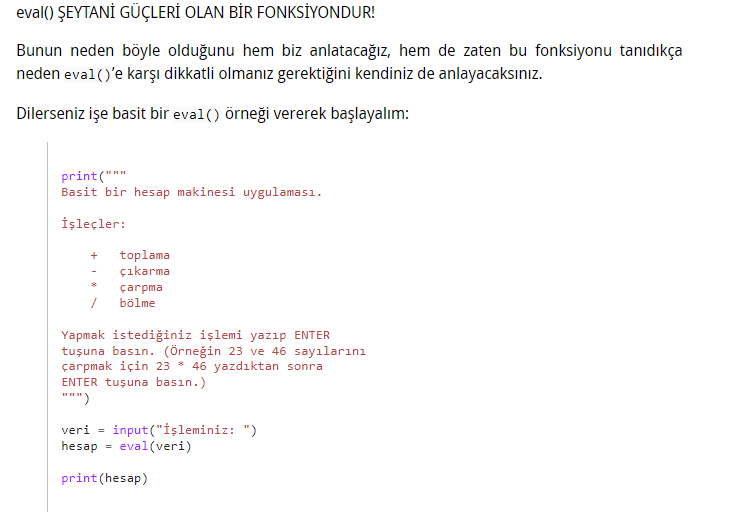
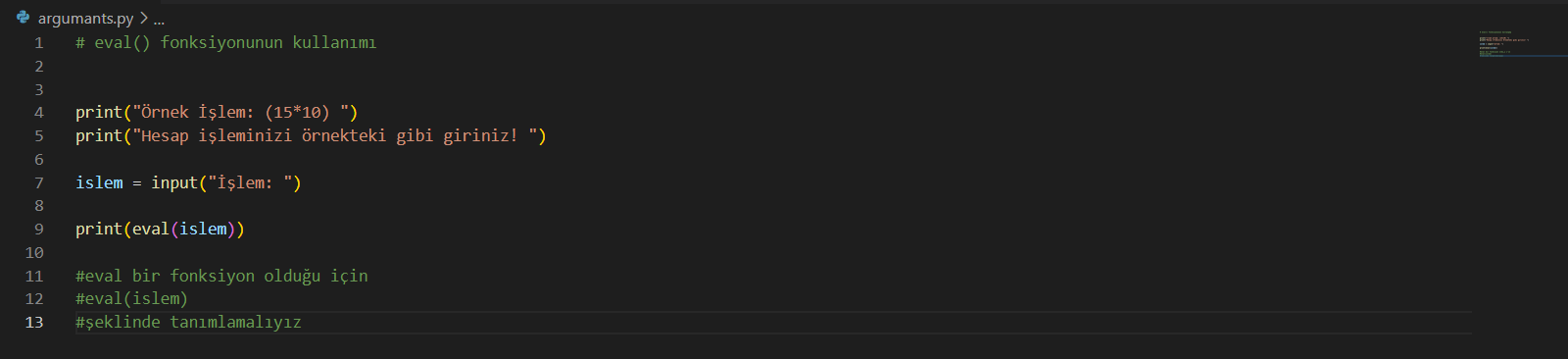
**Eval() ve Exec() Fonksiyonları**





İngilizcede evaluate diye bir kelime bulunur. Bu kelime, ‘değerlendirmeye tabi tutmak, işleme sokmak, işlemek’ gibi anlamlar taşır. İşte eval() fonksiyonundaki eval kelimesi bu evaluate kelimesinin kısaltmasıdır. Yani bu fonksiyonun görevi, kendisine verilen karakter dizilerini değerlendirmeye tabi tutmak ya da işlemektir.

Aslında yukarıdaki örnek programı çalıştırdığımızda bu sorunun yanıtını kendi kendimize verebiliyoruz. Bu programı çalıştırarak, “İşleminiz: “ ifadesinden sonra, örneğin, 45 \* 76 yazıp Enter tuşuna basarsak programımız bize 3420 çıktısı verecektir. Yani programımız hesap makinesi işlevini yerine getirip 45 sayısı ile 76 sayısını çarpacaktır. Dolayısıyla, yukarıdaki programı kullanarak her türlü aritmetik işlemi yapabilirsiniz. Hatta bu program, son derece karmaşık aritmetik işlemlerin yapılmasına dahi müsaade eder.

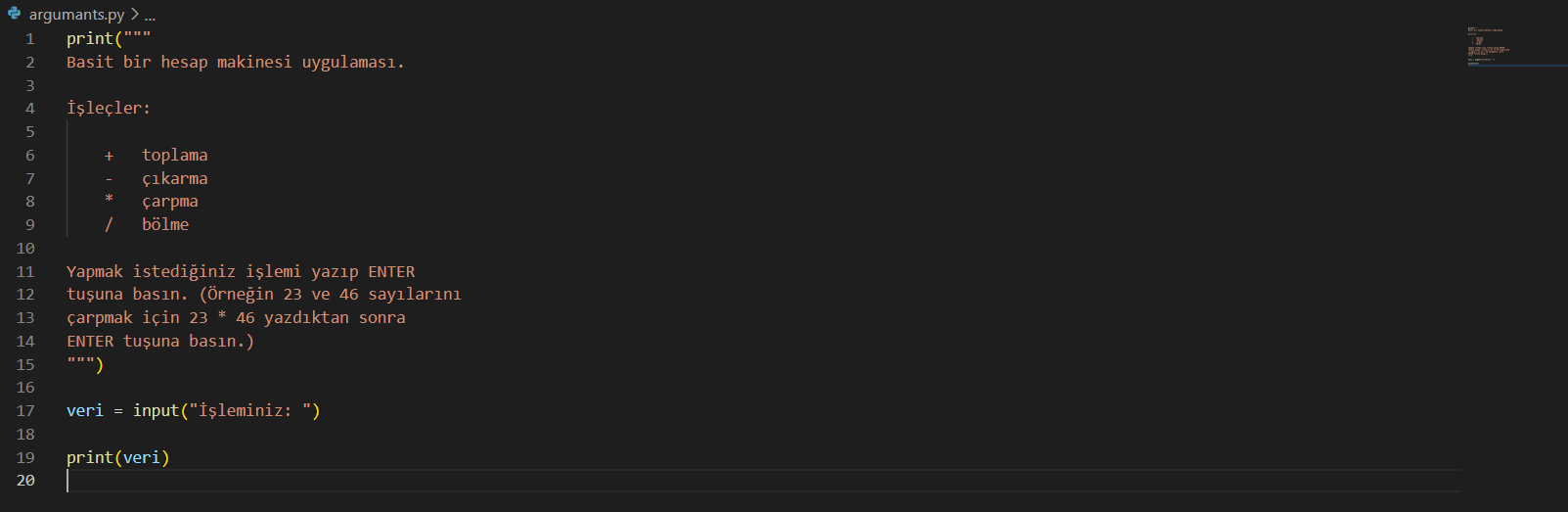
Peki programımız bu işlevi nasıl yerine getiriyor? İsterseniz kodların üzerinden tek tek geçelim.

Öncelikle programımızın en başına kullanım kılavuzuna benzer bir metin yerleştirdik ve bu metni print() fonksiyonu yardımıyla ekrana bastık.

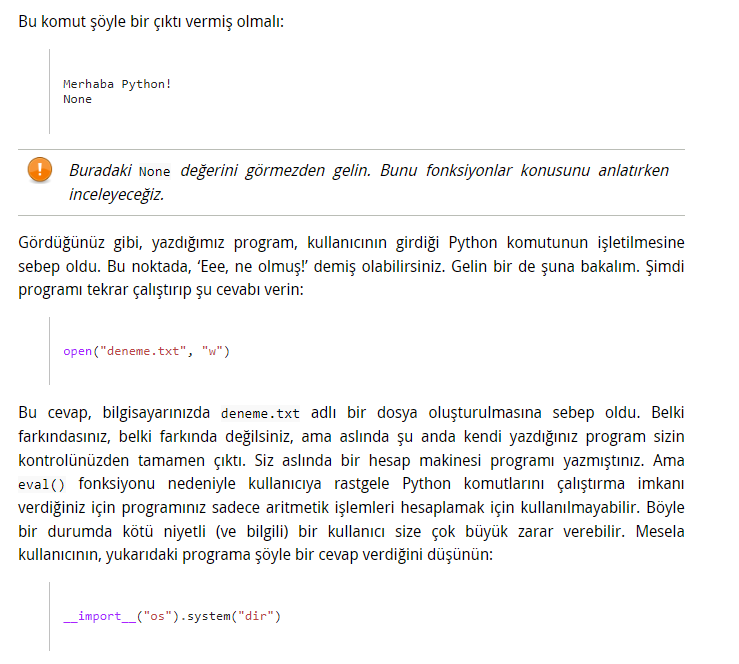
Daha sonra kullanıcıdan alacağımız komutları veri adlı bir değişkene atadık. Tabii ki kullanıcıyla iletişimi her zaman olduğu gibi input() fonksiyonu yardımıyla sağlıyoruz.

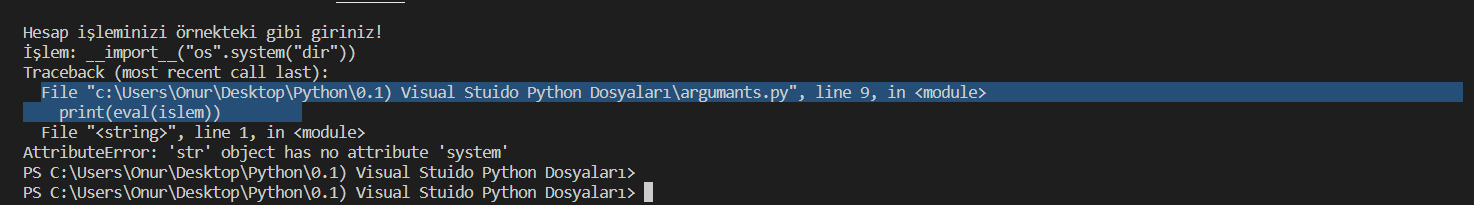
Ardından, kullanıcıdan gelen veriyi eval() fonksiyonu yardımıyla değerlendirmeye tabi tutuyoruz. Yani kullanıcının girdiği komutları işleme sokuyoruz. Örneğin, kullanıcı 46 / 2 gibi bir veri girdiyse, biz eval() fonksiyonu yardımıyla bu 46 / 2 komutunu işletiyoruz. Bu işlemin sonucunu da hesap adlı başka bir değişken içinde depoluyoruz.

Eğer burada eval() fonksiyonunu kullanmazsak, programımız, kullanıcının girdiği 45 \* 76 komutunu hiçbir işleme sokmadan dümdüz ekrana basacaktır. Yani:



Eğer programımızı yukarıdaki gibi, eval() fonksiyonu olmadan yazarsak, kullanıcımız 45 \* 76 gibi bir komut girdiğinde alacağı cevap dümdüz bir 45 \* 76 çıktısı olacaktır. İşte eval() fonksiyonu, kullanıcının girdiği her veriyi bir Python komutu olarak algılar ve bu veriyi işleme sokar. Yani 45 \* 76 gibi bir şey gördüğünde, bu şeyi doğrudan ekrana yazdırmak yerine, işlemin sonucu olan 3420 sayısını verir.



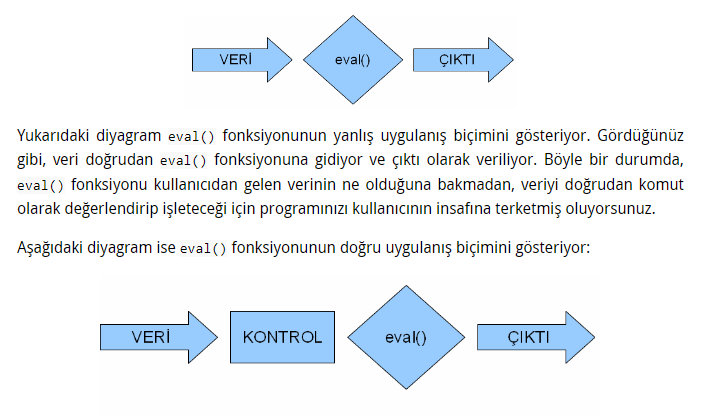


Burada anlamadığınız şeyleri şimdilik bir kenara bırakıp, bu komutun sonuçlarına odaklanın. Gördüğünüz gibi, yukarıdaki programa bu cevabı vererek mevcut dizin altındaki bütün dosyaları listeleyebildik. Yani programımız bir anda amacını aştı. Artık bu aşamadan sonra bu programı şeytani bir amaca yönelik olarak kullanmak tamamen programı kullanan kişiye kalmış… Bu programın, bir web sunucusu üzerinde çalışan bir uygulama olduğunu ve bu programı kullananların yukarıdaki gibi masumane bir şekilde dizin içindeki dosyaları listeleyen bir komut yerine, dizin içindeki dosyaları ve hatta sabit disk üzerindeki her şeyi silen bir komut yazdığını düşünün… Yanlış yazılmış bir program yüzünden bütün verilerinizi kaybetmeniz işten bile değildir. (Bahsettiğim o, ‘bütün sabit diski silen komutu’ kendi sisteminizde vermemeniz gerektiğini söylememe gerek yok, değil mi?)

Eğer SQL Injection kavramını biliyorsanız, yukarıdaki kodların yol açtığı güvenlik açığını gayet iyi anlamış olmalısınız. Zaten internet üzerinde yaygın bir şekilde kullanılan ve web sitelerini hedef alan SQL Injection tarzı saldırılar da aynı mantık üzerinden gerçekleştiriliyor. SQL Injection metoduyla bir web sitesine saldıran cracker’lar, o web sitesini programlayan kişinin (çoğunlukla farkında olmadan) kullanıcıya verdiği rastgele SQL komutu işletme yetkisini kötüye kullanarak gizli ve özel bilgileri ele geçirebiliyorlar. Örneğin SQL Injection metodu kullanılarak, bir web sitesine ait veritabanının içeriği tamamen silinebilir. Aynı şekilde, yukarıdaki eval() fonksiyonu da kullanıcılarınıza rastgele Python komutlarını çalıştırma yetkisi verdiği için kötü niyetli bir kullanıcının programınıza sızmasına yol açabilecek potansiyele sahiptir.

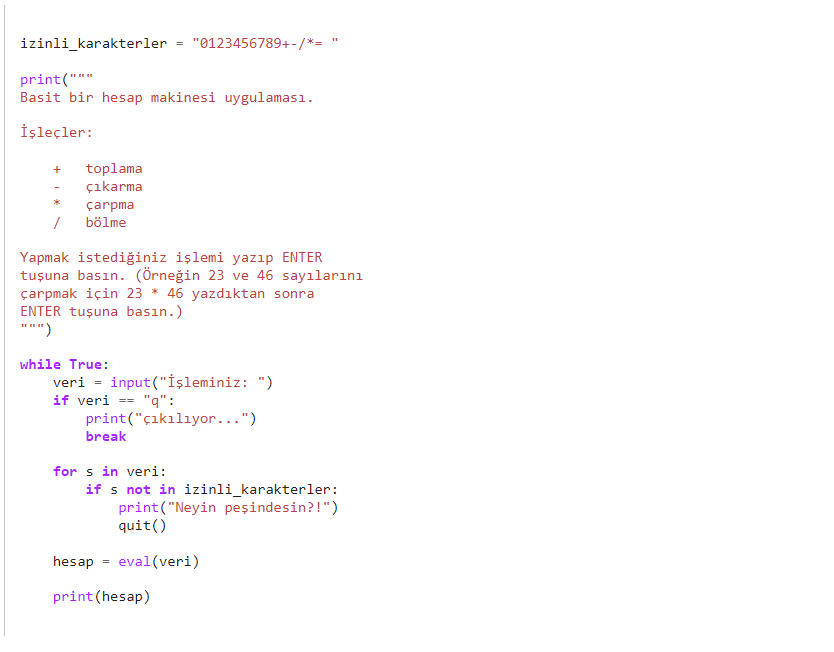
Peki eval() fonksiyonunu asla kullanmayacak mıyız? Elbette kullanacağız. Bu fonksiyonun kullanımını gerektiren durumlarla da karşılaşabilirsiniz. Ama şunu asla aklınızdan çıkarmayın: eval() fonksiyonu her ne kadar son derece yetenekli ve güçlü bir araç da olsa yanlış ellerde yıkıcı sonuçlar doğurabilir. Program yazarken, eğer eval() kullanmanızı gerektiren bir durumla karşı karşıya olduğunuzu düşünüyorsanız, bir kez daha düşünün. eval() ile elde edeceğiniz etkiyi muhtemelen başka ve çok daha iyi yöntemlerle de elde edebilirsiniz. Üstelik performans açısından eval() pek iyi bir tercih değildir, çünkü bu fonksiyon (çoğu durumda farketmeseniz de) aslında yavaş çalışır. O yüzden, eval() fonksiyonunu kullanacağınız zaman, bunun artı ve eksilerini çok iyi tartın: Bu fonksiyonu kullanmak size ne kazandırıyor, ne kaybettiriyor?

Ayrıca eval() fonksiyonu kullanılacağı zaman, kullanıcıdan gelen veri bu fonksiyona parametre olarak verilmeden önce sıkı bir kontrolden geçirilir. Yani kullanıcının girdiği veri eval() aracılığıyla doğrudan değerlendirmeye tabi tutulmaz. Araya bir kontrol mekanizması yerleştirilir. Örneğin, yukarıdaki hesap makinesi programında kullanıcının gireceği verileri sadece sayılar ve işleçlerle sınırlandırabilirsiniz. Yani kullanıcınızın, izin verilen değerler harici bir değer girmesini engelleyebilirsiniz. Bu durumu somutlaştırmak için şöyle bir diyagram çizebiliriz:



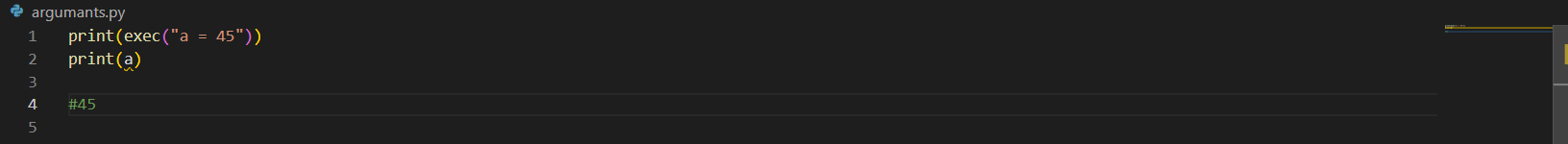
**Örnek**

**Eval fonks. Kontrollerini sağlayarak güvenli girişe izin vermek**



**BİLGİ:** quit() programın tamamen kapatılmasını sağlamaktadır.

**Exec()**



**eval() fonksiyonunu anlatırken güvenlik ile ilgili olarak söylediğimiz her şey exec() fonksiyonu için de geçerlidir. Dolayısıyla bu iki fonksiyonu çok dikkatli bir şekilde kullanmanız ve bu fonksiyonların doğurduğu güvenlik açığının bilincinde olmanız gerekiyor.**

**Kaynaklar:**

**https://python-istihza.yazbel.com/input.html#eval-ve-exec-fonksiyonlari**

**https://nedbatchelder.com/blog/201206/eval\_really\_is\_dangerous.html**