تابع float

آخرین به‌روزرسانی: ۰۱ خرداد ۱۴۰۵

تابع `float` در پایتون یک **تابع پیش‌ساخته** (`built-in`) است که یک عدد یا رشته متنی را به عدد اعشاری تبدیل می‌کند. این تابع بر اساس استاندارد `IEEE-754` عمل می‌کند و علاوه بر اعداد معمولی، از مقادیر ویژه‌ای مانند بی‌نهایت (`inf`) و عدد غیرمعتبر (`NaN`) نیز پشتیبانی می‌کند. `float` یکی از توابع پرکاربرد پایتون در پردازش داده‌های عددی، محاسبات علمی و تبدیل ورودی‌های متنی است.

فهرست مطالب:

سینتکس تابع `float`

سینتکس کلی تابع `float` به صورت زیر است:

```
float(x=0.0)
```

پارامترها یا آرگومان‌های این سینتکس عبارتند از:

- آرگومان اختیاری `x` مقداری است که می‌خواهیم به عدد اعشاری تبدیل شود. می‌تواند یک عدد صحیح، یک عدد اعشاری دیگر یا یک رشته متنی باشد که نمایش معتبری از یک عدد یا مقادیر ویژه "`inf`" و "`nan`" داشته باشد. همچنین هر آبجکتی که متد `__float__` یا `__index__` را پیاده‌سازی کرده باشد نیز قابل قبول است. اگر آرگومانی وارد نشود، مقدار پیش‌فرض 0.0 برگردانده می‌شود.

مقدار بازگشتی تابع float همیشه یک عدد اعشاری از نوع float است.

راهنمای جامع عدد اعشاری (float)

در مورد نوع داده عدد اعشاری یا همان float در پایتون بیشتر بدانید: **عدد اعشاری (float)**

نکته: در صورتی که رشته ورودی نمایش معتبری از عدد نباشد، خطای ValueError ایجاد می‌شود. اگر عدد ورودی بیش از حد بزرگ باشد و از محدوده float خارج شود، خطای OverflowError رخ می‌دهد.

کاربردهای تابع float

ساده‌ترین مثال از کاربرد این تابع می‌تواند حالت زیر باشد:

```
>>> float()
0.0
>>> float(42)
42.0
>>> float(3.14)
3.14
>>> float("19.99")
19.99
```

یا می‌توان مقادیر ویژه بی‌نهایت و NaN را از رشته ساخت:

```
>>> float("inf")
inf
>>> float("-inf")
-inf
>>> float("nan")
nan
```

فاصله‌های ابتدا و انتهای استرینگ به طور خودکار نادیده گرفته می‌شوند و پردازش استرینگ‌های ویژه بدون حساسیت به بزرگی و کوچکی حروف انجام می‌شود:

```
>>> float("  3.14  ")
3.14
>>> float("INF")
inf
>>> float("Infinity")
inf
>>> float("NaN")
nan
```

رایج‌ترین کاربردهای تابع `float` عبارتند از:

- تبدیل ورودی کاربر یا داده‌های خوانده‌شده از فایل به عدد اعشاری،
- تبدیل اعداد صحیح به اعشاری برای انجام محاسباتی که دقت اعشاری نیاز دارند،
- ساخت مقادیر ویژه `inf` و `nan` برای محاسبات علمی،
- اطمینان از اینکه نتیجه یک تقسیم یا محاسبه اعشاری خواهد بود.

تابع `float` رشته‌هایی را که نمایانگر اعداد صحیح یا اعشاری هستند می‌پذیرد. نماد علمی (`scientific notation`) نیز پشتیبانی می‌شود:

```
>>> float("3.14")
3.14
>>> float("-1.5")
-1.5
>>> float("+99")
99.0
>>> float("1.5e-4")
0.00015
>>> float("2.5E2")
250.0
```

اما رشته‌هایی که جداکننده هزارگان یا کاراکترهای غیرعددی دارند، خطای `ValueError` ایجاد می‌کنند:

```
>>> float("1,234.56")
ValueError: could not convert string to float: '1,234.56'

>>> float("سه و نیم")
ValueError: could not convert string to float: 'سه و نیم'

>>> float("")
ValueError: could not convert string to float: ''
```

سفارشی‌سازی رفتار تابع `float`: وقتی این تابع روی یک آبجکت از کلاس دلخواه فراخوانی می‌شود، پایتون ابتدا متد `__float__` و در صورت نبود آن، متد `__index__` آن کلاس را صدا می‌زند. با تعریف متد جادویی `__float__` می‌توانید نحوه تبدیل آبجکت‌های دلخواه به `float` را کنترل کنید:

```
class Temperature:
    def __init__(self, celsius):
        self.celsius = celsius

    def __float__(self):
        return float(self.celsius * 9 / 5 + 32)

    def __repr__(self):
        return f"{self.celsius}°C"


temp = Temperature(100)
print(float(temp))              # output: 212.0
print(float(Temperature(0)))    # output: 32.0
print(float(Temperature(37)))   # output: 98.6
```

در این مثال، رفتار متد `float` آبجکت‌های کلاس `Temperature` سفارشی‌سازی شده‌اند؛ به نحوی که برای تبدیل دمای سلسیوس به دمای فارنهایت استفاده شده‌اند.

مثال واقعی از تابع float

در یک سناریوی فرضی، شما داده‌های فروش ماهانه را از یک فایل CSV می‌خوانید. مقادیر ستون قیمت به صورت رشته متنی ذخیره شده‌اند و باید با مدیریت خطای مناسب به float تبدیل شوند:

```
raw_sales = [
    {"product": "Laptop", "price": "45000000", "discount": "10.5"},
    {"product": "Mouse", "price": "850000", "discount": "5"},
    {"product": "Keyboard", "price": "N/A", "discount": "0"},
    {"product": "Headphone", "price": "3200000", "discount": "15.75"},
]

def parse_price(value, default=0.0):
    try:
        return float(value)
    except (ValueError, TypeError):
        return default

total_revenue = 0.0
for item in raw_sales:
    price = parse_price(item["price"])
    discount = parse_price(item["discount"])
    final_price = price * (1 - discount / 100)
    total_revenue += final_price
    print(f"{item['product']}: {final_price:,.0f} Rials")

print(f"\nTotol Income: {total_revenue:,.0f} Rials")
```

نمونه خروجی این کد به صورت زیر خواهد بود:

```
Laptop: 40,275,000 Rials
Mouse: 807,500 Rials
Keyboard: 0 Rials
Headphone: 2,696,000 Rials

Totol Income: 43,778,500 Rials
```

در این مثال، تابع `parse_price` از یک بلوک `try/except` برای مدیریت مقادیر غیرعددی مثل "N/A" استفاده می‌کند و به جای توقف برنامه، مقدار پیش‌فرض 0.0 را جایگزین می‌کند.

تفاوت تابع `float` با `int` و `round`

هر سه تابع می‌توانند در تبدیل داده‌های عددی استفاده شوند، اما رفتار متفاوتی دارند:

```
>>> float(7)
7.0
>>> int(7.9)
7
>>> round(7.9)
8
>>> round(7.555, 2)
7.56
```

به صورت کلی:

- تابع `float` یک نقطه اعشاری به عدد صحیح ورودی اضافه می‌کند.
- تابع `int` بخش اعشاری عدد را حذف می‌کند و فقط بخش صحیح را نگه می‌دارد.
- تابع `round` برای گرد کردن اعداد استفاده می‌شود.

سوالات متداول

1. تفاوت `float(x)` و `x * 1.0` چیست؟

هر دو نتیجه یکسانی برای اعداد دارند، اما تابع `float` صریح‌تر و خواناتر است و می‌تواند استرینگ‌ها را هم تبدیل کند؛ اما `x * 1.0` فقط روی اعداد کار می‌کند و برای رشته‌های متنی، خطای `TypeError` می‌دهد.

---

2. آیا تابع `float` از نماد علمی پشتیبانی می‌کند؟

بله. در نماد علمی ورودی به تابع، هم `e` کوچک و هم `E` بزرگ معتبر هستند.

---

3. اگر تابع `float` را بدون آرگومان فراخوانی کنم چه اتفاقی می‌افتد؟

این تابع بدون آرگومان مقدار 0.0 برمی‌گرداند. این رفتار مشابه مقداردهی اولیه یک متغیر عددی اعشاری با صفر است.

---

4. چرا عدد با جداکننده هزارگان در تابع `float` خطا می‌دهد؟

این تابع جداکننده هزارگان (کاما) را به عنوان بخشی از عدد نمی‌شناسد. برای تبدیل چنین رشته‌هایی ابتدا باید کاما را با متدهای مربوط به استرینگ‌ها حذف کنید.

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

دسته: توابع پیش‌ساخته - پایتون - دانش‌نامه برنامه‌نویسی