

| Python 101 | Output |
|---|------------------------------------|
| "a"+"b" | ab |
| str birlestirir "a" "b" | ab |
| str birlestirir "a"-"b" | Type Error |
| "a"/3 "a"*3 | Type error |
| Sayi kadaryanyana yazar len("veri bilimi") | 11 |
| Boyut inceler ver_bil="veri_bilimi" | Atama yapar |
| ver_bil.upper() | 'VERI_BILIMI' |
| Buyuk harf ver_bil.lower() | 'veri_bilimi' |
| Kucuk harf ver_bil.islower() | True |
| Kucuk mu? T/F ver_bil.isupper() | False |
| Buyuk mu? T/F Replace() | Str karekter degistirir. |
| ver_bil.replace("i","e") | 'vere_beleme' |
| i harflerini e harfine donusturdu. strip() | Istenmeyen karekteri kirpar |
| ver_bil.strip() | 'veri_bilimi' |
| basta ve sonda bosluk siler dir(ver_bil) | Alt çizgi olanlar oncelik |
| metotlari gozlemler ver_bil.capitalize() | 'Veri_bilimi' |
| ilk harfi buyuk harf yapar ver_bil.title() | 'Veri_Bilimi' |
| her kelimenin ilk harfi buyuk harf yazar. ver_bil[] | koseli parantez ile secim yapilir. |
| ver_bil[0] | 'v' |
| 0.elemanı secer ver_bil[5:9] | 'bili' |
| 5.elemandan 9.elemanı kadar | |

| Dictionary | Output |
|--|---|
| sozluk={"reg":{"regresyon":10,"msx":20},"loj":{"lojistik":5,"msxs":6}} | Sozluk atama |
| len(sozluk) | 2 |
| sozluk[0] | KeyError |
| sozluk["loj"] | {'lojistik': 5, 'msxs': 6} |
| sozluk["reg"]["msx"] | 20 |
| sozluk["gbm"]="gradient" | {'reg': {'regresyon': 10, 'msx': 20}, 'loj': {'lojistik': 5, 'msxs': 6}, 'gbm': 'gradient'} |
| eleman ekleme | |

| Donusumler | Output |
|---|--|
| input() | Kullanıcıdan bilgi alır. |
| input("sayi gir 1: ") + input("sayi gir 2: ") | 2 ayrı sayıyı kullanıcıdan alır. Str birleştirir |
| int(input("sayi gir 1: ")) + int(input("sayi gir 2: ")) | 2 ayrı sayıyı kullanıcıdan alır. Int birleştirir |
| int(11.0) Int donusturur | 11 |
| float(11) Float donusturur | 11.0 |
| str(12) str donusturur | '12' |
| print("veri","bilimi") ekrana yazdirma | veri bilimi |
| print("veri","bilimi",sep="_") | veri_bilimi |
| ?print | Ozellikler ogren |

| Tuple | Output |
|--|-------------|
| t=("ali","veli",1,2,[6,7,8]) t="a","b",1.3 t=("j" ,) t[2]=99 | Tuple atama |
| | Type error |

| Listeler | Output |
|---|---|
| notlar=[10,20,30] liste=["a","b",10,notlar,10] | Atama yapmak Liste icinde list |
| liste[2] liste[0:2] | 10 ['a', 'b'] |
| liste[3][2] | 30 |
| liste[2]="okul" | ['a', 'b', 'okul', [10, 20, 30], 10] |
| liste+"python" | ['a', 'b', 'okul', [10, 20, 30], 10, 'python'] |
| liste.append("spyder") | ['a', 'b', 'okul', [10, 20, 30], 10, 'spyder'] |
| liste.remove("spyder") | ['a', 'b', 'okul', [10, 20, 30], 10] |
| liste.insert(0,"hill") 0.index `hill` eklendi | ['hill', 'a', 'b', 'okul', [10, 20, 30], 10] |
| liste.insert(len(liste),"c") listenin sonuna `c` eklendi | ['hill', 'a', 'b', 'okul', [10, 20, 30], 10, 'c'] |
| liste.pop(4) listenin 4.eleman kaldırır. | ['hill', 'a', 'b', 'okul', 10, 'c'] |
| liste.count(10) | 2 |
| elemanda kac tane oldugunu sayar. liste_yedek=liste.copy() | List yedekler |
| liste.extend(["w","e","34"]) list icine yeni list ekler. | ['hill', 'a', 'b', 'okul', 10, 'c', 'w', 'e', '34'] |
| liste.index("a") a elemanı indeksi | 1 |
| liste.reverse() tersten yazar. | ['34', 'e', 'w', 'c', 10, 'okul', 'b', 'a', 'hill'] |
| liste.sort() elemanlari siralar | liste=[7,5,9,1,7] output=[1, 5, 7, 7,9] |
| liste.clear() listde tum elemanlari sildi | [] |

| Set | Output |
|--|--|
| s=["w","e",123,14,"f",123] set(s) k=["t","r","e",14] set(k) | Set atama {123, 14, 'e', 'f', 'w'} {14, 'e', 'r', 't'} |
| len(set(a)) | 5 |
| s.add("ile") eleman ekleme | {123, 14, 'e', 'f', 'ile', 'w'} |
| s.remove("ile") eleman kaldırma | {123, 14, 'e', 'f', 'w'} |
| k.difference(s) k-s | {'r', 't'} |
| k.symmetric_difference(s) ikisinde olmayan | {123, 'f', 'r', 't', 'w'} |
| k.intersection(s) k&s | {14, 'e'} |
| kesisim k.isdisjoint(s) | False |
| Kesisim bos kume mi? T/F k.issubset(s) | False |
| Altkumesi mi? T/F k.issuperset(s) | False |
| Kapsiyor mu? T/F | |