

クラス・スコープの立ち位置と特別な扱い方法

スコープ構造

- LEGBルールに照らし合わせてみる
- 関数スコープが、E・L(外側の関数がenclosing, 内側の関数がlocalに当たる)
- クラススコープは、関数と並列関係

Built-ins

```
└─ Global
   └─ 関数スコープ (ネスト可能)
      └─ クラススコープ (特殊ルール)
```

クラス内のスコープについて

- クラス変数へのアクセス
 - 定義時のエリアでは、`クラス変数名` だけでアクセス出来る
 - `クラス名・クラス変数名`
 - インスタンスメソッドでも、クラスメソッドでもアクセス出来る
 - インスタンスメソッド内では、`self.クラス変数名` でアクセス出来る
 - クラスメソッド内では、`cls.クラス変数名` でアクセス出来る

```
class MyClass:
    class_var = "クラス変数"
    class_var_2 = "クラス変数_2"

print(class_var) # OK: "クラス変数" (定義時は見える)
# print(MyClass.class_var_2) # 見えているので、アクセス方法がおかしい
print(class_var_2) # 当然こう書くべきである

def method(self):
    # print(cls.class_var) # clsは引数にない
    # print(class_var) # NG: 見えない!
    print(self.class_var) # OK: これなら見える
    print(MyClass.class_var_2) # OK: クラス名.クラス変数名

@classmethod
def class_method(cls):
    # print(self.class_var) # selfは引数に無い
    # print(class_var) # 見えない
    print(cls.class_var) # OK: これなら見える
    print(MyClass.class_var_2) # OK: クラス名.クラス変数名
```

初期化の際に実行されたprint文による

```
obj = MyClass()
"""
クラス変数
クラス変数_2
"""

# インスタンスメソッドにアクセス
obj.method()
"""
クラス変数
クラス変数_2
"""

# クラスメソッドにアクセス
MyClass.class_method()
"""
クラス変数
クラス変数_2
"""
```

応用問題

LEGBルール（復習）

- Local（ローカル）：関数内
- Enclosing（エンクロージング）：外側の関数
- Global（グローバル）：モジュールレベル
- Built-in（ビルトイン）：組み込み

クラススコープの特殊性

```
var1 = "Global"

def outer():
    var2 = "Enclosing"
    print(var1)
    print(var2)

# classスコープを、LEGB的なルールに当てはめて、
# 関数的に考えることはできない！
class MyClass:
    # クラス変数
    var3 = "Class" # ← これはLEGB的なルールに含まれない！
    print()
    print(var1)
```

```

print(var2)
print(f"クラス定義時: {var3}") # "Class"

def method(self):
    var4 = "Local"
    # Enclosing, Global, Built-inの順で探す
    # でもClass変数はスキップされる!
    print()
    print(var1)
    print(var2)
    # print(var3) # エラー (class変数)

    print(" ---")
    print("== メソッド内からクラス変数にアクセスする方法 ==")
    print("クラス名.クラス変数")
    print(f"MyClass.var3: {MyClass.var3}")
    print(f"メソッド内: {var4}") # "Local" (L)

obj = MyClass()
obj.method()

```

```
outer()
```

```
"""
```

```
Global
Enclosing
```

```
Global
Enclosing
クラス定義時: Class
```

```
Global
Enclosing
```

```

---
== メソッド内からクラス変数にアクセスする方法 ==
クラス名.クラス変数
MyClass.var3: Class
メソッド内: Local
"""

```

重要なポイント:

- クラス定義時は、クラス変数が見える
- メソッド内からは、**クラス変数** はLEGBの検索対象外
- メソッド内では L → E → G → B の順で検索
- メソッド内で、クラス変数にアクセスするには `self.var3` や `cls.var3` が必要
 - ここでは、`self.var3` が使える
 - クラス名.クラス変数 (MyClass.var3) でもOK

つまり、クラススコープは「特別なスコープ」で、通常のルールから除外されているんです。

END