

فضای نام و محدوده‌ها در پایتون

فضای نام (Namespace)

فضای نام یک سیستم نامگذاری است که برای تمایز میان متغیرها، توابع و ویژگی‌ها در برنامه‌نویسی پایتون استفاده می‌شود. هر کلاس یا متغیر محلی (local) در داخل یک تابع یک فضای نام مجزا ایجاد می‌کند. به عنوان مثال، فرض کنید در یک کلاس چند نفر با نام "علی" داریم. برای اینکه بتوانیم این افراد را از هم تمایز دهیم، باید به نام خانوادگی‌شان اشاره کنیم. مثلاً اگر فردی به نام "علی احمدی" باشد، باید نام خانوادگی‌اش را نیز در نظر بگیریم تا از سایر افرادی که نام "علی" دارند، متمایز شویم. این مفهوم در فضای نام نیز مشابه است. به این معنی که دو شیء ممکن است نام یکسانی داشته باشند، اما اگر در دو فضای نام مختلف قرار داشته باشند، هیچ ارتباطی با هم ندارند.

به مثال زیر توجه کنید:

```
1 def greet_user():
2     name = "Ali"
3     print(name)
4
5 def greet_admin():
6     name = "Ali"
7     print(name)
```

[Copy](#)[Python](#)

در این مثال، هر تابع فضای نام جداگانه‌ای برای متغیر name دارد، بنابراین هیچ تداخلی در استفاده از این متغیرها نخواهیم داشت.

انواع فضای نام

فضاهای نام در زمان‌های مختلف ایجاد می‌شوند و طول عمر متفاوتی دارند. به طور کلی، فضای نام‌ها به سه دسته تقسیم می‌شوند:

۱. **فضای نام داخلی (Built-in)** فضای نام داخلی زمانی ساخته می‌شود که مفسر پایتون شروع به کار می‌کند و تا زمانی که برنامه به پایان برسد، باقی می‌ماند. توابع و خطاهای پیش‌فرض پایتون در این فضا قرار

دارند. به طور مثال، توابعی مثل `print()` یا `id()` جزو فضای نام داخلی هستند.

۲. **فضای نام عمومی (Global)** فضای نام عمومی برای متغیرهایی است که در سطح ماژول یا اسکریپت تعریف شده‌اند. این متغیرها از هر نقطه در برنامه قابل دسترسی هستند. به مثال زیر دقت کنید:

```
1 | x = 50
2 |
3 | def show_value():
4 |     print(x)
5 |
6 | show_value()
7 | print(x)
```

در اینجا، متغیر `x` در فضای نام عمومی تعریف شده است و در هر دو مکان قابل دسترسی است.

۱. **فضای نام محلی (Local)** فضای نام محلی مربوط به متغیرهایی است که در داخل توابع تعریف شده‌اند و تنها در همان تابع قابل دسترسی هستند. به مثال زیر توجه کنید:

```
1 | def calculate_sum():
2 |     x = 10
3 |     print(x)
4 |
5 | calculate_sum()
6 | print(x)
```

در اینجا، متغیر `x` فقط در داخل تابع `calculate_sum` معتبر است و دسترسی به آن از خارج تابع باعث خطا می‌شود.

محدوده‌ها (Scope)

محدوده‌ها در پایتون برای تعیین دسترسی به متغیرها استفاده می‌شوند و انواع مختلفی دارند:

۱. **محدوده محلی (Local Scope)** این محدوده مربوط به متغیرهایی است که داخل توابع یا بلوک‌های کد تعریف می‌شوند. به طور مثال:

```

1  def my_function():
2      y = 10
3      print(y)
4
5  my_function()

```

در اینجا، متغیر `y` فقط در داخل `my_function` قابل دسترسی است.

۱. **محدوده عمومی (Global Scope)** این محدوده مربوط به متغیرهایی است که در سطح ماژول یا اسکریپت تعریف می‌شوند و از هر نقطه در برنامه قابل دسترسی هستند.

```

1  x = 100
2
3  def display():
4      print(x)
5
6  display()

```

در اینجا، متغیر `x` در سطح جهانی (`global`) تعریف شده است و از هر نقطه قابل دسترسی است.

۱. **محدوده غیرمحلی (Nonlocal)** گاهی اوقات ممکن است بخواهید به متغیری دسترسی پیدا کنید که در یک تابع بیرونی تعریف شده است. در این موارد از کلمه کلیدی `nonlocal` استفاده می‌کنیم:

```

1  def outer():
2      age = 25
3      def inner():
4          nonlocal age
5          age = 30
6          print(age)
7      inner()
8      print(age)
9
10 outer()

```

در اینجا، متغیر `age` در تابع بیرونی `outer` قرار دارد و با استفاده از `nonlocal` در تابع `inner` تغییر می‌کند.

مقایسه انواع متغیرها: محلی، عمومی، و غیرمحلی

در این مثال، می‌توانید نحوه استفاده از متغیرهای محلی، عمومی و غیرمحلی را مقایسه کنید:

```
1 message = "Hello"
2
3 def example_function():
4     message = "How are you?"
5
6     def local_function():
7         message = "Good"
8         print("Inside local: " + message)
9
10    def nonlocal_function():
11        nonlocal message
12        message = "Great"
13        print("Inside nonlocal: " + message)
14
15    def global_function():
16        global message
17        message = "Amazing"
18        print("Inside global: " + message)
19
20    local_function()
21    print(message)
22    nonlocal_function()
23    print(message)
24    global_function()
25    print(message)
26
27 example_function()
28 print(message)
```

در اینجا، ابتدا از یک متغیر محلی در `local_function` استفاده می‌کنیم، سپس با استفاده از `nonlocal` مقدار متغیر در تابع بیرونی تغییر می‌دهیم، و در نهایت با استفاده از `global` مقدار متغیر سراسری تغییر می‌کند.