Variable’lar data depolamak için kullanılan parametrelerdir. Örneğin 🡪 character\_name = “John”. Variableler sonraki satılarda aynı variableye farklı değer atayarak update edilebilir, error vermez.

Stringler için “” kullanılır, sayılar veya boolean function için bu gerekli değildir.

Print içerisindeki \n komutu ile alt satıra geçilir. \” ile quot mark oluşturulur.

Str(number) komutuyla string olmayan değerleri stringe dönüştürebiliriz. Python aynı print içerisinde number ve stringi desteklemez, ikisinin kullanılması gerekiyorsa number olan bu komut ile stringe çevrilir.

word.lower() veya word.upper() komutlarıyla print içerisindeki stringin harf boyutunu değiştirebiliriz, word.isupper/lower() komutlarıyla boolean çekebiliriz, bize harflerin büyük/küçük olduğunu true/false olarak gösterir.

Len(word) komutuyla stringin kaç karakterden oluştuğunu görebiliriz. Word[n] komutuyla stringin istediğimiz karakterini numarasını yazarak çekebiliriz, index 0 ile başlar. Word.index(“o”) komutuyla harfin nerede olduğunu sayı olarak gösterir. Indeximize birden fazla harf yazdıysak veya birden fazla harfi kapsıyorsa, sistem başlangıç harifini yazdırır. Word.replace(“word”, “sentence”) komutu ile komutumuzdaki parametreleri istediğimiz başka kelimelerle değiştirebiliriz.

Print komutu içerisinde variable ve “” leri birleştirmek için + kullanılır. “” kullanmadan istediğimiz numerik işlemleri print içerisinde yapabiliriz.

Abs(number) komutuyla sayının mutlak değerini alabiliriz. Pow(x, y) komutuyla x^y’ı elde ederiz. Max/min(x, y) komutlarıyla parantez içindekilerden büyük/küçük olanı belirleyebiliriz. Round(x) komutuyla değer, matematiksel olarak yuvarlanır. Floor/ceil(x) komutları değeri aşağı veya yukarı yuvarlar. Sqrt(x) komutuyla karekök alınabilir.

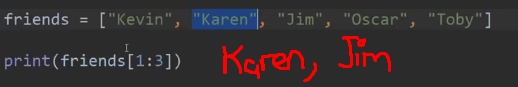
From math import \* komutu pythonda çok daha geniş matematiksel işlemlerin kapısını açar.

Kullanıcıdan input almak için input() komutu kullanılır.



int(x) komutu ile doğal sayılardan oluşan değerler oluşturulur, decimal number konulursa sistem error verir, bunun önüne geçmek için float(y) komutu kullanılır.

Variableyi []’e eşitlemek liste oluşturmak anlamına gelir, örneğin friends = [“Kevin”, “Karen”, “Jim”]. List içerisindeki değerler boolean/number gibi farklı parametrelerle değiştirilebilir. Friends[0] ile aradığımız indexe ulaşabiliriz, eksi değer vermek listi sondan doğru belirler. Friends[x] = “something” komutuyla list içerisindeki değeri değiştirebiliriz.



Something.extend(someone) komutuyla someone stringini somethingin sonuna ekletebiliriz. Something.append(“anan”) komutuyla tekli ekleme de yapabiliriz, bu komut da önceki gibi stringin sonuna ekleme yapar. İstediğimiz bir yere ekleme yapmak istersek something.insert(1, “Kelly”) komutunu kullanmalıyız. Something.remove(“anan”) komutuyla stringdeki “anan”’ı sildirebiliriz. Something.pop() komutuyla son elementi yok edebiliriz.

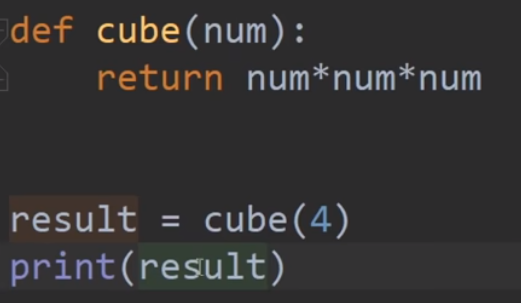
Something.index(“anan”) komutu ile anan’ın listede olup olmadığını öğrenebiliriz, var ise konumunu 0-n, yok ise sistem error verecektir. Something.count(“anan”) komutuyla ana’ın kaç defa kullanıldığını görebiliriz. Something.sort() komutuyla list değerlerini alfabetik sıraya göre dizdirebilir, Something.reverse() komutu ile olan listemizi tersten yazdırabiliriz.

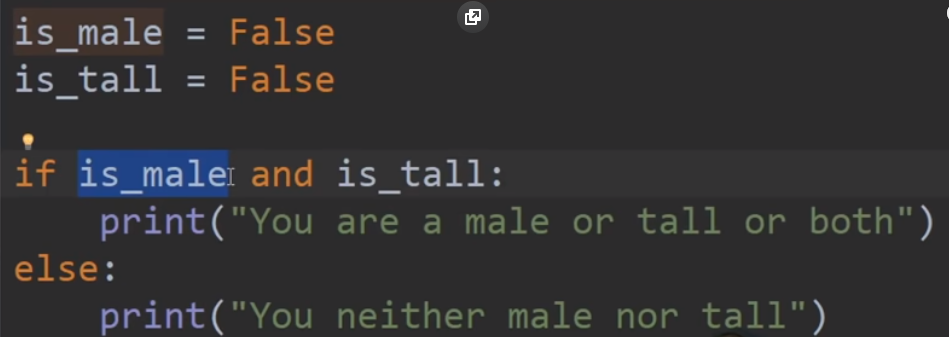
Something2 = something.copy() komutu ile something2’ye somethingin içeriğini kopyalayabiliriz.

Coordinates = (4, 5) 🡪 tuple is immutable. Bu örnekte coordinates[0] = 4 (koordinatı çektik). Tuple’lar list içerisinde de kullanılabilir. Tuple dediğimiz fonksiyon değiştirilemez değerler bütünündür, koordinat oluşturma buna güzel bir örnektir.

Function oluşturabilmek için öncelikle def komutuyla bir isim tanımlamalıyız. Örn. def something():, print(“cekomo):. Fonksiyon default olarak çağırılmayacağından someting() yazarak fonksiyonu çağırmamız gerekir. Fonksiyon içerisine değer tanımlayabiliriz (birden fazla). something(“anan”) ile anan komutunu function içine çağırabiliriz.

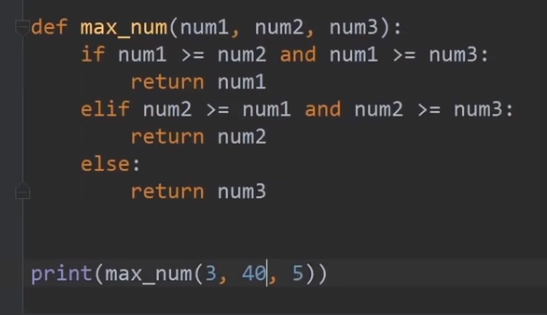
Return komutu kullandığımız functiondan değeri tekrar çekmek için kullanılır. Return komutundan sonra kullanılan function içi aksiyonlar yoksayılır. Örn:



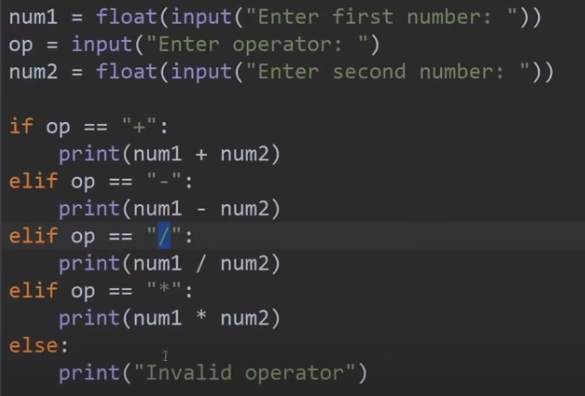


Else if komutunun kısaltması “elif” olarak ifade edilir. If komutu içerisinde and, and not, or, or not gibi komutlar kullanılabilir. Not komutu variableyi parantez ile coverlamalıdır.

Something = True/False komutu ile boolean oluşturabiliriz. Boolean, logic operationslara olanak tanır.

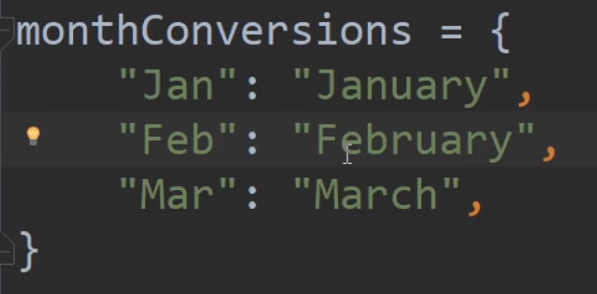


Something = input(“”) komutu ile command windowdan değer çekebiliriz. Örneğin bir sayı çekiyorsak float(input()) komutunu kullanarak sayı dışı değerlerde sistemi error verdirtebiliriz ki bug olmasın.

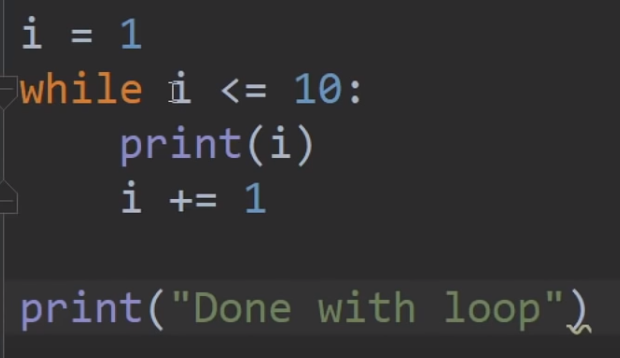


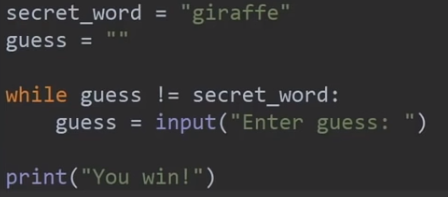
Şekil 1: Calculator

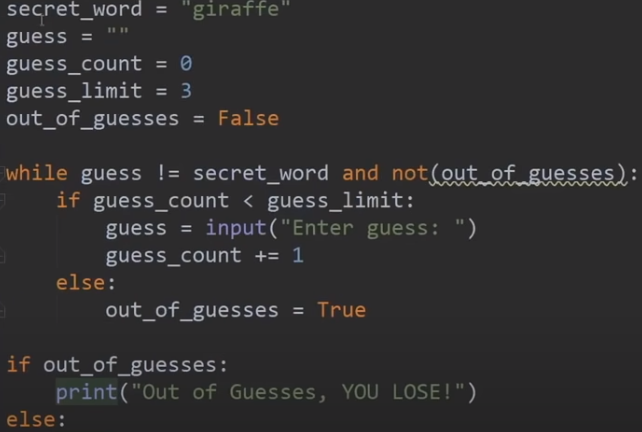
Dictionary oluşturmak için süslü parantez kullanılır. Örneğin anan = { (all the things) }, dictionary key terimlerden ve onların değerlerinden oluşur. Print(something[“”]) komutu ile istediğimiz keyi yazdırabiliriz, bunu yapmanın bir diğer yolu ise print(something.get(“”)) komutudur, bu komut ile aradığımız key dictionaryde yok ise None cevabını verir, bu cevap örneğin print(monthConversions.get(“anan”, “Invalid key term”)) komutu ile değiştirilebilir.



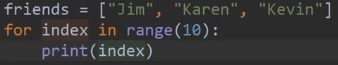
While loop, yapılan işlemin yanlış olmasına kadar devam eder.

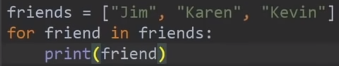




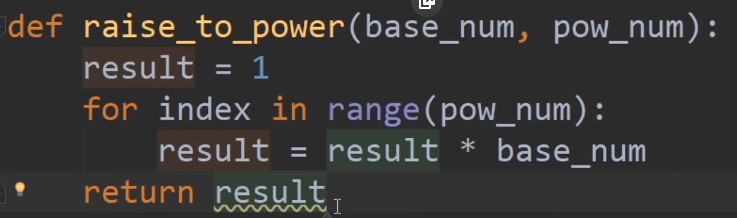


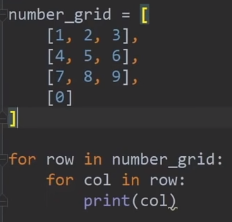
For something in “anan” 🡪 print(something) komutu ile anan kelimesinin flood yaparak yazdırabiliriz. Range komutu ile iki sayı arasındaki değerleri flood olarak yazdırabiliriz bu sayı tekli/çiftli olabilir (tekli olursa 0’dan başlar).





Üssel fonksiyon “^” yerine “\*\*” komutuyla elde edilebilir. Bunun yerine for loopuyla da exponent function kurabiliriz.

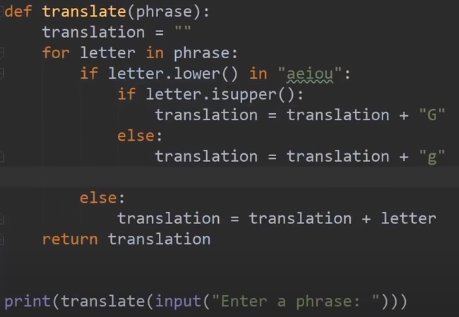




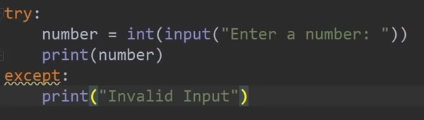
Köşeli parantez ile bir değer atayarak list oluşturabilir, bu listin içerisine de mini listler oluşturarak 2D array yaratabiliriz.



Figure 1: Komutu ile spesifik sayıyı bastırabiliriz, 0'dan başlanıldığını unutma (accessing elements in the list)

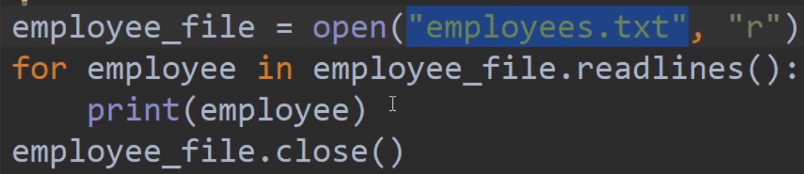


Try/except block kodun doğru çalışmaması halinde belirli bloğun yoksayılmasına dayanır. Bu komut program korur.

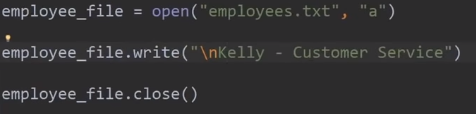


Except kısmına except ValueError: şeklinde error tipini belirterek kodumuzun outputta vereceği iletiyi ayarlayabiliriz. except ValueError as err: olarak kullanırsak sistemde olan ileti outputta basılır.

Open(“”) komutu ile dosya ismimizi türüyle beraber yazarak kod içerisine herhangi bir dosyayı çağırabiliriz, Open(“”, “r”/”w”/”a”/”r+”) komutlarıyla sırasıyla okuma/yazma/ekleme(değiştirmeden iliştirme)/(hem okuma hem yazma tam erişim) yaptırabiliriz. Dosyayı kullanabilmek için bir degree eşitlememiz gerekir, file = open(“”). Dosyayı kullandıktan sonra some.close() ile dosyayı kapamamız gerekir. print kullanarak some.read ile dosyanın tamamını outputa okutturabiliriz, .readable komutu dosyanın okunup okunamayacağını belirtir ve booleandır. Some.readline() komutu ile ilk satır okutulabilir, bu işlem tekrarlandığı takdirde diğer satırlar da kod tarafından alt satırlarda okunur, .readlines()[index] ile belirli bir satırı okutabiliriz, indexi çıkarırsak tüm linelar outputta okunur.



Yukarıdaki komut ile for komutu kullanarak bütün variableleri ayrı ayrı yazdırabiliriz.



“w” komutunu varolan dosya üzerinde kullanırsak overwrite olmaktadır, bu komut ile yeni bir dosya da oluşturabiliriz.

Import something komutu ile modül çağırabiliriz, bu modül something içerisindeki kodu kullanabilmemize olanak tanır. Modül içerisindeki herhangi bir modülü kullanabilmek için something.some() şeklinde yazılır. Modüller, built-in ve external module olarak ikiye ayrılır. External modulelar Lib dosyasında depolanır.

Pip komutu Python3 ile default olarak gelerek python modüllerini yüklemeye olanak tanır. Lib🡪site-packages klasörüne depolanan bu modüller cmd’ye pip install something yazarak internetten indirilebilirler. Aynı şekilde bir dosyayı silmek için pip uninstall something komutunu kullanabiliriz.

Class komutu çoğunlukla int/string/Boolean variablelarının yetersiz geldiği, farklı bir tanımlama biçimine sahip olabilecek oluşumlar için kullanılır. Indented olarak konumlandırılan kodlar class içerisinde sayılır. Farklı bir kod dosyasından bu class’I çağırmak için from somefile import something komutu kullanılır.

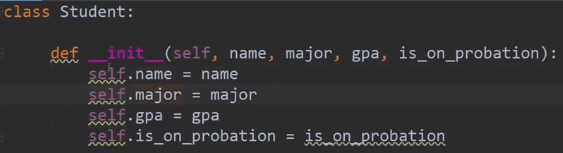
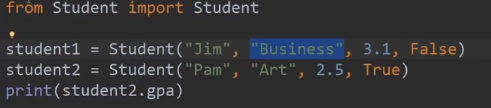


Figure 2: \_\_init\_\_ komutu classın içerisine yeni variableleri oluşturmak için çağırılır.



**Creating Multiple Choice Question** *(Koda tekrar bak)*

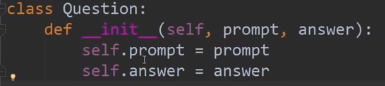
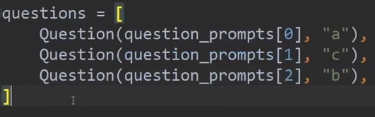
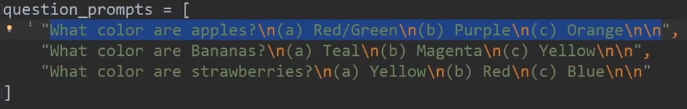
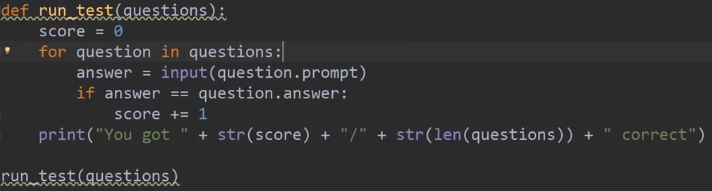
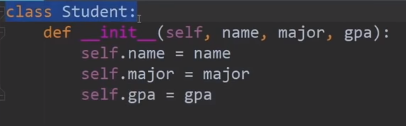


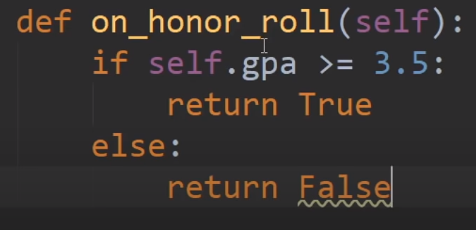
Figure 3: Başka bir dosyadan sağlanan Class

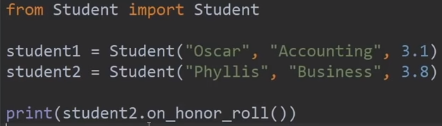




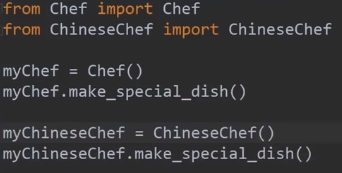
**“**\_\_init\_\_” fonksiyonunda self komutunu belirtmek zorunlu olsa da “self.some” diyerek parametre tanımlamamız gerekmez, bu bir kısayoldur ve kod içerisinde normal tanımlama yöntemi bu metod içerisinde de geçerlidir. Fakat örneğin aşağıdaki name parametresini \_\_init\_\_’e tanıtmak için aşağıdaki gibi veya name = …. Diye tanımlamamız gerekir. Eğer self parametresine ait herhangi bir değer belirtmeyeceksek örneğin \_\_init\_\_(self, name = “Cemil”) diyerek name belirtilmediği zamanlarda “Cemil” outputunu alabiliriz.

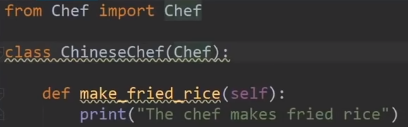






Inheritence fonksiyonu ile bir class içerisine başka bir class’ı çağırabiliriz. Bunu da class’ın içerisinde başa yazacak şekilde from some import some komutuyla oluştururuz.





Python Interpreter cmd içerisinde kullanılabilir. (Bende yok yüklemem lazım) Bu interpreter pythonu cmd içerisinde kullanabilmemizi sağlar. Herhangi bir komut vesaire denemek için kullanılabilir.

**Kaynak:** [**https://www.youtube.com/watch?v=rfscVS0vtbw&t=4225s**](https://www.youtube.com/watch?v=rfscVS0vtbw&t=4225s)