

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

فصل نهم

فادی پیامدستی

در این فصل مطالب زیر را خواهید آموخت

تاپل ها (tuples) در پایتون (python)

دسترسی به عناصر در تاپل ها (tuples)

بروزرسانی تاپل ها (tuples)

حذف عناصر از تاپل (tuple)

عملگرها در تاپل (tuple)

ماتریس با استفاده از تاپل (tuple)

جدا کننده ها بدون پرانتز

توابع پیش ساخته تاپل ها (tuples)

کیا مدتی

تاپل ها (tuples) در پایتون (python)

تاپل ها (tuples) از انواع داده ای در زبان برنامه نویسی پایتون (python) می باشند که مانند لیست ها از تعدادی داده که پشت سر هم قرار گرفته اند تشکیل شده اند . تفاوت داده ها با تاپل ها (tuples) اینست که بر خلاف لیست ها تاپل ها از انواع داده ای تغییر ناپذیر هستند و نشانه شناسایی آن ها پرانتزها (parentheses) می باشد

نحوه مقدار اولیه دادن به تاپل ها (tuples) در زیر نشان داده شده است

```
tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5 );
tup3 = "a", "b", "c", "d";
```

برای ایجاد یک تاپل (tuple) خالی به روش زیر عمل می نمایم

```
tup1 = ();
```

برای ایجاد یک تاپل (tuple) شامل فقط یک عنصر حتما باید شامل یک علامت کاما (comma) باشد مانند مثال زیر

```
tup1 = (50,);
```

مانند رشته ها اندیس ها (tuples) از صفر شروع می گردد

دسترسی به عناصر در تاپل ها (tuples)

برای دسترسی به عناصر در تاپل ها (tuples) باید از براکت باز و بسته به همراه اندیس استفاده کرد. برای آشنایی بیشتر به مثال زیر توجه نمایید

```
#!/usr/bin/python
tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5, 6, 7 );
print "tup1[0]: ", tup1[0];
print "tup2[1:5]: ", tup2[1:5];
```

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

```
tup1[0]: physics
tup2[1:5]: [2, 3, 4, 5]
```

بروزرسانی تاپل ها (tuples)

عناصر تاپل ها (tuples) را نمی شود بروزرسانی کرد تنها می توان آن ها را باهم ترکیب و تاپل جدید ساخت برای آشنایی بیشتر به مثال زیر توجه نمایید

```
#!/usr/bin/python

tup1 = (12, 34.56);
tup2 = ('abc', 'xyz');

# Following action is not valid for tuples
# tup1[0] = 100;

# So let's create a new tuple as follows
tup3 = tup1 + tup2;
print tup3;
```

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

```
(12, 34.56, 'abc', 'xyz')
```

حذف عناصر از تاپل (tuple)

پاک کردن یک عنصر به تنهایی از تاپل (tuple) ممکن نیست برای روشن شدن این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python

tup = ('physics', 'chemistry', 1997, 2000);
print tup;
del tup;
print "After deleting tup : ";
print tup;
```

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

```
('physics', 'chemistry', 1997, 2000)
After deleting tup :
Traceback (most recent call last):
  File "test.py", line 9, in <module>
```

```
print tup;
NameError: name 'tup' is not defined
```

عملگرها در تاپل (tuple)

تاپل ها (tuples) مانند رشته ها به عملگرهای * و + واکنش نشان می دهند برای آشنایی بیشتر با این درس به مثال های موجود در جدول زیر توجه نمایید

کد پایتون	خروجی	توضیح
<code>len((1, 2, 3))</code>	3	طول تاپل
<code>(1, 2, 3) + (4, 5, 6)</code>	<code>(1, 2, 3, 4, 5, 6)</code>	الحاق دو تاپل
<code>('Hi!',) * 4</code>	<code>('Hi!', 'Hi!', 'Hi!', 'Hi!')</code>	تکرار عناصر تاپل
<code>3 in (1, 2, 3)</code>	True	بررسی عضویت در تاپل
<code>for x in (1, 2, 3): print x,</code>	1 2 3	حلقه های تکرار برای تاپل ها

ماتریس با استفاده از تاپل (tuple)

برای پیاده سازی یک ماتریس با استفاده از تاپل (tuple) باید در ابتدا به خوبی مفهوم اندیس در تاپل ها را بدانید

```
L = ('spam', 'Spam', 'SPAM!')
```

کد پایتون	خروجی	توضیحات
<code>L[2]</code>	<code>'SPAM!'</code>	اندیس تاپل ها از صفر شروع می گردد
<code>L[-2]</code>	<code>'Spam'</code>	اندیس منفی در تاپل
<code>L[1:]</code>	<code>['Spam', 'SPAM!']</code>	استخراج چند عنصر از تاپل

جدا کننده ها بدون پرانتز

برای جدا سازی عناصر تاپل (tuple) از علامت کاما (comma) استفاده می گردد حتی عناصر تاپل (tuple) را بدون پرانتزها (parentheses) نیز می توان بکار برد برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python
print 'abc', -4.24e93, 18+6.6j, 'xyz';
x, y = 1, 2;
print "Value of x , y : ", x,y;
```

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

```
abc -4.24e+93 (18+6.6j) xyz
Value of x , y : 1 2
```

توابع پیش ساخته تاپل ها (tuples)

در زبان برنامه نویسی پایتون (python) توابع پیش ساخته ای برای کار بر روی تاپل ها (tuples) وجود دارد که در زیر به همراه توضیح آمده اند

توابع با توضیحات

```
cmp(tuple1, tuple2)
```

مقایسه دو تاپل (tuple) را انجام می دهد

```
#!/usr/bin/python
tuple1, tuple2 = (123, 'xyz'), (456, 'abc')
print cmp(tuple1, tuple2)
print cmp(tuple2, tuple1)
tuple3 = tuple2 + (786,);
print cmp(tuple2, tuple3)
```

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


```
-1  
1  
-1
```

len(tuple)

تعداد عناصر تاپل (tuple) را بر می گرداند

```
#!/usr/bin/python  
  
tuple1, tuple2 = (123, 'xyz', 'zara'), (456, 'abc')  
print "First tuple length : ", len(tuple1)  
print "Second tuple length : ", len(tuple2)
```

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

```
First tuple length : 3  
Second tuple length : 2
```

max(tuple)

بزرگترین عنصر در تاپل (tuple) را بر می گرداند

```
#!/usr/bin/python  
  
tuple1, tuple2 = (123, 'xyz', 'zara', 'abc'), (456, 700, 200)  
print "Max value element : ", max(tuple1)  
print "Max value element : ", max(tuple2)
```

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

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

min(tuple)

کوچکترین عنصر در تاپل (tuple) را بر می گرداند

```
#!/usr/bin/python
tuple1, tuple2 = (123, 'xyz', 'zara', 'abc'), (456, 700, 200)
print "min value element : ", min(tuple1)
print "min value element : ", min(tuple2)
```

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

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

tuple (seq)

برای تبدیل لیست به تاپل بکار می رود

```
#!/usr/bin/python
aList = [123, 'xyz', 'zara', 'abc']
aTuple = tuple(aList)
print "Tuple elements : ", aTuple
```

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

```
Tuple elements : (123, 'xyz', 'zara', 'abc')
```