**Fonksiyonlar :**

Fonksiyon(Alt program, yordam, modül, posedür) tanım olarak, bir ana program ya da bir alt program tarafından çağrılan ve kendi içinde bir bütün oluşturan (yani belli bir iş yapan) program parçaçığıdır. Programların, anlışılabilirliği, hata takibini ve program üzerinde ileri-geri kod takibi kolaylaştırmak için programın fonksiyonlara ayrılması gerekir. Böylece program bir nevi modülerlik kazanmış olur.

**Fonksiyonar kendi içinde:**

* Parametreli (dışarıdan değer alan) veya parametresiz (dışarıdan değer almayan),
* Geriye (çağırıldığı yere) değer döndüren veya değer döndürmeyen fonksiyonlar şeklinde sınıflandırabiliriz.

**Fonksiyonları genel özellikleri**

* Başka programlar tarafından çağırılırlar.
* Fonksiyonlar bir defa oluşturulur, bir çok kez çağırılabilirler.
* Genellikle çağırıldıkalrı programlardan aldıkalrı girdileri/verileri işleyerek sonuçları yine çağıran programa göndeririler.
* Fonksiyon girdilerine parametre veya argüman denir.
* Parametre geçişinde (fonksiyon dışından değer alımına) izin verirler.
* Eğer fonksiyon çağırıldığı yere bir değer döndürücekse, döndürülücek değer return deyimi ile belirtilir.
* Python kod satırlarında hiyerarşik sıraya dikkat edilmeli, önce fonksiyon tanımlanmalı sonra çağırılmalı.

**Fonksiyonları niçin kullanmalıyız ?**

Bir program içerisinde baynı işi gören bir grup kod satırını programın çeşitli yerlerinde tekrar ve tekrar yazıp kullanmak yerine bu kod satırlarını bir defa yazıp sürekli çağırabilriz. Bunun için bu kod grubunu bir fonksiyon başlığı/ismi altında tanımlamamız yeterlidir. Fonksiyonlar ile hem kodlama tekrarını önlemiş, hem de programın anlaşılabilirliğini arttırmış oluruz.

**Python dilinde fonksiyon tanımlama:**

Her program dili belli amaçlara yönelik yerleşik/hazır fonksiyonlar (örneğin mutlak değeri almka için abs(), sinüs almak için sin(),üs almak için pow() gibi) içerir. Bu ve benzeri fonksiyonlar, programlama dilinin ilgili modülünde (kütüphanesinde) yer almakta ve işlemleri gerçekleştrmede kolaylık sağlamaktadırlar. Örneğin sin() fonksiyonunu kullanmak için 'math' kütüphanesinin program başında 'import math' şeklinde eklenmesi gerekir. Pythonda hazır olan fonksiyonların yanında kenimizde fonksiyon tanımlayabilriz. Fonksiyon tanımlamak için **'def'** anahtar kelimesi kullanılır. Tanımlanan fonksiyon başka bir fonksiyon içerisinden sadece ismi belirtilerek çağrılır.

**Açıklama**: Python dilinde her modül, '\_name\_' isimli bir özelliğe sahiptir. Modül ana program olarak çağırıldığında '\_name\_' özelliğinin değeri '\_main\_' olarak ayarlanır. (if\_name\_=='\_main\_': [tab] benimfonksiyon()) benimfonksiyonu ana program olarak çalıştır demek.

def fon\_adi():

yapılıcaklar

return Sonuç

**pass Komutu:** 'Hiçbir şey yapma' komutudur. Pythonda kod gövdesi (işlevi) olmayan sadece başlığı olan fonksiyon tanımı geçersizdir. Eğer henüz içeriğine karar vermdeğiniz fonksiyonunuz var ve hata mesajı da almak istemiyorsanız fonksiyon içerisine pass komutu yazarak sonlandirabilirsiniz.

**return Komutu:** Fonksiyonun geriye döndüreceği değeri üreten komuttur. Python dilinde return ile birlikte aynı anda birden fazla değer geri döndürülebilir.

**lambda Fonksiyonu**: Tek satırlık kısa fonksiyonlar tanımlamak için kullanılır. Kullanım şekli fonk\_ismi=lambda değişkenler: komut satırı. örn(topla=lambda a,b:a+b) print(topla(3,8)) #11

**map Fonksiyonu:** Bir grup veriye(liste,dizi..) fonksiyonun değerini object(nesne) türünde döndürür. Kullanım şekli a=map(fonk\_ismi,liste). map fonksiyonun döndürdüğü değeri doğrudan print fonksiyonu ile yazdıramayız ama list() fonksiyonu ile listeye dönüştürüp yazdırabilriz.

**filter Fonksiyonu**: Bir grup veriye (liste,dizi..) fonksiyonun True olduğu durumlardaki değerlerini object(nesne) türünde döndürür. Yani mevcut liste üzerinde filtreleme yapar. Kulanım şekli a=filter(fonk\_ismi,liste) bu tür de list() fonksiyonu ile yazdırılır.

*def ciftS(sayi):*

*return sayi%2==0*

*liste=range(1,20)*

*f\_liste=filter(ciftS,liste)*

*print(list(f\_liste))*

**Matematiksel Fonksiyonlar** : Python dilinde bazı matematiksel fonksiyonlar,yerleşik fonksiyon(built in functions) olarak tanımlanıştır. Bu fonksiyonları (abs() gibi) doğrudan ismi ile çağırabiliriz. Bazıları ise math,cmath modüller içerisinde yer alır ve bu fonksiyonalrı kullanabilmek için math kütüphanesini programın başına eklemek gerekir. (import math)

**Python'da rastgele sayı üretimi (random) :** Python'da rastgele sayı üretimi için random (improt random) kütüphanesine ait random(), randrage(), randit() ve uniform() gibi fonksiyonlar kullanılır.

* random.random() = 0.0<=sayi<1 arasında float tipinde sayı üretir.
* random.randint(s1,s2) = s1<=sayi<=s2 olacak şekilde int tipinde sayı üretir
* random.randrage(s1,s2,[adım]) = s1-s2 değerleri arasında adım miktarına dikkat ederek adım miktarında rastgele float tipinde sayı üretir. Adım miktarı belitilmezse 1 olarak kabul edilir.
* random.choice([liste]) = Liste dizisinden rastgele bir eleman seçer.
* random.sample([liste],a) = Liste dizisinden rastgele a adet örnek seçer.

**Özyinelemeli Fonksiyonlar :** Kendini doğrudan veya dolaylı olarak çağıran fonksiyonlara özyinelemeli(recursive) fonksiyonlar adı verilir. Özyinelemli fonksiyonlar algoritma tasarımını basitleştirir ancak fonksiyon çağırma sayısı arttığı için daha fazla bellek alanı kullanır. Özyineleme, en genel anlamda bir yapının kendi kendini çağırmasıdır. Matruşka özyinelemeye örnek verilebilir.