

پایتون

به زبان ساده

فصل اول: مبانی زبان پایتون

۷.....	پایتون چیست.....
۸.....	دانلود و نصب Python 3.7.....
۱۱.....	ساخت یک برنامه ساده.....
۱۷.....	توضیحات.....
۱۸.....	کاراکترهای کنترلی.....
۱۹.....	متغیر.....
۲۱.....	انواع داده.....
۲۱.....	استفاده از متغیرها.....
۲۵.....	عبارات و عملگرها.....
۲۶.....	عملگرهای ریاضی.....
۲۷.....	عملگرهای تخصیصی (جایگزینی).....
۲۸.....	عملگرهای مقایسه‌ای.....
۳۰.....	عملگرهای منطقی.....
۳۷.....	عملگرهای خاص.....
۳۸.....	گرفتن ورودی از کاربر.....
۳۹.....	ساختارهای تصمیم.....
۳۹.....	دستور if.....
۴۲.....	دستور if..else.....
۴۳.....	دستور if..elif..else.....
۴۴.....	دستور if تو در تو.....
۴۵.....	استفاده از عملگرهای منطقی.....
۴۷.....	عملگر شرطی.....
۴۹.....	تکرار.....
۵۰.....	حلقه While.....
۵۱.....	حلقه for.....
۵۲.....	حلقه‌های تو در تو (Nested Loops).....

۵۳	خارج شدن از حلقه با استفاده از break, continue و pass
۵۴	تابع
۵۵	مقدار برگشتی از یک تابع
۵۷	پارامترها و آرگومان‌ها
۵۹	آرگومان‌های کلمه کلیدی (Keyword Arguments)
۶۰	آرگومان‌های متغیر
۶۲	محدوده متغیر
۶۴	پارامترهای پیشفرض
۶۴	بازگشت (Recursion)
۶۶	توابع داخلی
۶۷	Decorator
۶۸	عبارات لامبدا (Lambda expressions)
۶۹	توابع از پیش تعریف شده (Built-in Function)
۷۱	توابعی خاص (Special Methods)



از سایر کتاب های **یونس ابراهیمی** در لینک های زیر دیدن فرمایید

<https://bit.ly/2kKGxYJ>

www.w3-farsi.com/product

بی شک این اثر، خالی از اشکال نیست و از شما خوانندگان عزیز می‌خواهم که با نظرات و پیشنهادات خود بنده را در تکمیل و رفع نواقص آن از طریق پست الکترونیکی younes.ebrahimi.1391@gmail.com یاری بفرمایید.

برای مطالعه مطالب بیشتر به سایت www.w3-farsi.com مراجعه فرمایید.

راه‌های ارتباط با نویسنده

وب سایت: www.w3-farsi.com

لینک تلگرام: https://telegram.me/ebrahimi_younes

ID تلگرام: @ebrahimi_younes

پست الکترونیکی: younes.ebrahimi.1391@gmail.com

فصل اول

مبانی زبان پایتون

پایتون چیست

پایتون (Python) یک زبان برنامه‌نویسی همه منظوره، شیء‌گرا و متن باز است که توسط خودو فان راسام (Guido van Rossum) در سال ۱۹۹۱ در کشور هلند طراحی شد. این زبان از زبان‌های برنامه‌نویسی مفسر بوده و به صورت کامل یک زبان شیء‌گرا است که به زبانهای تفسیری Perl و Ruby شباهت دارد و از مدیریت خودکار حافظه استفاده می‌کند.

پایتون، کد باز (Open Source) است، زبانی که گوگل و یا یاهو از آن به عنوان یکی از اصلی‌ترین ابزارهای توسعه استفاده می‌کنند. برنامه‌های پایتون مثل PHP قابل اجرا روی اغلب سیستم عامل‌هاست. پایتون، دستور زبانی شبیه گفتار ساده‌ی انگلیسی دارد و با دارا بودن ۳۳ کلمه کلیدی جزء ساده‌ترین زبان‌ها است.

سادگی و خوانایی از ویژگی‌های بارز زبان برنامه‌نویسی پایتون است، آنچنان ساده که حتی کودکان نیز قادر به آموختن آن هستند و قدرت در کنار این سادگی و خوانایی، معجزه پایتون می‌باشد. از نگاه هر برنامه‌نویسی، برنامه‌های پایتون مجموعه‌ای از کدهای زیبا هستند، بدون هیچ آشفتگی و پیچیدگی. جالب است بدانید مایکروسافت نیز این زبان را با نام IronPython در تکنولوژی .Net. خود گنجانده است. هم اکنون پایتون در شرکت‌ها و سازمان‌های بزرگی مثل ناسا و گوگل و یاهو و ... به صورت گسترده مورد استفاده قرار می‌گیرد. تا کنون نسخه‌های مختلفی از این زبان ارائه شده است که لیست آنها را در جدول زیر مشاهده می‌کنید:

نسخه	تاریخ پیدایش
Python 1.0	January 1994
Python 1.2	April 10, 1995
Python 1.3	October 12, 1995
Python 1.4	October 25, 1996
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003
Python 2.4	November 30, 2004

September 19, 2006	Python 2.5
October 1, 2008	Python 2.6
July 3, 2010	Python 2.7
December 3, 2008	Python 3.0
June 27, 2009	Python 3.1
February 20, 2011	Python 3.2
September 29, 2012	Python 3.3
March 16, 2014	Python 3.4
September 13, 2015	Python 3.5
December 23, 2016	Python 3.6
October 20, 2018	Python 3.7

حال که با پایتون به طور مختصر آشنا شدید، در درس‌های بعد در مورد این زبان برنامه‌نویسی بیشتر توضیح می‌دهیم.

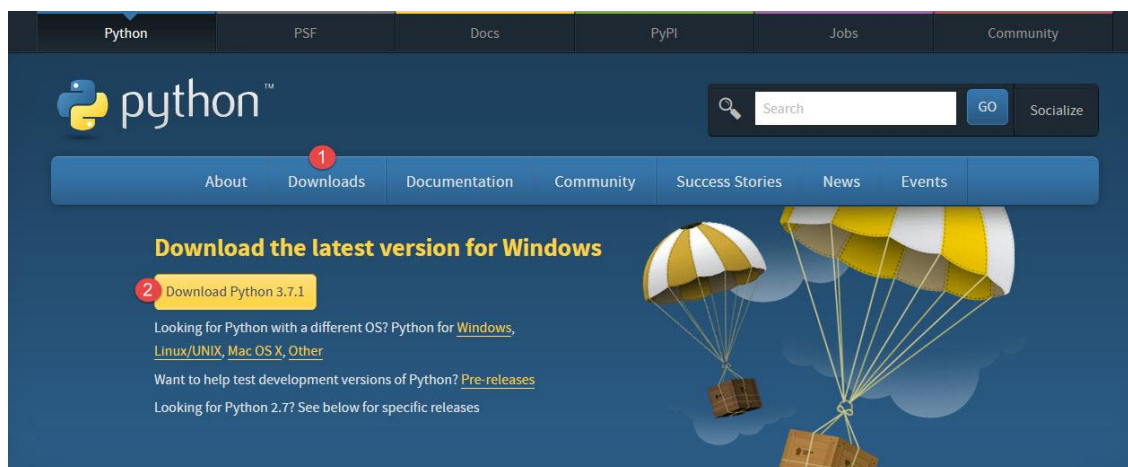
W3-farsi.com تخصصی‌ترین سایت آموزش پایتون در ایران

دانلود و نصب Python 3.7

برای برنامه‌نویسی به زبان‌های مختلف محیط‌های توسعه‌ی یکپارچه یا IDE های مختلفی وجود دارند که به برنامه نویسان در نوشتن و ویرایش کدها، پیدا کردن خطاها، نمایش خروجی، و برخی موارد دیگر کمک می‌کنند. برای اجرای کدهای Python محیط‌های مختلفی وجود دارد که ساده‌ترین آنها، IDEL می‌باشد. برای دانلود این محیط کدنویسی، بر روی لینک زیر کلیک کنید:

<https://www.python.org/downloads>

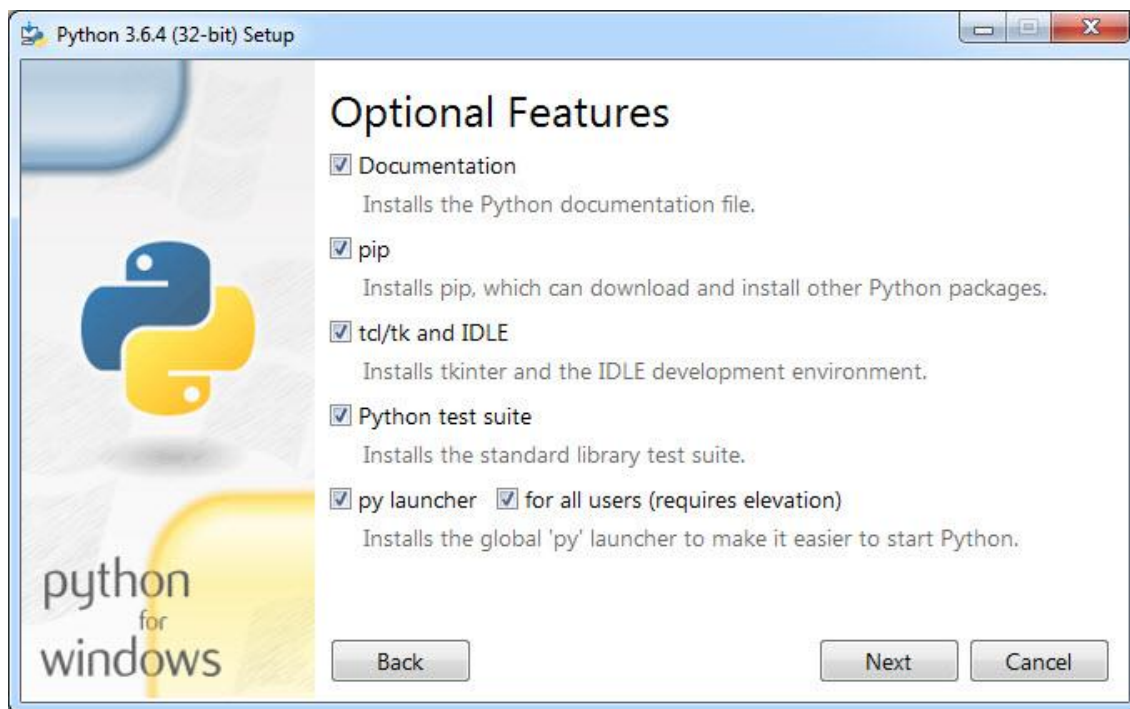
با کلیک بر روی لینک بالا، صفحه‌ای به صورت زیر باز می‌شود، که در این صفحه همانطور که در شکل زیر مشاهده می‌کنید، بر روی دکمه‌ای که با فلش نشان داده شده کلیک کرده، تا آخرین نسخه IDEL دانلود شود:



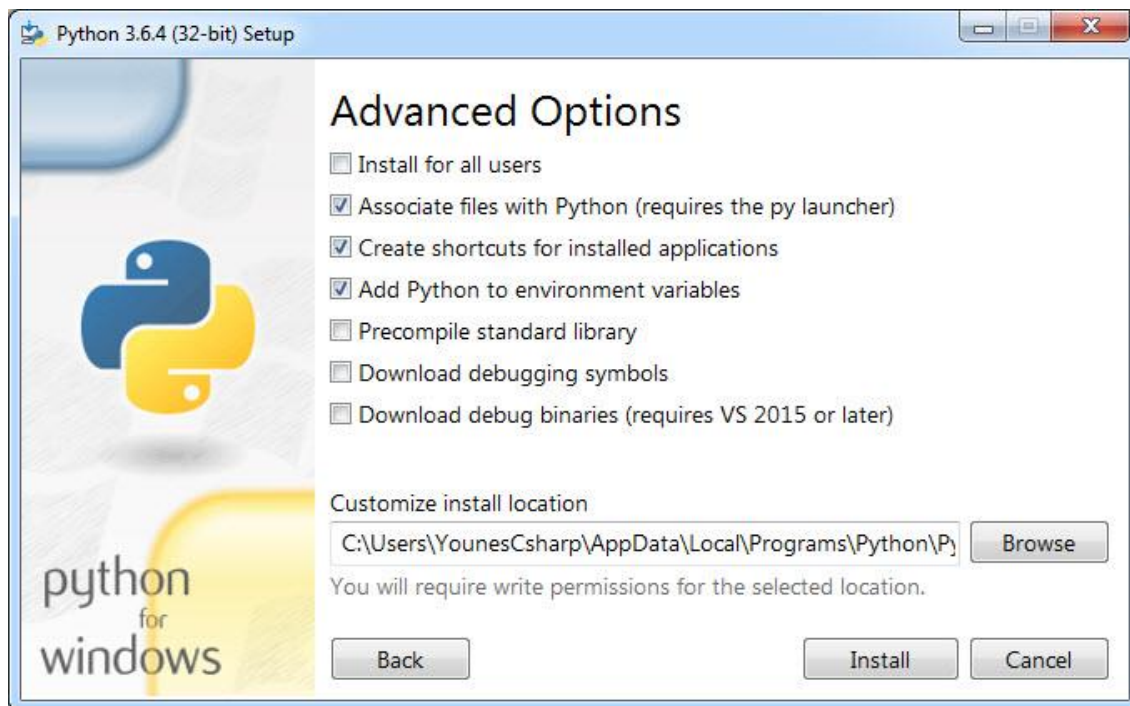
بعد از دانلود فایل مورد نظر به محل ذخیره آن رفته و بر روی فایل دو بار کلیک کنید. سپس در صفحه‌ای که به صورت زیر نمایش داده می‌شود، تیک مورد نظر را زده و سپس بر روی گزینه Customize installation کلیک کنید:



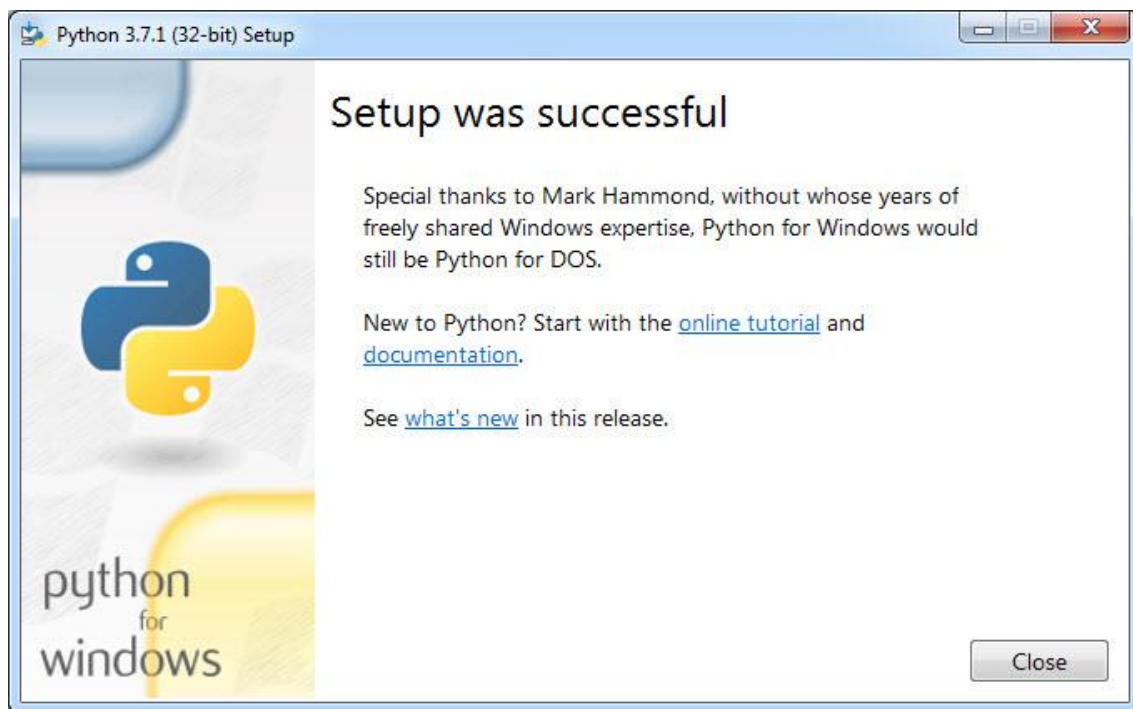
بعد از کلیک بر روی گزینه مذکور صفحه‌ای به صورت زیر نمایش داده می‌شود. در این صفحه بر روی دکمه Next کلیک کنید:



در صفحه بعد بر روی دکمه Install کلیک کرده و منتظر بمانید تا برنامه نصب شود:

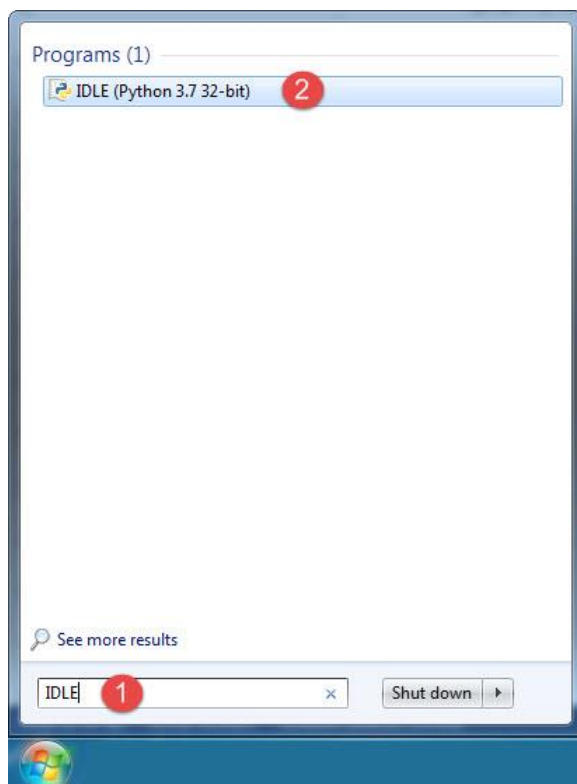


در آخر و بعد از نصب کامل برنامه پیغامی مبنی بر موفقیت آمیز بودن، نصب برنامه به شما نمایش داده می‌شود و شما می‌توانید دکمه Close را بزنید:

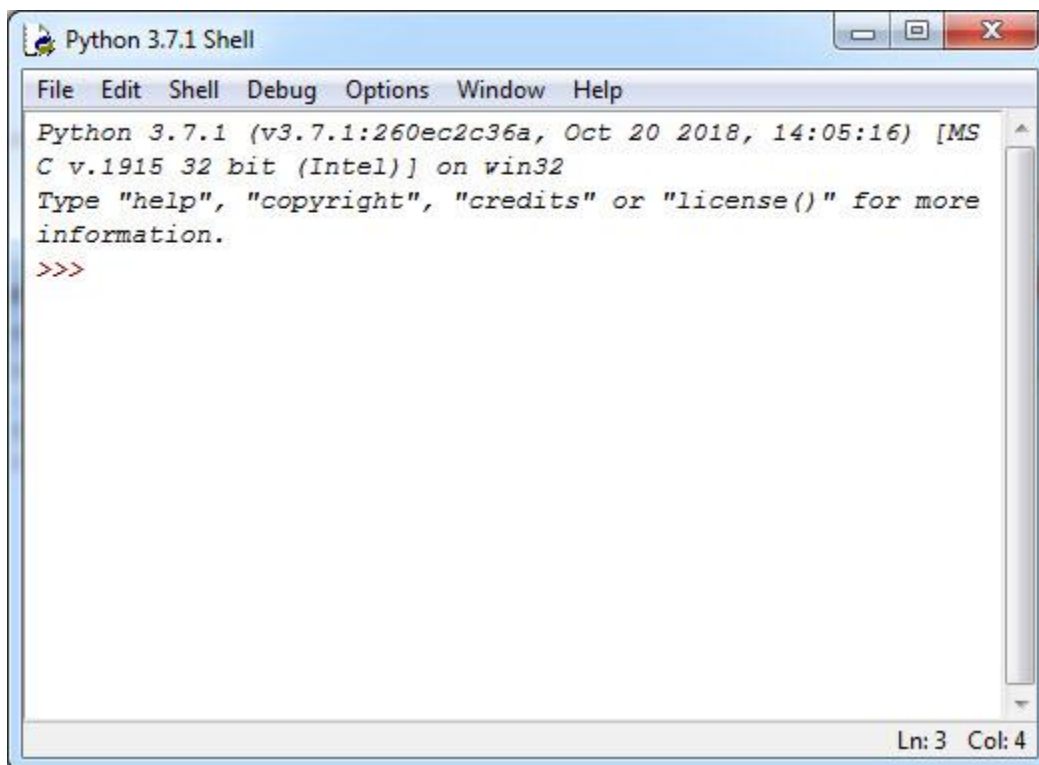


ساخت یک برنامه ساده

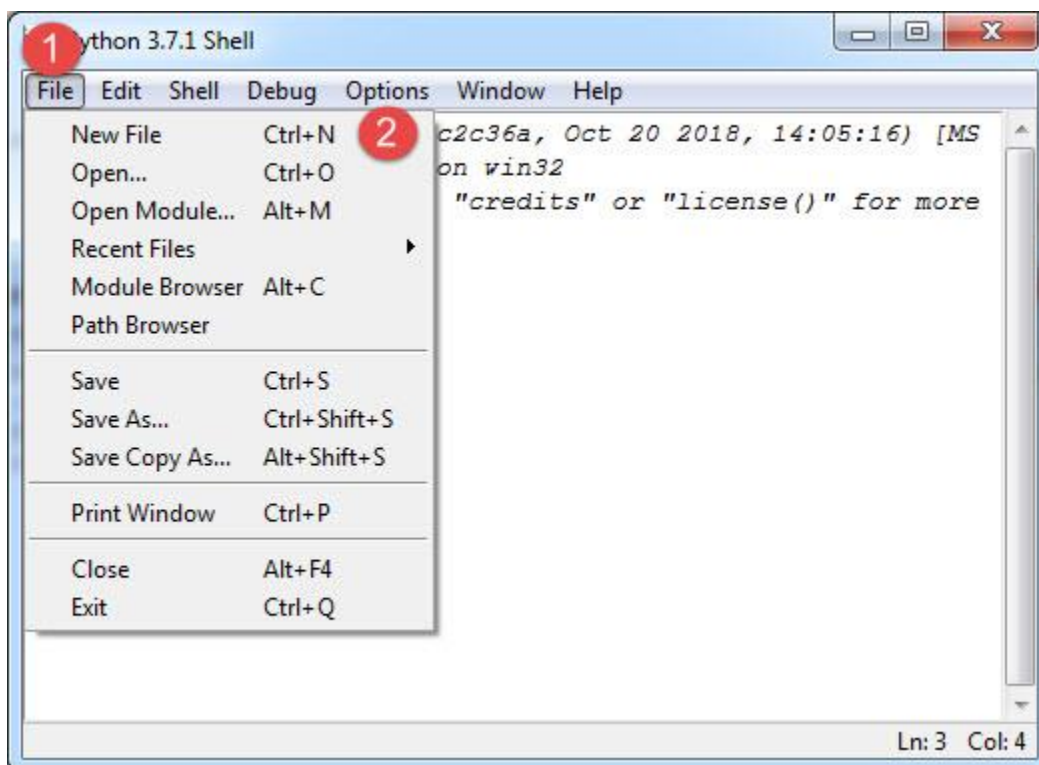
اجازه بدهید یک برنامه بسیار ساده به زبان پایتون بنویسیم. این برنامه یک پیغام را نمایش می‌دهد. از منوی Start محیط برنامه‌نویسی IDLE را به صورت زیر اجرا کنید:



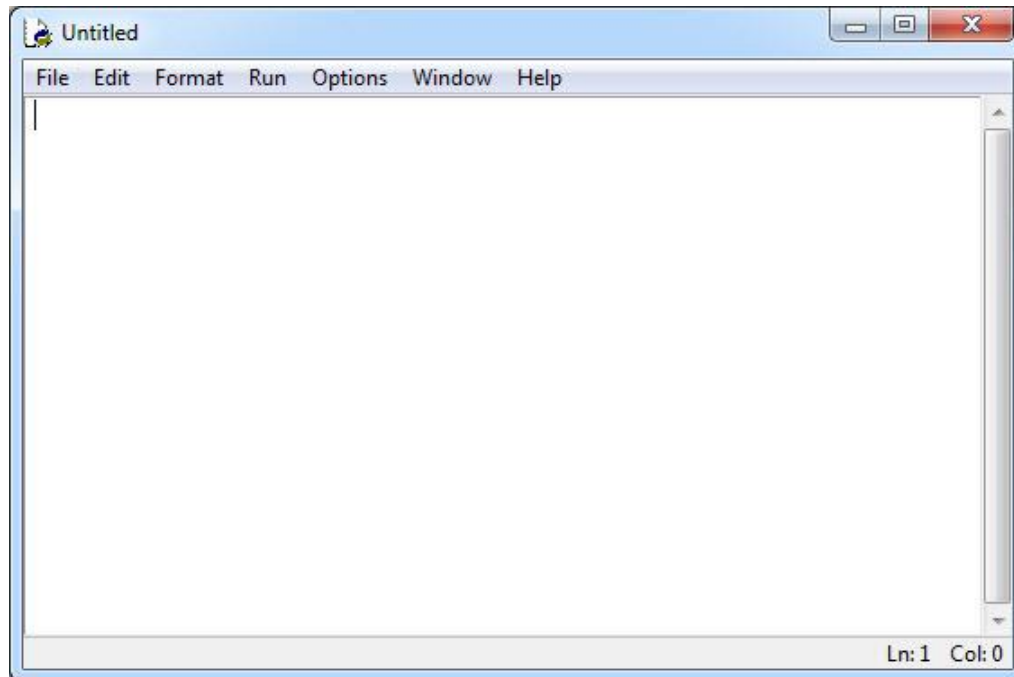
با کلیک بر روی IDEL صفحه‌ای به صورت زیر نمایش داده می‌شود:



در صفحه باز شده به صورت زیر بر روی منوی File و سپس گزینه New File کلیک کنید:

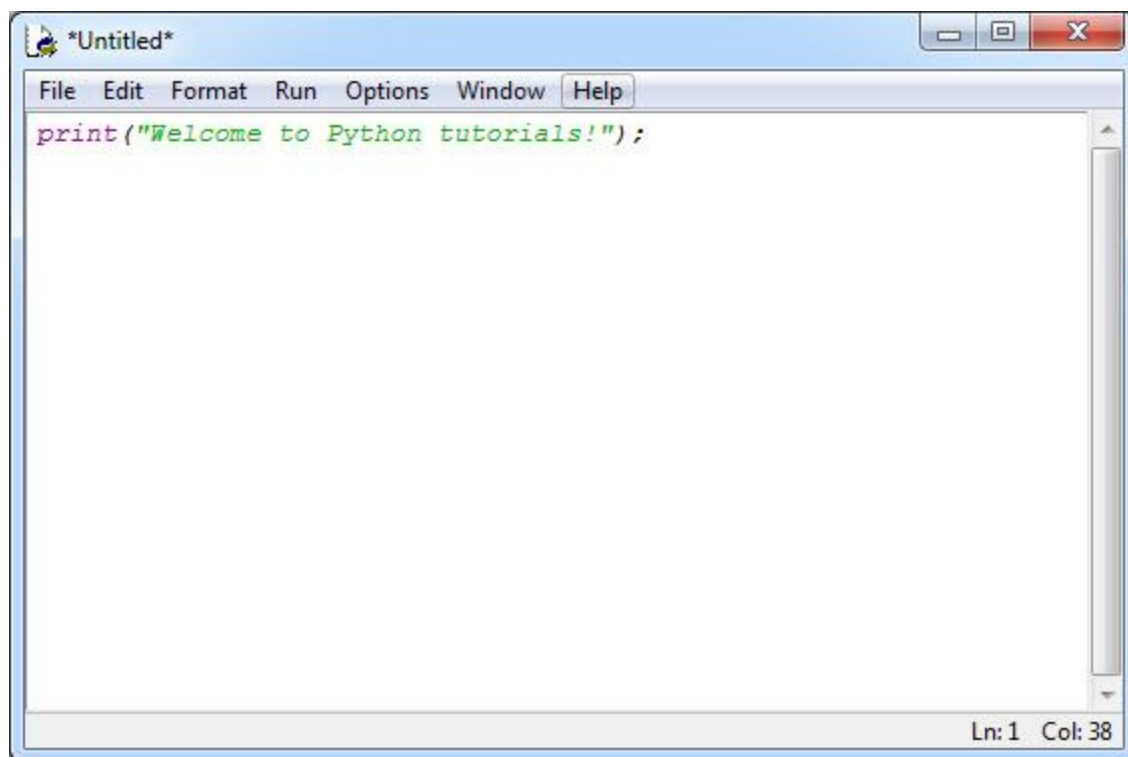


با کلیک بر روی گزینه New File صفحه‌ای به صورت زیر نمایش داده می‌شود که شما می‌توانید کدهای خود را در داخل آن بنویسید:

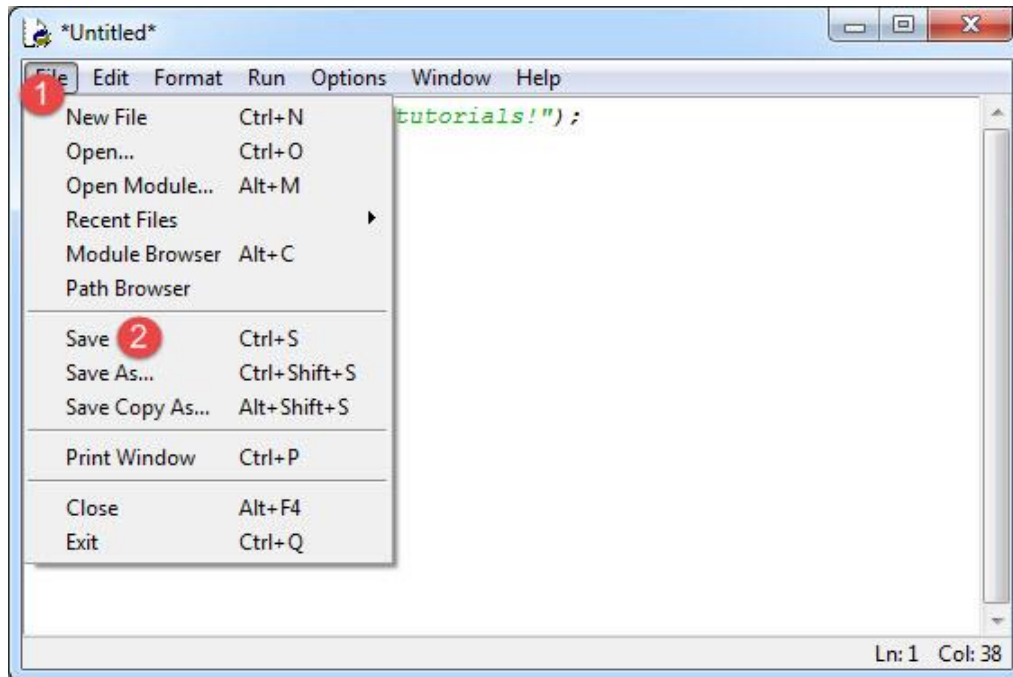


در پنجره بالا کدهای زیر را بنویسید:

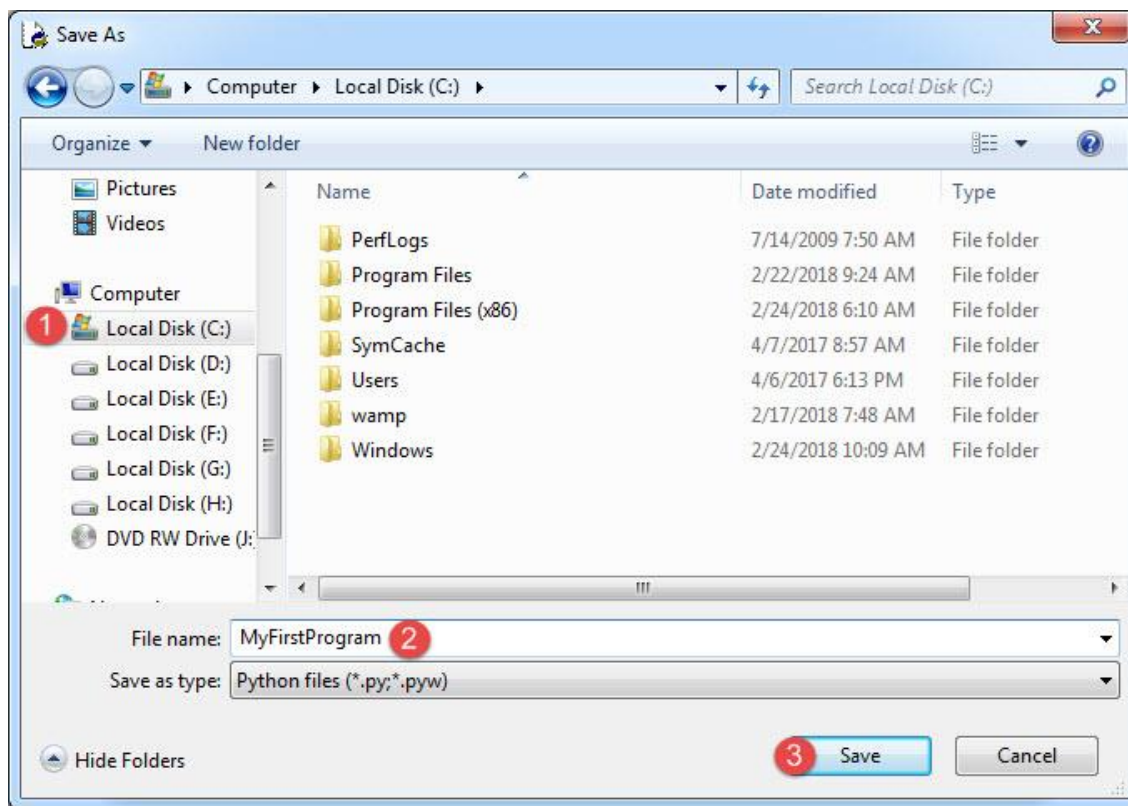
```
print("Welcome to Python Tutorials!")
```



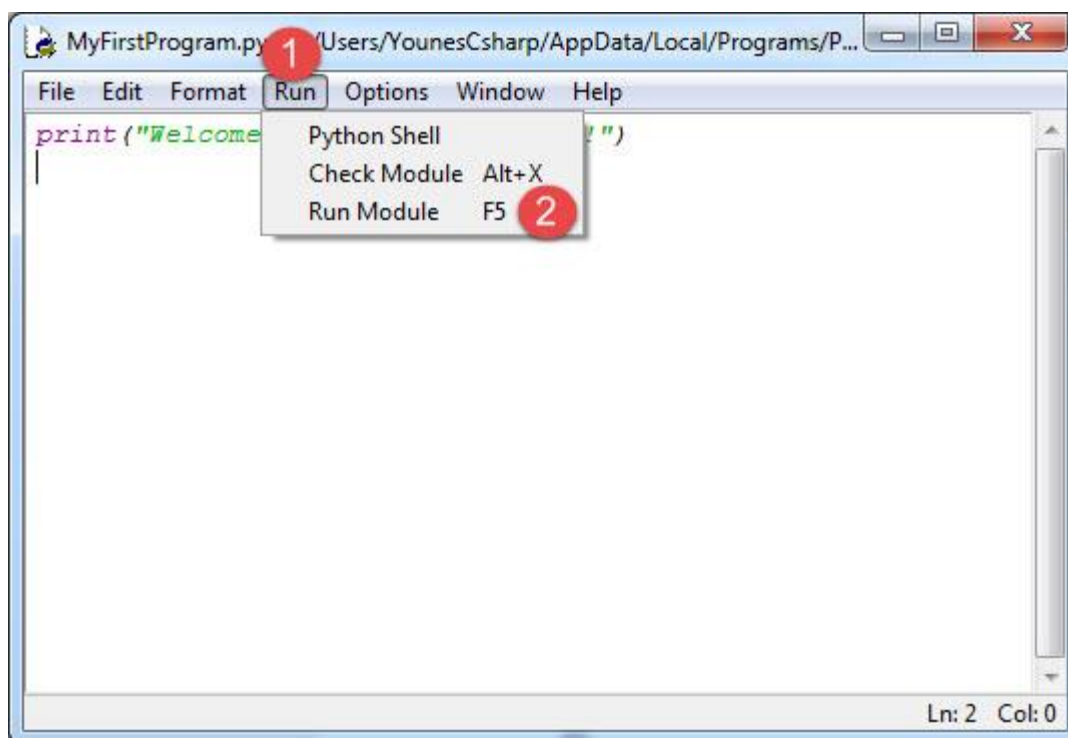
مانند شکل زیر از منوی File گزینه Save را بزنید:



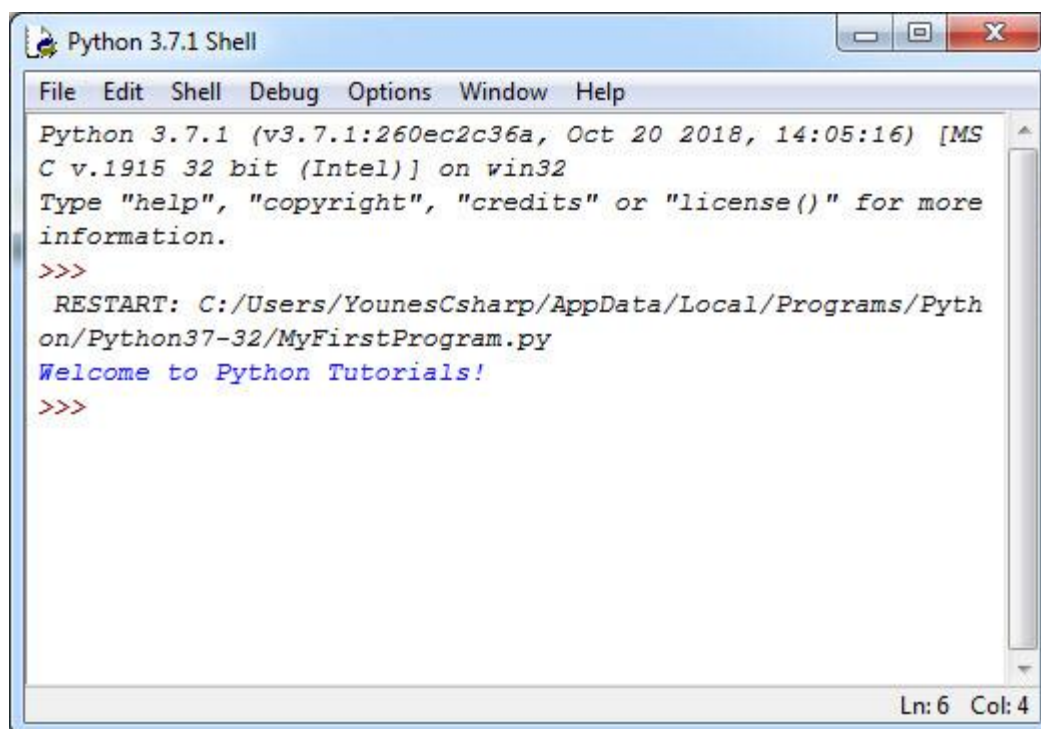
سپس یک مسیر برای ذخیره فایل انتخاب کنید. ما در شکل زیر فایل را در درایو C ذخیره کرده‌ایم:



بعد از ذخیره فایل به محیط کدنویسی برگشته و از منوی Run گزینه Run Module و یا دکمه F5 را بزنید:



مشاهده می‌کنید که برنامه اجرا شده و پیغام Welcome to Python tutorials! چاپ می‌شود:



مثال بالا ساده‌ترین برنامه‌ای است که شما می‌توانید در Python بنویسید. هدف در مثال بالا نمایش یک پیغام در صفحه نمایش است. هر زبان برنامه‌نویسی دارای قواعدی برای کدنویسی است. پایتون دارای توابع از پیش تعریف شده‌ای است که هر کدام برای مقاصد خاصی به کار می‌روند. هر چند که در آینده در مورد توابع بیشتر توضیح می‌دهیم ولی در همین حد به توضیح تابع بسنده می‌کنیم که توابع مجموعه‌ای از کدها هستند که دارای یک نام بوده و در جلوی نام آنها علامت () قرار می‌گیرد. یکی از این توابع، تابع `print()` است. از تابع `print()` برای چاپ یک رشته استفاده می‌شود. یک رشته گروهی از کاراکترها است، که به وسیله دابل کوتیشن (") محصور شده است. مانند: "Welcome to Python Tutorials!". یک کاراکتر می‌تواند یک حرف، عدد، علامت یا ... باشد. در کل مثال بالا نحوه استفاده از تابع `print()` است. توضیحات بیشتر در درس‌های آینده آمده است. پایتون فضای خالی بالا را نادیده می‌گیرد و از کد زیر اشکال نمی‌گیرد:

```
print(
    "Welcome to Python Tutorials!")
```

همیشه به یاد داشته باشید که Python به بزرگی و کوچکی حروف حساس است. یعنی به طور مثال MAN و man در Python با هم فرق دارند. رشته‌ها و توضیحات از این قاعده مستثنی هستند که در درس‌های آینده توضیح خواهیم داد. مثلاً کدهای زیر با خطا مواجه می‌شوند و اجرا نمی‌شوند:

```
Print("Welcome to Python Tutorials!")
PRINT("Welcome to Python Tutorials!")
Print("Welcome to Python Tutorials!")
```

تغییر در بزرگی و کوچکی حروف از اجرای کدها جلوگیری می‌کند. اما کد زیر کاملاً بدون خطا است:

```
print("Welcome to Python tutorials!")
```

نکاتی در مورد کدنویسی در پایتون

در زبان‌هایی مثل جاوا و سی شارپ، از علامت آکولاد ({}) برای ایجاد یک بلاک کد

```
Block
{
    statement
}
```

ولی در زبان پایتون از ترکیب علامت دو نقطه (:) و تو رفتگی برای اینکار استفاده می‌شود:

```
Block:
    statement
```

ما با دونقطه به پایتون می‌گوییم که قصد داریم یک بلوک کد را آغاز کنیم و با تو رفتگی ابتدای خطوط دستورات آن بلوک را تعریف می‌کنیم. برای تورفتگی می‌توانیم از ۳ یا ۴ یا ۱۰ فضای خالی استفاده کنیم. میزان این فضای خالی تا زمانی که در تمام کد رعایت شود، اهمیتی ندارد. در کد زیر به اهمیت تو رفتگی‌ها پی می‌برید:


```

1 Block1:
2     statement
3     statement
4 Block2:
5     statement
6     Block3:
7         statement
8         statement
9         Block4:
10             statement
11     statement
12 statement

```

در کد بالا، بلاک اول (Block1) از خط ۱ تا ۳ را شامل می‌شود. به این نکته توجه کنید که خطوط بعد از علامت: حتماً باید دارای تو رفتگی باشند. بلاک دوم (Block2) از خط ۴ شروع و به خط ۱۱ ختم می‌شود. نکته‌ای که باید در اینجا دوباره به آن اشاره کنیم این است که دستور یا بلاک‌هایی که دارای فاصله‌های برابر از سمت چپ هست جزو یک بلاک می‌باشند. مثلاً در کد بالا خطوط ۲ و ۳ جز Block1 هستند چون تو رفتگی آنها از سمت چپ برابر است و اگر مثلاً فاصله‌های خط ۳ از سمت چپ را حذف کنیم دیگر جز بلاک محسوب نمی‌شود. یک بلاک را می‌توان زیر مجموعه بلاک دیگر کرد. مثلاً در خط ۶، Block3 را زیر مجموعه Block2 و در خط ۹، Block4 را زیر مجموعه Block3 کرده‌ایم. در نهایت خط ۱۲ جز هیچکدام از بلاک‌ها نیست و مستقل اجرا می‌شود. گاهی اوقات و هنگام کدنویسی، لازم است که رشته‌های طولانی را در چند خط بنویسید. برای اینکار در پایتون می‌توان از علامت \ به صورت زیر استفاده کنید:

```

print("Welcome \
to \
Python \
Tutorials!")

```

توضیحات

وقتی که کدی تایپ می‌کنید شاید بخواهید که متنی جهت یادآوری وظیفه آن کد به آن اضافه کنید. در Python (و بیشتر زبانهای برنامه‌نویسی) می‌توان این کار را با استفاده از توضیحات انجام داد. توضیحات متونی هستند که توسط مفسر نادیده گرفته می‌شوند و به عنوان بخشی از کد محسوب نمی‌شوند.

هدف اصلی از ایجاد توضیحات، بالا بردن خوانایی و تشخیص نقش کدهای نوشته شده توسط شما، برای دیگران است. فرض کنید که می‌خواهید در مورد یک کد خاص، توضیح بدهید، می‌توانید توضیحات را در بالای کد یا کنار آن بنویسید. از توضیحات برای مستند سازی برنامه هم استفاده می‌شود. در برنامه زیر نقش توضیحات نشان داده شده است:

```

#This line will print the message hello world
print("Hello World!")

```

در کد بالا، خط اول کد بالا یک توضیح درباره خط دوم است که به کاربر اعلام می‌کند که وظیفه خط دوم چیست؟ با اجرای کد بالا فقط جمله Hello World چاپ شده و خط اول در خروجی نمایش داده نمی‌شود چون مفسر توضیحات را نادیده می‌گیرد. همانطور که مشاهده

می‌کنید برای درج توضیحات در پایتون از علامت # استفاده می‌شود. برای توضیحات طولانی هم باید در ابتدای هر خط از توضیح این علامت درج شود:

```
#This line will print
#the message hello world
print("Hello World!")
```

کاراکترهای کنترلی

کاراکترهای کنترلی، کاراکترهای ترکیبی هستند که با یک بک اسلش (\) شروع می‌شوند و به دنبال آنها یک حرف یا عدد می‌آید و یک رشته را با فرمت خاص نمایش می‌دهند. برای مثال برای ایجاد یک خط جدید و قرار دادن رشته در آن می‌توان از کاراکتر کنترلی \n استفاده کرد:

```
print("Hello\nWorld!")
```

```
Hello
World
```

مشاهده کردید که مفسر بعد از مواجهه با کاراکتر کنترلی \n نشانگر ماوس را به خط بعد برده و بقیه رشته را در خط بعد نمایش می‌دهد. جدول زیر لیست کاراکترهای کنترلی و کاربرد آنها را نشان می‌دهد:

عملکرد	کاراکتر کنترلی	عملکرد	کاراکتر کنترلی
Form Feed	\f	چاپ کوتیشن	\'
خط جدید	\n	چاپ دابل کوتیشن	\"
سر سطر رفتن	\r	چاپ بک اسلش	\\
حرکت به صورت افقی	\t	چاپ فضای خالی	\0
حرکت به صورت عمودی	\v	صدای بیپ	\a
چاپ کاراکتر یونیکد	\u	حرکت به عقب	\b

ما برای استفاده از کاراکترهای کنترلی، از بک اسلش (\) استفاده می‌کنیم. از آنجاییکه \ معنای خاصی به رشته‌ها می‌دهد برای چاپ بک اسلش (\) باید از (\\) استفاده کنیم:

```
print("We can print a \\ by using the \\ escape sequence.")
```

We can `print` a `\` by using the `\\` escape sequence.

یکی از موارد استفاده از `\\`، نشان دادن مسیر یک فایل در ویندوز است:

```
print("C:\\Program Files\\Some Directory\\SomeFile.txt")
```

```
C:\Program Files\Some Directory\SomeFile.txt
```

از آنجاییکه از دابل کوتیشن (") برای نشان دادن رشته‌ها استفاده می‌کنیم برای چاپ آن از \" استفاده می‌کنیم:

```
print("I said, \"Motivate yourself!\".")
```

```
I said, "Motivate yourself!".
```

همچنین برای چاپ کوتیشن (') از \' استفاده می‌کنیم:

```
print("The programmer\'s heaven.")
```

```
The programmer's heaven.
```

برای ایجاد فاصله بین حروف یا کلمات از `\t` استفاده می‌شود:

```
print("Left\tRight")
```

```
Left   Right
```

برای چاپ کاراکترهای یونیکد می‌توان از `\u` استفاده کرد. برای استفاده از `\u`، مقدار در مبانی ۱۶ کاراکتر را درست بعد از علامت `\u` قرار می‌دهیم. برای مثال اگر بخواهیم علامت کپی رایت (©) را چاپ کنیم، باید بعد از علامت `\u` مقدار `00A9` را قرار دهیم مانند:

```
print("\u00A9")
```

```
©
```

برای مشاهده لیست مقادیر مبانی ۱۶ برای کاراکترهای یونیکد به لینک زیر مراجعه نمایید:

<http://www.ascii.cl/htmlcodes.htm>

اگر مفسر به یک کاراکتر کنترلی غیر مجاز برخورد کند، برنامه پیغام خطا می‌دهد. بیشترین خطا زمانی اتفاق می‌افتد که برنامه نویس برای چاپ اسلش (`\`) از `\\` استفاده می‌کند.

متغیر

متغیر مکانی از حافظه است که شما می‌توانید مقادیری را در آن ذخیره کنید. می‌توان آن را به عنوان یک ظرف تصور کرد که داده‌های خود را در آن قرار داده‌اید. محتویات این ظرف می‌تواند پاک شود یا تغییر کند. هر متغیر دارای یک نام نیز هست. که از طریق آن می‌توان متغیر را از دیگر متغیرها تشخیص داد و به مقدار آن دسترسی پیدا کرد. همچنین دارای یک مقدار می‌باشد که می‌تواند توسط کاربر انتخاب شده باشد یا نتیجه یک محاسبه باشد. مقدار متغیر می‌تواند تهی نیز باشد. متغیر دارای نوع نیز هست بدین معنی که نوع آن با نوع داده‌ای که در آن ذخیره می‌شود یکی است.

متغیر دارای عمر نیز هست که از روی آن می‌توان تشخیص داد که متغیر باید چقدر در طول برنامه مورد استفاده قرار گیرد. و در نهایت متغیر دارای محدوده استفاده نیز هست که به شما می‌گوید که متغیر در چه جای برنامه برای شما قابل دسترسی است. ما از متغیرها به عنوان یک انبار موقتی برای ذخیره داده استفاده می‌کنیم. هنگامی که یک برنامه ایجاد می‌کنیم احتیاج به یک مکان برای ذخیره داده، مقادیر یا داده‌هایی که توسط کاربر وارد می‌شوند، داریم. این مکان، همان متغیر است.

برای این از کلمه متغیر استفاده می‌شود چون ما می‌توانیم بسته به نوع شرایط هر جا که لازم باشد، مقدار آن را تغییر دهیم. متغیرها موقتی هستند و فقط موقعی مورد استفاده قرار می‌گیرند که برنامه در حال اجراست و وقتی شما برنامه را می‌بندید محتویات متغیرها نیز پاک می‌شود. قبلاً ذکر شد که به وسیله نام متغیر می‌توان به آن دسترسی پیدا کرد. برای نامگذاری متغیرها باید قوانین زیر را رعایت کرد:

- نام متغیر باید با یکی از حروف الفبا (a-z or A-Z) یا علامت _ شروع شود.
- نمی‌تواند شامل کاراکترهای غیرمجاز مانند \$, ^, ?, #, باشد.
- نمی‌توان از کلمات رزرو شده در پایتون برای نام متغیر استفاده کرد.
- نام متغیر نباید دارای فضای خالی (spaces) باشد.

اسامی متغیرها نسبت به بزرگی و کوچکی حروف حساس هستند. در پایتون دو حرف مانند a و A دو کاراکتر مختلف به حساب می‌آیند.

دو متغیر با نامهای myNumber و MyNumber دو متغیر مختلف محسوب می‌شوند چون یکی از آنها با حرف کوچک m و دیگری با حرف بزرگ M شروع می‌شود. متغیر دارای نوع هست که نوع داده‌ای را که در خود ذخیره می‌کند را نشان می‌دهد. در درس بعد در مورد انواع داده‌ها در پایتون توضیح می‌دهیم. لیست کلمات کلیدی پایتون، که نباید از آنها در نامگذاری متغیرها استفاده کرد در زیر آمده است:

False	def	if	raise
None	del	import	return
True	elif	in	try
and	else	is	while
as	except	lambda	with
assert	finally	nonlocal	yield
break	for	not	
class	from	or	
continue	global	pass	