8)
15
>>> int('0o17', 0)
15

# convert base 16 to base 10
>>> int('FF', 16)
255
>>> int('0xFF', 0)
255
```

رایج‌ترین کاربردهای تابع `int` عبارتند از:

- تبدیل ورودی کاربر (که همیشه به صورت استرینگ دریافت می‌شود) به عدد صحیح،
- حذف بخش اعشاری اعداد و تبدیل آن‌ها به عدد صحیح،
- تبدیل اعداد در مبناهای ۲، ۸ و ۱۶ به مبنای ۱۰،
- اعتبارسنجی و پاکسازی داده‌های عددی ورودی.

مثال واقعی از تابع `int`

در یک سناریوی فرضی، شما یک برنامه ساده برای محاسبه فاکتور فروشگاه دارید. کاربر تعداد محصولات و قیمت هر کدام را وارد می‌کند و برنامه باید مبلغ نهایی را حساب کند. از آنجا که تابع `input` همیشه استرینگ برمی‌گرداند، باید از `int` برای تبدیل ورودی استفاده کنید:

```
def calculate_invoice():
    try:
        quantity = int(input("Enter number of product: "))
        price = int(input("Enter price per unit: "))
        discount = int(input("Enter discount amount: "))

        total = quantity * price
        discount_amount = total * discount // 100
        final_price = total - discount_amount

        print(f"Total Price: {total:,} Rials")
        print(f"Discount Amount: {discount_amount:,} Rials")
        print(f"Final Price: {final_price:,} Rials")

    except ValueError:
        print("Error: Just integer values are valid.")

calculate_invoice()
```

نمونه خروجی این کد به صورت زیر خواهد بود:

```
Enter number of product: 3
Enter price per unit: 250000
Enter discount amount: 10

Total Price: 750,000 Rials
Discount Amount: 75,000 Rials
Final Price: 675,000 Rials
```

در این مثال، تابع `int` ورودی‌های متنی کاربر را به عدد صحیح تبدیل می‌کند و بلوک `try/except` خطای احتمالی ورودی نامعتبر را مدیریت می‌کند.

تفاوت تابع `int` با `round` و `math.floor`

هر سه تابع می‌توانند عدد اعشاری را به صحیح تبدیل کنند، اما رفتار آن‌ها متفاوت است:

```
import math

x = 9.7

print(int(x))           # output: 9
print(round(x))         # output: 10
print(math.floor(x))    # output: 9
print(math.ceil(x))     # output: 10

y = -9.7

print(int(y))           # output: -9
print(math.floor(y))    # output: -10
```

انتخاب بین این توابع بستگی به نیاز شما دارد: اگر می‌خواهید فقط بخش اعشاری را حذف کنید، از `int` استفاده کنید. اگر به گرد کردن نیاز دارید، از `round` یا توابع ماژول `math` کمک بگیرید.

سوالات متداول

1. تفاوت توابع `int` و `round` در پایتون چیست؟

تابع `int` بخش اعشاری را بدون هیچ گرد کردنی حذف می‌کند؛ یعنی همیشه به سمت صفر می‌رود. تابع `round` به نزدیک‌ترین عدد صحیح گرد می‌کند. برای مثال `int(9.9)` مقدار 9 و `round(9.9)` مقدار 10 برمی‌گرداند.

---

2. چرا `int('3.14')` خطا می‌دهد؟

تابع `int` نمی‌تواند استرینگی را که نمایانگر عدد اعشاری است مستقیماً تبدیل کند. برای این کار ابتدا باید استرینگ را با تابع `float` تبدیل کنید، سپس به تابع `int` بدهید.

---

3. آیا تابع `int` با اعداد بسیار بزرگ هم کار می‌کند؟

بله. برخلاف بسیاری از زبان‌های برنامه‌نویسی، پایتون محدودیتی در اندازه اعداد صحیح ندارد و `int` می‌تواند اعداد دلخواه بزرگ را ذخیره و پردازش کند.

---

4. پارامتر `base=0` دقیقاً چه کاری می‌کند؟

وقتی `base=0` باشد، پایتون مبنای عدد را از پیشوند استرینگ تشخیص می‌دهد: `0b` یا `0B` برای مبنای ۲، `0o` یا `0O` برای مبنای ۸، `0x` یا `0X` برای مبنای ۱۶ و بدون پیشوند برای مبنای ۱۰.

---

5. اگر تابع `int` را بدون آرگومان فراخوانی کنم چه اتفاقی می‌افتد؟

این تابع بدون آرگومان مقدار 0 را برمی‌گرداند. این رفتار گاهی برای مقداردهی اولیه متغیرهای شمارنده استفاده می‌شود.

جهت کسب اطلاعات بیشتر می‌توانید به **مستندات رسمی پایتون برای تابع `int`** مراجعه کنید.
