01. Scope of objects and Names

Python में प्रत्येक object का एक name हो सकता है। Scope वह क्षेत्र है जहाँ किसी name को program में पहचाना और उपयोग किया जा सकता है।

Local और Global scope

Function के अंदर बनाया गया name सामान्यतः local scope में होता है। Function के बाहर बनाया गया name module के global scope में होता है।

```
x = 10              # global name

def show():
    y = 20            # local name
    print(x, y)
```

एक ही नाम अलग-अलग scope में अलग objects को refer कर सकता है। Scope name की visibility और lifetime को समझने में मदद करता है।

02. LEGB Rule

Python किसी name को खोजते समय LEGB Rule का पालन करता है। Search सबसे पहले Local, फिर Enclosing, फिर Global और अंत में Built-in scope में होती है।

Letter	Scope का अर्थ
L	Local: current function के अंदर
E	Enclosing: बाहरी nested function
G	Global: current module का स्तर
B	Built-in: Python के built-in names, जैसे len() और print()

```
x = "Global"

def outer():
    x = "Enclosing"
    def inner():
        x = "Local"
        print(x)
    inner()
outer()
```

ऊपर print(x) का output Local होगा, क्योंकि Python सबसे पहले Local scope में name खोजता है।

03. Module Basics

Module एक Python file है जिसमें functions, classes, variables और executable statements रखे जा सकते हैं। Module का file extension .py होता है।

Module बनाने का उदाहरण

```
# calculation.py
def add(a, b):
    return a + b

PI = 3.14
```

Module का उपयोग

```
import calculation
result = calculation.add(2, 3)
print(result)
```

Modules code को reuse करने, program को छोटे भागों में बाँटने और maintenance आसान बनाने में मदद करते हैं।

04. Module Files as Namespaces

जब कोई module import होता है, तो Python उसके लिए एक namespace बनाता है। Namespace names और उनके objects के बीच mapping रखता है।

```
# tools.py
name = "Python"
def greet():
    return "Hello " + name

# main.py
import tools
print(tools.name)
print(tools.greet())
```

tools.name और tools.greet() में dot (.) यह बताता है कि name और function tools module के namespace से लिए जा रहे हैं।

अलग-अलग modules में समान name हो सकते हैं, क्योंकि हर module का अपना namespace होता है।

05. Import Model

import statement module को load करता है और उसका name current namespace में उपलब्ध कराता है। Python सामान्यत: एक module को एक ही बार load करके sys.modules में रखता है।

Import के सामान्य तरीके

```
import math
print(math.sqrt(25))

from math import sqrt
print(sqrt(25))

import math as m
print(m.pi)
```

import module करने पर module name के साथ access करें। from ... import ... में चुना हुआ name सीधे current namespace में आ जाता है। as alias से छोटा नाम बनाया जा सकता है।

Name conflict से बचने के लिए सामान्यत: import module वाला तरीका स्पष्ट और सुरक्षित माना जाता है।

06. Reloading Modules

यदि module file में बदलाव किया गया हो और उसे फिर से पढ़ना हो, तो importlib.reload() का उपयोग किया जा सकता है।

```
import mymodule
import importlib

importlib.reload(mymodule)
```

reload() उसी module object को update करने का प्रयास करता है। यह module को दोबारा import करने से अलग है, क्योंकि सामान्य import पहले से loaded module को फिर से execute नहीं करता।

Reloading मुख्यतः interactive development और testing में उपयोगी है। Program के बड़े design में dependencies को स्पष्ट रखना जरूरी है।

07. Quick Revision और Practice

एक नज़र में पूरा Unit

Name search: Local → Enclosing → Global → Built-in Module: .py file → namespace → import → आवश्यकता हो तो reload()

Topic	याद रखने योग्य बात
Scope	name कहाँ दिखाई देगा, यह बताता है
LEGB	name search का क्रम
Module	reusable Python file
Namespace	names और objects की mapping
Import	module को उपलब्ध कराता है
reload()	loaded module को फिर से पढ़ने में मदद करता है

Practice questions

LEGB में E का क्या अर्थ है? Module namespace क्या होता है? import math और from math import sqrt में क्या अंतर है? Module में बदलाव के बाद कौन-सा function उपयोग करेंगे?