بسمه تعالی

معرفی فهرست ( **List**)

لیست در پایتون نوعی از داده است که میتواند هم‌زمان چند عنصر را در خود ذخیره کند. لیست ها در پایتون با استفاده از یک جفت براکت  [ ]  ایجاد میشوند. عنصرهایی که در لیست ذخیره میشوند با نام آیتم(item) شناخته میشوند.

ویژگی های مهم :

* لیست های پایتون مرتب(ordered) هستند به این معنی که ترتیب ورود اطلاعات در آنها حفظ میشود. شماره‌ای به داده‌ها به صورت خودکار داده می‌‌شود که از صفر آغاز شده و تا هر زمان که مقدار وارد لیست شود این اعداد به ترتیب اضافه می‌شوند. به این عدد در پایتون، ایندکس (INDEX) می‌گویند.
* لیست ها میتوانند هر نوع عنصر(شیء) را در خود ذخیره کنند.
* به آیتم های درون لیست میتوان با ایندکس دسترسی داشت.
* لیست های پایتون میتوانند تا عمق دلخواه تودرتو باشند.
* لیست ها تغییر پذیر هستند یعنی میتوان آیتم های آنها را تغییر داد.
* لیست ها پویا هستند به معنی که میتوان تعداد آیتم های آنها را کم یا زیاد کرد.

دستورات **List**

|  |  |  |
| --- | --- | --- |
| ردیف | **برنامه** | **توضیح** |
| 1 | **a = ["ali","reza","javad"]**  **b = [ 1 , 2 , 3 , 4 , 5 , 6 , 7, 8]**  **print(a)**  **print(type(a))** | ساخت لیست با استفاده از براکت [] و انتساب آن به یک متغییر |
| 2 | **a = [1,2,"ali",4,5,"rez"]**  **b = [15, 3.14, 'amin', [1, 2]]**  **c = []**  **print(a)** | لیست‌ها می‌توانند شامل انواع مختلف داده مانند عدد و رشته و حتی خود لیست باشند و یا لیست تهی باشد |
| 3 | **a = ["ali","reza","javad"]**  **print(a[0])**  **print(a[1])**  **print(a[-1])** | دسترسی به آیتم با اندیس‌گذاری :  الف\_اندیس گذاری معمول که از صفر شروع می‌شود.  ب\_ اندیس گذاری منفی که از آخرین عنصر لیست از سمت راست با عدد 1- شماره گذاری می‌کند. |
| 4 | **a = ["ali","reza","javad"]**  **print(a[1.5])**  **print(a[5])** | اگر ایندکس در لیست نباشد پیغام خطای IndexError  شماره ایندکس عدد(integer) است عدد غیر آن موجب خطای TypeError |
| 5 | **a = [1,2,3]**  **a[1]=55**  **print(a)**  **b = ["ali","reza","javad"]**  **b[2]="amin"**  **print(b)** | تغییر مقادیر آیتمهای لیست با انتساب مجدد  ( لیست ها تغییرپذیرهستند) |
| 6 | **a = ["ali","reza","javad"]**  **print(len(a))**  **n=len(a)**  **print(type(a))** | تعداد عضوهای لیست با تابع len |
| 7 | **a = [ 1 , 2 , 3 , 4 , 5 , 6 , 7, 8]**  **print (max(a))**  **print (min(a))** | مقدار max, min لیست |
| 8 | **a = [ 1 , 2 , 3 , 4 , 5 , 6 , 7, 8]**  **print (sum(a))** | جمع عناصر عددی لیست |
| 9 | **print([1, 2, 3] + [4, 5, 6])** | جمع دو لیست |
| 10 | **print(['reza'] \* 4)** | ضرب لیست |
| 11 | **[1, 10, 100] < [1, 11, 100]**  **[1, 10, 100] < [1, 10, 100]** | مقایسه دو لیست از ابتدا |
| 12 | **3 in [1, 2, 3]**  **4 in [1, 2, 3]**  **if 2 in [1, 2, 3]:**  **print('ok')** | عضویت در لیست |
| 13 | **for x in [1, 2, 3]:**  **print (x)** | پیمایش لیست با حلقه for با دو روش :  1\_ پیمایش مستقیم بر روی اعضا لیست  2\_ پیمایش با استفاده از متغیر شمارنده اندیس |
| **a=[1, 2, 3]**  **for i in range(0 , len(a)) :**  **print(a[i])** |
| 14 | **a = [ 1 , 2 , 3 , 4 , 5 , 6 , 7, 8]**  **b = a[0:3]**  **print(b)**  **b=a[:] # all of List**  **b=a[2:] # from 3’ item to end**  **b=a[:3] # from 0 to 4’ itrm** | برش یا slicing برای دسترسی به یک محدوده خاص از لیست ها با استفاده از شماره ایندکس و با علامت دو نقطه  [start:end]  نکته: برش تا قبل از end می باشد |
| 15 | **a = [ 0,1 , 2 , 3 , 4 , 5 , 6 , 7, 8]**  **b = a[0:6:2]**  **print(b)**  **b=a[::2]**  **print(b)**    **b=a[::-1]**  **print(b)** | برش با شروع و انتها همراه با گام پرش [start:end:step]  وقتی گام منفی باشد اندیس شروع باید بیشتر از اندیس پایان باشد |
| 16 | **cars = ['tondar', 'peykan','samand', 'Sourena', 'Peraid', 'peugeot']**  **cars[1:3] = ['quick', 'pe206']**  **print(cars)**  **cars[6:7] = ['quick', 'pe206']** | تغییر لیست با دستور برش |
| **متدهای لیست** | | |
| 17 | **a=[0,1,2,3]**  **a.append(6)**  **a.append(7)**  **print(a)** | List.append(item)  اضافه کردن یک آیتم جدید به آخر لیست از متد |
| 18 | **a = [1, 2, 3, 4, 5, 6, 7, 7]**  **b = [8, 9, 10]**  **a.extend(b)**  **print(a)**  **a.extend(range(3))** | List.extend(list\_2)  اضافه کردن لیست جدید به صورت تک به تک به انتهای لیست اصلی |
| 19 | **a = [1, 2, 3, 4]**  **b = a #** don'tCopy  **a.append(55)**  **print(b)**  **b is a** | ایجاد یک کپی از لیست :  وقتی از "=" استفاده شود در واقع لیست اصلی به یک نام جدید اختصاص داده میشود. یعنی هم لیست اصلی و هم نام جدید هر دو به یک لیست اشاره می کنند.در هر کدام تغییری ایجاد کنیم، بر روی هر دو تغییر ایجاد می‌شود که سه روش برای کپی دارد: |
| **a = [1, 2, 3, 4]**  **b = a[:] ; b is a** |
| **b = list(a) ; b is a** |
| **import copy**  **a = [1, 2, 3, 4]**  **b = copy.copy(a)**  **a.append(66)**  **print(b) ; b is a** |
| 20 | **a = [ 0 ,1 , 2 , 3 , 4 , 5 ,6]**  **a.insert(3,55)**  **print(a)** | list.insert(index, obj) (افزودن آیتم در ایندکس مشخص شده)  آیتم در خانه مورد نظر درج شده و عنصر فعلی این خانه و خانه‌های بعد از آن یکی به سمت راست شیفت داده می‌شوند. |
| 21 | **a = [1,4, 2, 3, 4, 5, 6,4]**  **b=a.count(4)**  **print(b)** | List.count(item)  شمارش تعداد دفعات تکرار یک آیتم در لیست |
| 22 | **a = [ 10 , 20 , 30 ,40]**  **b=a.index(30)**  **print(b)** | پیدا کردن اولین اندیس یک شی در لیست  List.index(value, [startIndex]) |
| 23 | **a = [ 10 , 20 , 30 ,40]**  **a.pop(2)**  **print(a)** | List.pop([index])  یک عنصر رو با اندیس مشخص شده حذف می کند اگر ایندکس مشخص نکنیم آخرین عنصر از لیست حذف میشود |
| 24 | **a = [ 10 , 20 , 30 ,40]**  **a.remove(30)**  **print(a)** | remove(value) list.  یک عنصر با مقدار مشخص شده حذف میکند |
| 25 | **a = list(range(10))**  **del a[ : :2] # a = [1, 3, 5, 7, 9]**  **print(a)**  **del a[-1] # a = [1, 3, 5, 7]**  **del a[ : ]** | حذف  کردن آیتم از لیست با استفاده از تکنیک برش |
| 26 | **a = [ 1 , 2 , 3 , 4 ,5]**  **a.reverse()**  **print(a)** | معکوس کردن لیست |
| 27 | **a = [ 10 , 2 , 9 , 8 ,8]**  **a.sort()**  **print(a)** | مرتب کردن لیست |
| 28 | **a.clear( )** | پاک کردن لیست |
| 29 | **b = [**  **[1,2,3],**  **[10,20,30],**  **[50,60,70] ]**    **print( b[0][0], b[1][0], b[2][2] )** | لیست تو در تو : یک لیست b که دارای 3 لیست داخلی می باشد برای دسترسی به آیتم‌ها باید دو شماره ایندکس را وارد کنیم. اولین مورد شماره ایندکس لیست اصلی و دومین ایندکس مربوط به یک عضو در لیست انتخابی. |
| 30 | **row = 3**  **col = 4**  **matrix = [[0, 0, 0, 0],**  **[0, 0, 0, 0],**  **[0, 0, 0, 0]]**  **for r in range(row):**  **for c in range(col):**  **matrix[r][c] = r\*c**  **print(matrix)** | ایجاد ماتریس با لیست تو در تو  و پیمایش با حلقه های for |