

برنامه نویسی به زبان

پایتون

مرجع کامل

تالیف

مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا
مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از
کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع
کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرید

hadikiamarsi@gmail.com

لازم به ذکر است کلیه حقوق مادی و معنوی این اثر برای مولف محفوظ می
باشد و هرگونه کپی برداری و استفاده از محتویات این کتاب به هر نوعی تحت
پیگرد قانونی قرار می گیرد

کیا مدتی

فصل هشتم

در این فصل مطالب زیر را خواهید آموخت

لیست ها در زبان برنامه نویسی پایتون (python)

دسترسی به عناصر لیست ها

به روز کردن لیست ها

حذف عناصر از لیست

عملگرهای لیست ها

اندیس و ماتریکس ها

متدها و توابع پیش ساخته لیست ها

کیا مدتی

لیست ها

در زبان برنامه نویسی پایتون (python) 6 نوع ساختمان داده وجود دارد که یکی از این ساختمان داده ها لیست ها می باشد . در زبان برنامه نویسی پایتون (python) برای لیست ها عملگرها و توابع و متدهای پیش ساخته ای وجود دارد که به طور مفصل در این فصل توضیح داده خواهند شد

لیست ها در زبان برنامه نویسی پایتون (python)

لیست ها یکی از انواع داده ای پرکاربرد در زبان برنامه نویسی پایتون (python) می باشد که مشخصه شناسایی اون براکت های باز و بسته می باشد

در زبان برنامه نویسی پایتون (python) برای ایجاد یک لیست خالی به شکل زیر عمل می نمایم

```
list1 = list()
```

در مثال زیر نحوه مقدار اولیه دادن به لیست ها نشان داده شده است

```
list1 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5 ];  
list3 = ["a", "b", "c", "d"]
```

مانند رشته اندیس عناصر در لیست ها از صفر شروع می گردد

دسترسی به عناصر لیست ها

برای دسترسی به عناصر لیست ها از براکت های باز و بسته به همراه شماره (اندیس) عنصر در جلوی نام لیست استفاده می گردد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python  
  
list1 = ['physics', 'chemistry', 1997, 2000];  
list2 = [1, 2, 3, 4, 5, 6, 7 ];  
print "list1[0]: ", list1[0]  
print "list2[1:5]: ", list2[1:5]
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
list1[0]: physics  
list2[1:5]: [2, 3, 4, 5]
```

به روز کردن لیست ها

شما می توانید عناصر لیست ها را به صورت تکی و گروهی بروزرسانی نمایید برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python

list = ['physics', 'chemistry', 1997, 2000];
print "Value available at index 2 : "
print list[2]
list[2] = 2001;
print "New value available at index 2 : "
print list[2]
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Value available at index 2 :
1997
New value available at index 2 :
2001
```

حذف عناصر از لیست

می توان عناصر را از لیست ها حذف نمود برای این کار از تابع `del` استفاده می گردد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python

list1 = ['physics', 'chemistry', 1997, 2000];
print list1
del list1[2];
print "After deleting value at index 2 : "
print list1
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
['physics', 'chemistry', 1997, 2000]
After deleting value at index 2 :
['physics', 'chemistry', 2000]
```

عملگرهای لیست ها

لیست ها به عملگرهای `+` و `*` همانند رشته ها پاسخ می دهند به این معنا که `+` برای الحاق و `*` برای تکرار استفاده می گردند برای آشنایی بیشتر با این درس به مثال های زیر توجه نمایید

کد پایتون	خروجی	توضیحات
<code>len([1, 2, 3])</code>	3	بدست آوردن طول لیست
<code>[1, 2, 3] + [4, 5, 6]</code>	<code>[1, 2, 3, 4, 5, 6]</code>	الحاق لیست ها
<code>['Hi!'] * 4</code>	<code>['Hi!', 'Hi!', 'Hi!', 'Hi!']</code>	تکرار عناصر لیست ها
<code>3 in [1, 2, 3]</code>	True	بررسی وجود یک عنصر در لیست
<code>for x in [1, 2, 3]: print x,</code>	1 2 3	حلقه ها در لیست ها

اندیس و ماتریکس ها

در زبان برنامه نویسی پایتون (python) ماتریکس ها نیز بوسیله لیست ها ایجاد می گردند برای ایجاد ماتریکس باید در ابتدا مفهوم و طرز کار با اندیس ها را به خوبی یاد بگیرید . برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
L = ['spam', 'Spam', 'SPAM!']
```

پایتون کد	خروجی	توضیحات
<code>L[2]</code>	SPAM!	با شروع اندیس از صفر
<code>L[-2]</code>	Spam	اندیس منفی از آخر لیست به اول می باشند
<code>L[1:]</code>	<code>['Spam', 'SPAM!']</code>	دریافت چند عنصر از لیست

متدها و توابع پیش ساخته لیست ها

زبان برنامه نویسی پایتون (python) دارای توابع پیش ساخته ای برای کار با لیست ها می باشد که در زیر به همراه توضیحات آورده شده اند

تابع ها به همراه توضیحات

cmp(list1, list2)

از این متد برای مقایسه عناصر دو لیست استفاده می گردد

```
#!/usr/bin/python
list1, list2 = [123, 'xyz'], [456, 'abc']
print cmp(list1, list2)
print cmp(list2, list1)
list3 = list2 + [786];
print cmp(list2, list3)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
-1
1
-1
```

len(list)

این متد تعداد عناصر یک لیست را بر می گرداند

```
#!/usr/bin/python
list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']
print "First list length : ", len(list1)
print "Second list length : ", len(list2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
First list length : 3
Second list length : 2
```


max(list)

این متد بزرگترین عنصر لیست را بر می گرداند

```
#!/usr/bin/python
list1, list2 = [123, 'xyz', 'zara', 'abc'], [456, 700, 200]
print "Max value element : ", max(list1)
print "Max value element : ", max(list2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Max value element : zara
Max value element : 700
```

min(list)

این متد کوچکترین عنصر لیست را بر می گرداند

```
#!/usr/bin/python
list1, list2 = [123, 'xyz', 'zara', 'abc'], [456, 700, 200]
print "min value element : ", min(list1)
print "min value element : ", min(list2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
min value element : 123
min value element : 200
```

list(seq)

از این متد برای تبدیل تاپل به لیست استفاده می گردد

```
#!/usr/bin/python
aTuple = (123, 'xyz', 'zara', 'abc');
aList = list(aTuple)
print "List elements : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
List elements : [123, 'xyz', 'zara', 'abc']
```

زبان برنامه نویسی پایتون (python) دارای متدهای پیش ساخته ای برای کار با لیست ها می باشد که در زیر به همراه توضیحات آورده شده اند

متدها به همراه توضیحات

list.append(obj)

از این متد برای اضافه کردن عناصر به لیست استفاده می گردد

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc'];
aList.append( 2009 );
print "Updated List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Updated List : [123, 'xyz', 'zara', 'abc', 2009]
```

list.count(obj)

این متد تعداد تکرار یک عنصر در لیست را بر می گرداند

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc', 123];
print "Count for 123 : ", aList.count(123)
print "Count for zara : ", aList.count('zara')
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Count for 123 : 2
Count for zara : 1
```

list.extend(seq)

از این متد برای ادغام دو لیست در هم استفاده می گردد

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc', 123];
bList = [2009, 'manni'];
aList.extend(bList)
print "Extended List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Extended List : [123, 'xyz', 'zara', 'abc', 123, 2009, 'manni']
```

list.index(obj)

از این متد برای جستجو و برگرداندن اندیس عنصر لیست استفاده می گردد

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc'];
print "Index for xyz : ", aList.index( 'xyz' )
print "Index for zara : ", aList.index( 'zara' )
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Index for xyz : 1
Index for zara : 2
```

list.insert(index, obj)

از این متد برای درج یک عنصر در لیست استفاده می گردد

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc']
aList.insert( 3, 2009)
```

```
print "Final List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Final List : [123, 'xyz', 'zara', 2009, 'abc']
```

list.pop(obj=list[-1])

این متد یک عنصر را از لیست حذف می نماید و آن را بر می گرداند

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc'];
print "A List : ", aList.pop()
print "B List : ", aList.pop(2)
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
A List : abc
B List : zara
```

list.remove(obj)

از این متد برای حذف عناصر از لیست استفاده می گردد

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc', 'xyz'];
aList.remove('xyz');
print "List : ", aList
aList.remove('abc');
print "List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
List : [123, 'zara', 'abc', 'xyz']
List : [123, 'zara', 'xyz']
```

list.reverse()

از این متد برای معکوس کردن عناصر لیست بکار می رود

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc', 'xyz'];
aList.reverse();
print "List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
List : ['xyz', 'abc', 'zara', 'xyz', 123]
```

list.sort([func])

این متد برای مرتب کردن عناصر لیست بکار می رود

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc', 'xyz'];
aList.sort();
print "List : ", aList
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
List : [123, 'abc', 'xyz', 'xyz', 'zara']
```