

Python'da gömülü fonksiyonlar, Python diline entegre edilmiş ve herhangi bir modül eklemeye gerek kalmadan kullanılabilen işlevlerdir , yani varsayılan olarak bulunan fonksiyonlardır. Bu fonksiyonlar, programlama süreçlerini hızlandırır ve yaygın işlemleri kolaylaştırır. İşte Python'da en çok kullanılan 10 gömülü fonksiyon ve açıklamaları:

1. `print()`

- **Tanım:** Bir ifadeyi veya veriyi konsola yazdırır.
- **Kullanım:**

```
print("Merhaba, Dünya!")  
# Çıktı: Merhaba, Dünya!
```

- **Özellikler:** Birden fazla argüman alabilir ve çıktıdaki öğeleri boşlukla ayırır. `sep` , `end` gibi parametrelerle yazdırma biçimi özelleştirilebilir.

2. `len()`

- **Tanım:** Bir dizinin (liste, string, tuple vb.) uzunluğunu döner.
- **Kullanım:**

```
s = "Python"  
print(len(s)) # 6
```

- **Özellikler:** Listenin, string'in veya diğer iterable türlerin içindeki eleman sayısını verir.

3. `type()`

- **Tanım:** Bir değişkenin veya objenin tipini döner.
- **Kullanım:**

```
x = 10
print(type(x))  # <class 'int'>
```

- **Özellikler:** Verinin türünü öğrenmek için kullanılır, özellikle dinamik tip kontrolü yaparken faydalıdır.

4. input()

- **Tanım:** Kullanıcıdan girdi alır. Girdi her zaman string tipinde döner.
- **Kullanım:**

```
isim = input("İsminizi girin: ")
print(f"Merhaba, {isim}")
```

- **Özellikler:** Terminal veya konsoldan kullanıcı etkileşimi için kullanılır. Girdi varsayılan olarak string döner, gerekiyorsa `int()` veya `float()` gibi fonksiyonlarla dönüştürülmelidir.

5. int()

- **Tanım:** Bir değeri tamsayıya dönüştürür.
- **Kullanım:**

```
x = int("10")
print(x)  # 10
```

- **Özellikler:** String, float gibi diğer veri türlerinden tamsayıya dönüşüm sağlar.

6. float()

- **Tanım:** Bir değeri ondalıklı sayıya dönüştürür.
- **Kullanım:**

```
x = float("3.14")
print(x)  # 3.14
```

- **Özellikler:** Tamsayı, string vb. veri türlerini ondalıklı sayıya çevirir.

7. `sum()`

- **Tanım:** Bir iterable'ın (örneğin liste) elemanlarının toplamını döner.
- **Kullanım:**

```
sayilar = [1, 2, 3, 4]
print(sum(sayilar))  # 10
```

- **Özellikler:** Sayı dizilerinin toplamını almak için kullanılır. İkinci bir argüman vererek başlangıç değeri belirlenebilir.

8. `max()`

- **Tanım:** Bir iterable'ın veya bir dizi argümanın en büyük değerini döner.
- **Kullanım:**

```
sayilar = [1, 2, 3, 4]
print(max(sayilar))  # 4
```

- **Özellikler:** Verilen iterable'daki en büyük değeri bulur. Birden fazla argüman verilirse, bu argümanlar arasındaki en büyüğü döner.

9. `min()`

- **Tanım:** Bir iterable'ın veya bir dizi argümanın en küçük değerini döner.
- **Kullanım:**

```
sayilar = [1, 2, 3, 4]
```

```
print(min(sayilar)) # 1
```

- **Özellikler:** Verilen iterable'daki en küçük değeri bulur. Birden fazla argüman verilirse, bu argümanlar arasındaki en küçüğü döner.

10. range()

- **Tanım:** Belirli bir aralıkta ardışık sayılar üretir.
- **Kullanım:**

```
for i in range(5):  
    print(i) # 0, 1, 2, 3, 4
```

- **Özellikler:** Belirli bir başlangıç ve bitiş noktası arasında sayı dizisi oluşturur. Genellikle `for` döngüleri ile birlikte kullanılır. Üç parametre alabilir: başlangıç, bitiş ve adım sayısı.

Özet:

Bu gömülü fonksiyonlar, Python programlama dilinin en temel ve yaygın kullanılan araçlarıdır. Verilerle etkileşim, tip dönüşümü, matematiksel işlemler ve döngü yapıları gibi birçok işlevi kolaylaştırarak programcılara büyük esneklik sağlarlar.