**Soru 1) Yorumlayıcı nedir?**

**Bilgisayar biliminde bir yorumlayıcı (interpreter), yüksek seviyeli bir programlama dilinde yazılmış talimatları, önceden makine koduna çevirmeye gerek kalmadan doğrudan çalıştıran bir programdır.**

Bir başka deyişle, derleyicilerden farklı olarak **yorumlayıcılar**, programı adım adım okur ve anında yürütür. Bu sayede hızlı testler yapmaya ve dinamik program geliştirmeye olanak tanır.

**Soru 2) Python yorumlayıcısının ismi nedir?**

Python’un en yaygın kullanılan yorumlayıcısı **CPython**’dur.

🔹 **CPython**, C diliyle yazılmış **Python’un resmi referans yorumlayıcısıdır**.  
🔹 Bunun dışında Python’un farklı platformlar için geliştirilmiş yorumlayıcıları da bulunmaktadır:

* **PyPy** → Daha hızlı çalışması için optimize edilmiş Python yorumlayıcısı
* **Jython** → Java platformunda çalışan Python yorumlayıcısı
* **IronPython** → C# ile geliştirilmiş ve .NET platformunda çalışan yorumlayıcı

Ancak, günlük kullanımda CPython en yaygın olanıdır ve "Python yorumlayıcısı" dendiğinde genellikle CPython kastedilir.

**Soru 3) Python yorumlayıcısı bir tanımlayıcı gördüğünde hangi kapsamda arar?**

Python, bir **değişkenin ya da fonksiyonun tanımını** bulmak için belirli bir **arama sırası** takip eder.

📌 **Öncelikle en içteki (yerel) kapsamdan başlar** ve **dışa doğru genişleyerek arama yapar**.  
📌 **Eğer değişken bulunduysa**, Python onu kullanır.  
📌 **Bulunamazsa**, daha geniş kapsamları kontrol eder.

Bu kapsamların nasıl çalıştığını daha iyi anlamak için **LEGB kuralına** bakalım.

**Soru 4) Bir değişkenin nerede olduğunu görmek için Python hangi kapsamları hangi sırayla denetler?**

**Python yorumlayıcısı, bir değişkeni veya fonksiyonu ararken LEGB sırasını takip eder:**

**L (Local - Yerel Kapsam):**  
 Öncelikle, değişkenin **o anda yürütülen fonksiyon içinde tanımlanmış olup olmadığına** bakılır.

**E (Enclosing - Kapsayıcı Kapsam):**  
 Eğer değişken **iç içe fonksiyonlar** varsa, Python **bir üstteki kapsama** bakar.  
 Örneğin, iç fonksiyonlar **dış fonksiyonların değişkenlerine erişebilir.**

**G (Global - Küresel Kapsam):**  
 Eğer değişken **ne yerel ne de kapsayıcı kapsamda** bulunamazsa, Python **tüm dosya boyunca geçerli olan küresel değişkenlere bakar.**

**B (Built-in - Yerleşik Kapsam):**  
 Eğer değişken hala bulunamazsa, Python **yerleşik (built-in) fonksiyonlar ve sabitler** içeren en geniş kapsamı kontrol eder.  
 Örneğin, print() veya len() gibi fonksiyonlar **yerleşik kapsamda** bulunur.

**Özetle:** Python, **LEGB sırasına göre** kapsamları araştırır ve ilk bulduğu tanımı kullanır.

**Soru 5) Örnek bir Python programında kapsamları belirleyelim!**

Aşağıdaki kod örneğinde **LEGB kapsamlarını** inceleyelim:

def hesapla(x, y): # 📌 Kapsayıcı (Enclosing) kapsam

toplam = x + y # 🔹 Yerel (Local) kapsam

fark = x - y # 🔹 Yerel (Local) kapsam

return toplam, fark

sonuc1, sonuc2 = hesapla(5, 3) # Küresel (Global) kapsam

print(f"Toplam: {sonuc1}, Fark: {sonuc2}") # Küresel (Global) kapsam

📌 **Kapsamların Açıklaması:**  
🔹 toplam ve fark değişkenleri **yerel (local) kapsamda** tanımlanmıştır.  
📌 hesapla(x, y) fonksiyonunun kendisi, **kapsayıcı (enclosing) kapsamda** yer almaktadır.  
 sonuc1, sonuc2 ve print() çağrıları ise **küresel (global) kapsamda** çalışmaktadır.

**Python bu kodu çalıştırırken:**  
Önce toplam ve fark değişkenlerini **yerel kapsamda** arar.  
Eğer bulamazsa, bir üst kapsama (kapsayıcı) çıkar.  
Eğer orada da yoksa **küresel kapsamda** arar.  
Hâlâ bulunamazsa, **yerleşik (built-in) fonksiyonlara** bakar.