禪

کدنویسی منطقی و درست

اگر برنامهنویس پایتون هستید از این دو اشتباه دوری کنید

همه ما در طول زندگی و دوره کاری خود مرتکب اشتباه میشویم. اشتباه کردن جزء ثابت یادگیری است. از راه رفتن گرفته تا یادگیری یک زبان برنامهنویسی جدید همچون پایتون، ما همواره اشتباهات مختلفی را مرتکب میشویم که این اشتباهات به ما در یادگیری بهتر کمک میکنند.

با توجه به اینکه در چند سال اخیر دانشگاهها و برنامهنویسان ایرانی به شکل جدی پایتون را مورد توجه قرار دادهاند انتظار میرود در چند سال آینده بر تعداد برنامهنویسان داخلی این زبان افزوده شود. اما اگر تازه به دنیای برنامهنویسی پایتون قدم گذاشتهاید، پیشنهاد میکنیم این مطلب را مطالعه کنید تا اشتباهاتی که در ادامه مشاهده خواهید کرد در کار شما وقفه ایجاد نکنند.

مطلب پیشنهادی

دروس دانشگاهی

چرا تدریس زبان پایتون در دانشگاهها محبوب شده است؟

بهکارگیری متغیرهای تغییرپذیر به عنوان پارامترهای پیشفرض در زمان تعریف یک تابع

شما تابعی در اختیار دارید که به منظور جستوجوی لینکهای روی یک صفحه جاری بهکار گرفته شده و لینکها را به فهرست دیگری اضافه میکند . به مثال زیر نگاه کنید .

```
def findlinks(onepage, add_to=[]):
    new_links = onepage.search_for_links()
    add_to.extend(new_links)
    return add_to
```

در ظاهر همه چیز خیلی خوب و عادی به نظر میرسد . این قطعه کد کار میکند . اما مشگلی در این زمینه وجود دارد . اگر ما یک فهرست را به پارامتر add_to اضافه کنیم، قطعه کد مطابق انتظار ما کار میکند . با این وجود اگر به تابع اجازه دهیم از مقادیر پیشفرض استفاده کند، اتفاق جالب دیگری رخ میدهد . قطعه کد را با مقدار زیر آزمایش کنید .

```
def fn(var1, var2=[]):
    var2.append(var1)
    print var2
fn(3)
fn(4)
fn(5)
```

شما انتظار دارید تا مقادیر [3] ، [4] و [5] را مشاهده کنید . اما در مقابل [3] ، [3,4] و [3,4,5] را دریافت خواهید کرد . چرا این اتفاق رخ داد؟ شما فهرست یکسانی را در هر زمان اجرا مشاهده میکنید . در پایتون، زمانی که ما یک تابع شبیه به این مورد مینویسیم، فهرست به عنوان بخشی از تعریف تابع به کار میرود . در نتیجه هر زمان تابع فراخوانی میشود، فهرست جدیدی ساخته نشده و از همان فهرست قبل استفاده میشود . البته این مشگل زمانی به وجود میآید که از مقادیر پیشفرض یا مقادیر تغییرپذیر در تابع استفاده کنیم یا تابع را با مقدار پیشفرض فراخوانی کنیم . برای حل این مشگل باید در زمان تعریف تابع از نوع تغییرناپذیر (Immutable) استفاده کنیم . در نتیجه از قطعه کد زیر باید استفاده کنیم .

```
def search_for_links(page, add_to=None):
    if not add_to:
        add_to = []
    new_links = page.search_for_links()
    add_to.extend(new_links)
    return add_to
```

BY
SA

بهکارگیری نوعهای دادهای تغییرپذیر در قالب متغیرهای کلاس

این مشکل یکی از رایجترین و مرسومترین خطاهایی است که اغلب برنامهنویسان پایتون با آن روبرو هستند. به قطعه کد زیر توجه کنید .

```
class URLCatcher(object):
   urls = []
   def add_url(self, url):
      self.urls.append(url)
```

این کد به نظر عادی است. ما یک شی با آدرسهای اینترنتی ذخیرهساز در اختیار داریم . زمانی که متد add-url را فراخوانی میکنیم، یک آدرس اینترنتی را برای ذخیرهسازی اضافه میکند . به نظر میرسد قطعه کد کامل است! اجازه دهید تابع را در عمل مشاهده کنیم .

```
a = URLCatcher()
a.add_url('http://www.google.com')
b = URLCatcher()
b.add_url('http://www.mysite.co')

b.urls
['http://www.google.com', 'http://www.mysite.co']

a.urls
['http://www.google.com', 'http://www.mysite.co']
```

صبر کنید چه اتفاقی افتاد؟ ما انتظار این مورد نداشتیم . ما دو شی جداگانه a و b را معرفی کنیم . یکی از اشیا یک آدرس اینترنتی ارائه میکند و شی b آدرس دیگری را ارائه میکند . اما چگونه است که این اشیا هر دو آدرس اینترنتی دارند؟ به نظر میرسد همان مشکل مثال قبل را تکرار کردهایم . فهرست آدرسهای اینترنتی در زمان تعریف کلاس ساخته شده و معرفی میشود . همه نمونههای تعریف شده از کلاس از یک فهرست یکسان استفاده میکنند. برای حل این مشکل قطعه کد زیر را میتوانیم همانند مثل زیر ویرایش کنیم .

```
class URLCatcher(object):
   def __init__(self):
      self.urls = []
```

```
def add_url(self, url):
    self.urls.append(url)
```

اکنون آدرسهای اینترنتی زمانی که شی ساخته میشود به درستی مقداردهی اولیه میشود . زمانی که دو شی جداگانه را تعریف میکنیم، آنها از دو فهرست جداگانه استفاده میکنند .

نویسنده :
حمیدرضا_تائبی
دسته بندی:
کارگاه
تاریخ انتشار :
18:30 - 17/09/1396
برچسب:
پایتون - برنامهنویسی - اشتباهات رایج پایتون

نشانی
□□□□:https://www.shabakeh-mag.com/workshop/10932/%D8%A7%DA%AF%D8%B1-%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-%D9%87%D8%B3%D8%AA%DB%8C%D8%AF-%D8%A7%D8%B2-%D8%A7%DB%8C%D9%86-%D8%AF%D9%88-%D8%A7%D8%B4%D8%AA%D8%A8%D8%A7%D9%87-%D8%AF%D9%88%D8%B1%DB%8C-%DA%A9%D9%86%DB%8C%D8%AF