

## Tuples

PYnative.com

### Tuples in Python

T = ( 20, 'Jessa', 35.75, [30, 60, 90] )

↑  
T[0]

↑  
T[1]

↑  
T[2]

↑  
T[3]

- ✓ **Ordered:** Maintain the order of the data insertion.
- ✓ **Unchangeable:** Tuples are immutable and we can't modify items.
- ✓ **Heterogeneous:** Tuples can contains data of types
- ✓ **Contains duplicate:** Allows duplicates data

آیا تا به حال با **tuple**ها در پایتون کار کرده‌ای؟ اگر نه، اینجا وقتشه که به دنیای **tuple**ها وارد بشی! شاید به نظر بیاد که **tuple**ها همونطور که از اسمشون پیداست، یه ساختار ساده و بی‌دردسر باشن، اما باید بدونی که این ساختار داده‌ای یکی از قدرتمندترین ابزارهایی هست که پایتون در اختیارت می‌ذاره.

فرض کن می‌خواهی اطلاعاتی رو ذخیره کنی که هرگز نباید تغییر کنه. مثلاً اطلاعاتی مثل تاریخ تولد یه فرد یا یک سری داده ثابت توی برنامه‌ها که نباید تغییر کنه. اینجاست که **tuple**ها به کار می‌آیند!

تو **tuple**ها می‌تونی داده‌ها رو توی یه جعبه غیرقابل تغییر بریزی، یعنی هر بار که یه **tuple** رو ساختی، نمی‌تونی محتویات اون رو تغییر بدی. خیلی شبیه یه جعبه قفل شده می‌مونه که باز هم می‌تونی ازش استفاده کنی، اما نمی‌تونی داخلش رو عوض کنی!

### تعریف tuple و نحوه استفاده از آن‌ها:

در پایتون، **tuple**ها یه ساختار داده‌ای برای ذخیره داده‌ها هستن. این ساختار می‌تونه شامل هر نوع داده‌ای باشه: از رشته‌ها (strings) گرفته تا اعداد صحیح (integers)، اعشاری (floats) و حتی بولین‌ها (Booleans).

پس این یعنی tupleها می‌تونن هر چیزی رو تو خودشون ذخیره کنن!

### چطور یک tuple بسازیم؟

برای ساختن یه tuple، کافیه داده‌ها رو داخل پرانتز گرد ( ) بذاریم. البته، tupleها مثل لیست‌ها نیستن که همگی تغییر پذیر باشن، بلکه غیرقابل تغییر هستن!

مثال:

```
1 | mytuple = ("apple", "banana", "cherry")
```

این کد یک tuple ساخته که شامل سه آیتم هست: "apple"، "banana" و "cherry".

### دسترسی به آیتم‌های tuple:

حالا که یه tuple ساختیم، چطور می‌تونیم به آیتم‌های داخلش دسترسی داشته باشیم؟ ساده است! از ایندکس‌ها استفاده می‌کنیم. ایندکس‌ها در پایتون از **صفر** شروع میشن.

مثال:

```
1 | mytuple = ("apple", "banana", "cherry")
2 | print(mytuple[1])
```

تو اینجا به آیتم با ایندکس 1 دسترسی پیدا کردیم و نتیجه "banana" بود.

### ایندکس‌های منفی:

یه ویژگی جالب در مورد tupleها اینه که می‌تونیم از ایندکس‌های منفی استفاده کنیم. یعنی اگر بخواهیم از انتهای tuple شروع کنیم، می‌تونیم از ایندکس -1 برای دسترسی به آخرین آیتم استفاده کنیم!

مثال:

```
1 | print(mytuple[-1])
```

## حلقه زدن روی tuple ها:

شاید بخواهیم تمام آیتم‌های یه tuple رو چاپ کنیم. در این صورت می‌تونیم از یه حلقه استفاده کنیم. ساده‌ترین راه برای این کار استفاده از **for loop** هست.

### مثال:

```
1 | for fruit in mytuple:  
2 |     print(fruit)
```

این کد به ترتیب همه‌ی آیتم‌های داخل tuple رو چاپ می‌کنه.

## حلقه با ایندکس‌ها:

همچنین می‌تونیم با استفاده از **range()** و **len()** به ایندکس‌های tuple دسترسی پیدا کنیم و آیتم‌ها رو بر اساس ایندکس چاپ کنیم.

### مثال:

```
1 | for i in range(len(mytuple)):  
2 |     print(mytuple[i])
```

## جابه‌جایی مقادیر در تاپل‌ها با unpacking :

حتما می‌دونید که وقتی از **=** برای تساوی دو تاپل استفاده می‌کنید، مقادیر هر عضو از یکی به دیگری منتقل میشه.

### مثال:

```

1  >>> a, b, c = 10, 20, 30
2  >>> a
3  10
4  >>> b
5  20
6  >>> c
7  30
8  >>> a, b = b, a
9  >>> a
10 20
11 >>> b
12 10

```

## نکات جذاب و کاربردی در تاپل‌ها

- برای ساختن تاپل خالی، از `()` استفاده کنید

مثال:

```

1  t = ()
2  print(len(t))

```

برای ساختن یه تاپل یک عضوی، حتما باید از کاما `,` استفاده کنید

مثال:

```

1  t = 1,
2  print(len(t))

```

## ادغام و ضرب tupleها:

حالا که می‌دونیم چطور با tupleها کار کنیم، بریم سراغ یکی دیگه از ویژگی‌های جالبشون. می‌تونیم چند tuple رو با استفاده از `+` به هم وصل کنیم و حتی با `*` یه tuple رو چند بار تکرار کنیم!

مثال:

```
1 | tuple1 = ("a", "b", "c")
2 | tuple2 = (1, 2, 3)
3 | tuple3 = tuple1 + tuple2
4 | print(tuple3)
```

مثال:

```
1 | fruits = ("apple", "banana", "cherry")
2 | mytuple = fruits * 2
3 | print(mytuple)
```

## روش‌های کاربردی در tupleها:

حالا می‌خواهیم درباره دو روش مفید و built-in پایتون که مخصوص tupleها هستند صحبت کنیم. این روش‌ها به شما کمک می‌کنن که راحت‌تر با tupleها کار کنید.

### 1. count():

این روش تعداد دفعاتی که یک آیتم خاص در tuple وجود داره رو می‌شماره.

مثال:

```
1 | mytuple = ("apple", "banana", "cherry", "apple")
2 | print(mytuple.count("apple"))
```

### 2. index():

این روش ایندکس اولین موقعیتی که یک آیتم در tuple پیدا می‌شه رو برمی‌گردونه.

مثال:

```
1 | print(mytuple.index("banana"))
```

## چند نکته جالب درباره tupleها:

- **غیرقابل تغییر بودن (Immutable):** توی tupleها نمی‌تونید داده‌ها رو تغییر بدید. اما همیشه می‌تونید یه نسخه جدید از tuple بسازید!
  - **ذخیره‌سازی داده‌های ثابت:** اگر داده‌هایی دارید که نباید تغییر کنن، بهترین گزینه استفاده از tupleهاست.
  - **کاربرد در امنیت:** چون tupleها تغییر ناپذیر هستند، می‌تونید مطمئن باشید که داده‌ها به اشتباه تغییر نمی‌کنن. این ویژگی باعث میشه که tupleها برای کار با داده‌های حساس و امنیتی مناسب باشن.
- تو این جزوه یاد گرفتیم که **tupleها** چطور در پایتون کار می‌کنن. از ساختن tuple تا دسترسی به آیتم‌ها، از حلقه زدن تا ادغام و ضرب tupleها، و حتی روش‌های کاربردی مثل `count()` و `index()` رو هم بررسی کردیم.