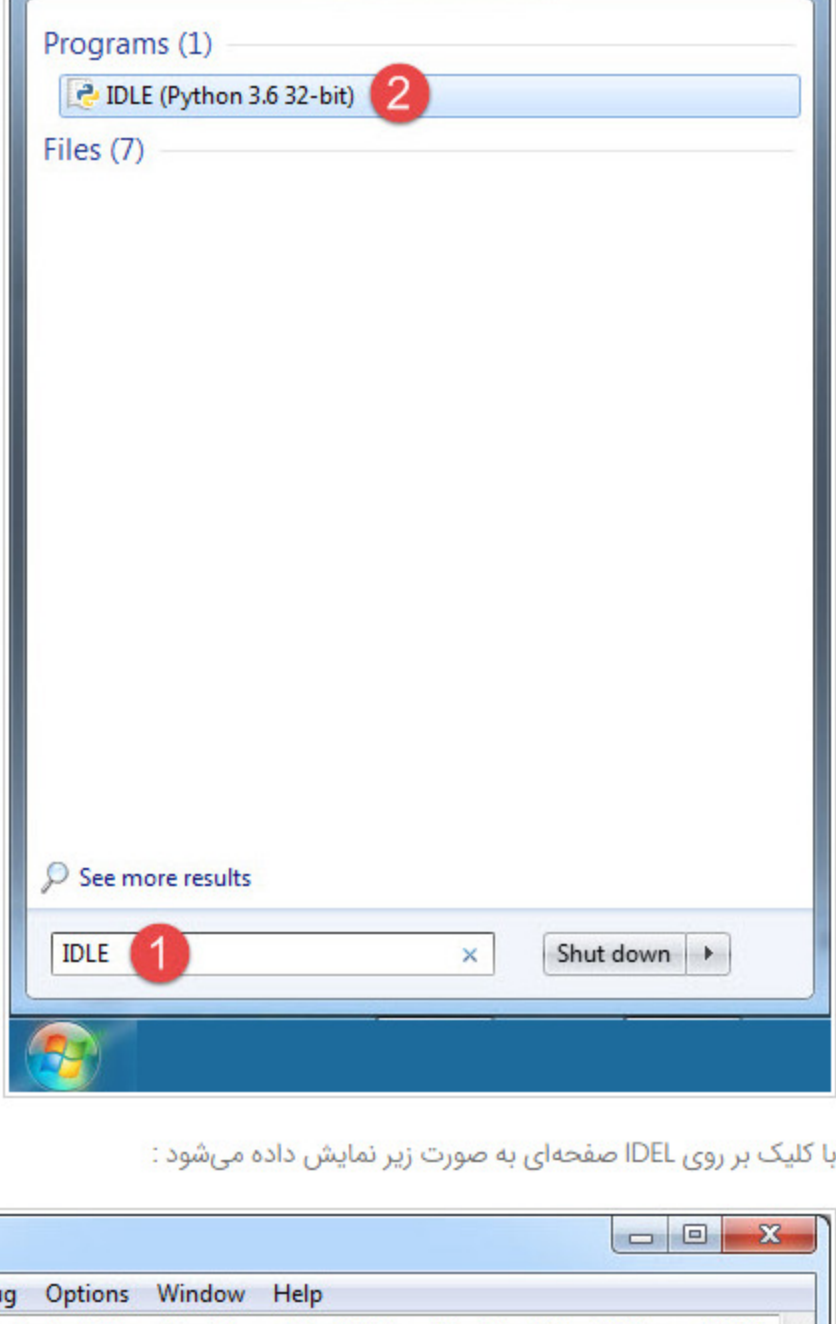
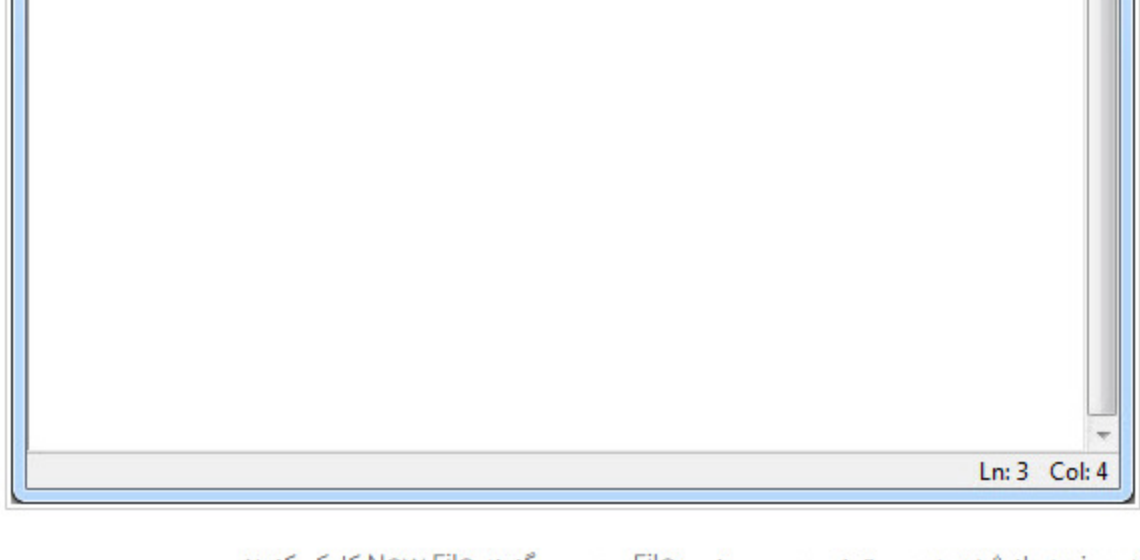


ساخت یک برنامه ساده

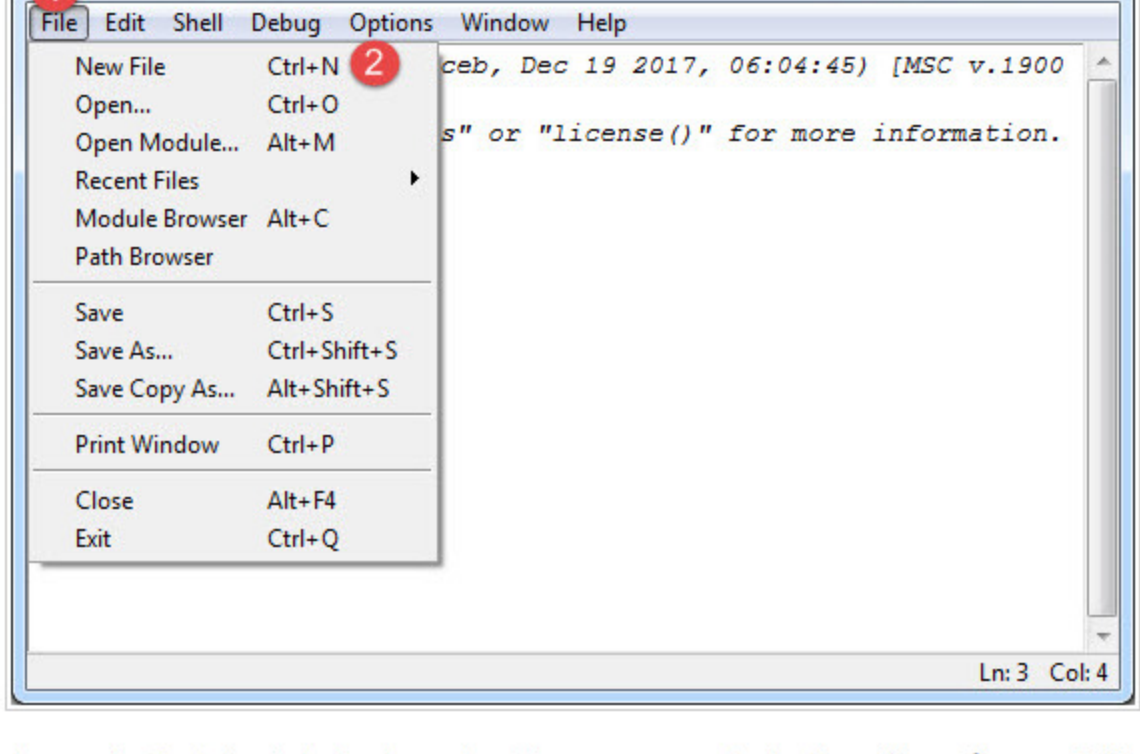
اجازه بدهید یک برنامه بسیار ساده به زبان پایتون بنویسیم. این برنامه یک پیغام را نمایش می‌دهد. از منوی Start محیط برنامه نویسی IDEL را به صورت زیر اجرا کنید:



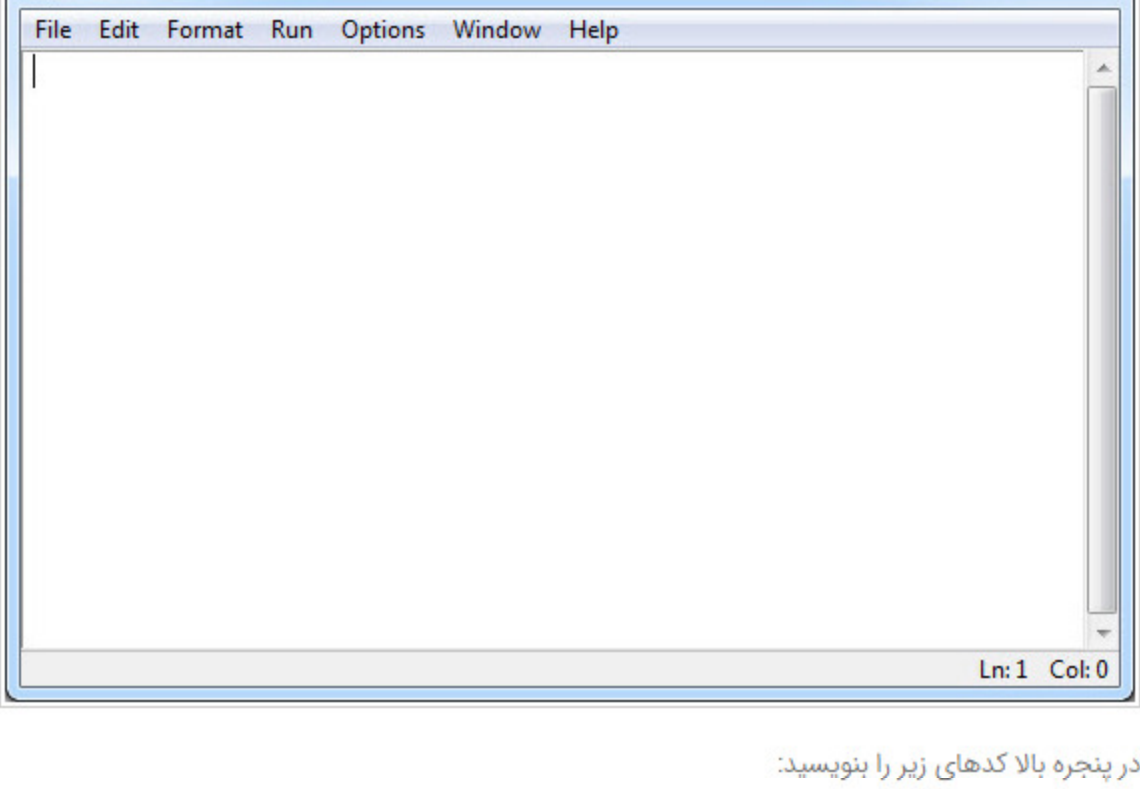
با کلیک بر روی IDEL صفحه‌ای به صورت زیر نمایش داده می‌شود :



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

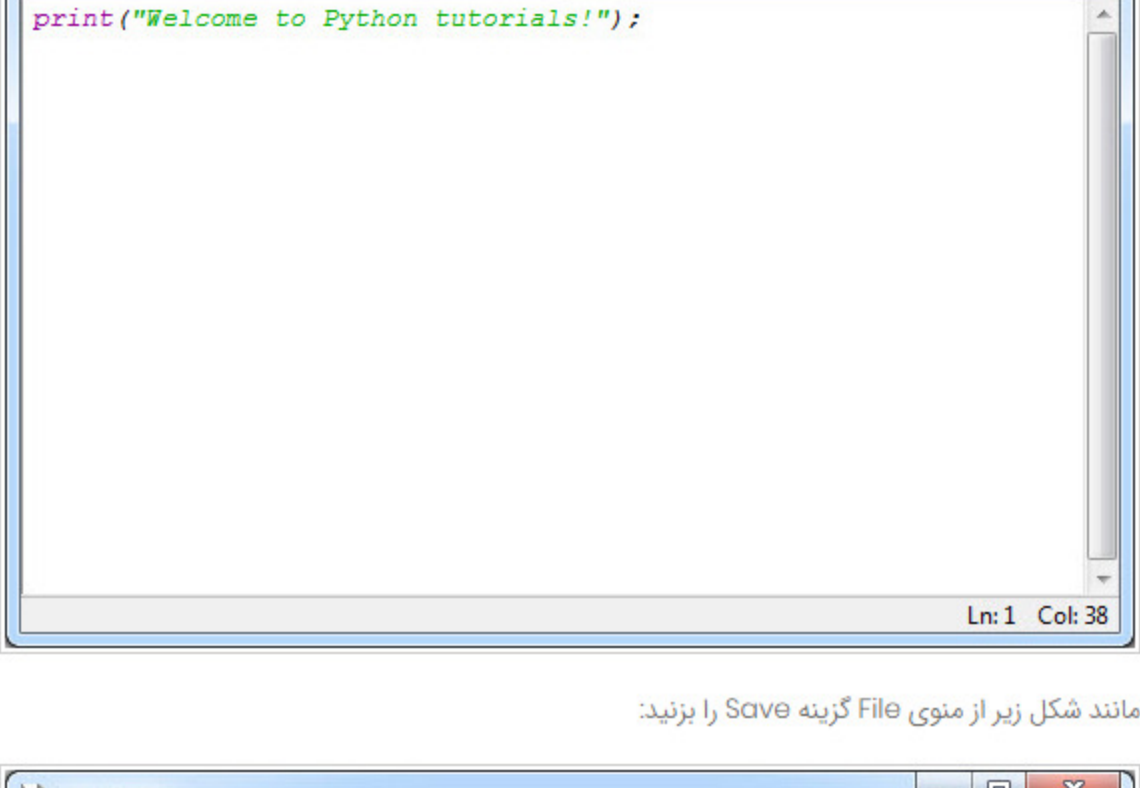


با کلیک بر روی گزینه New File صفحه‌ای به صورت زیر نمایش داده می‌شود که شما می‌توانید کدهای خود را در داخل آن بنویسید:

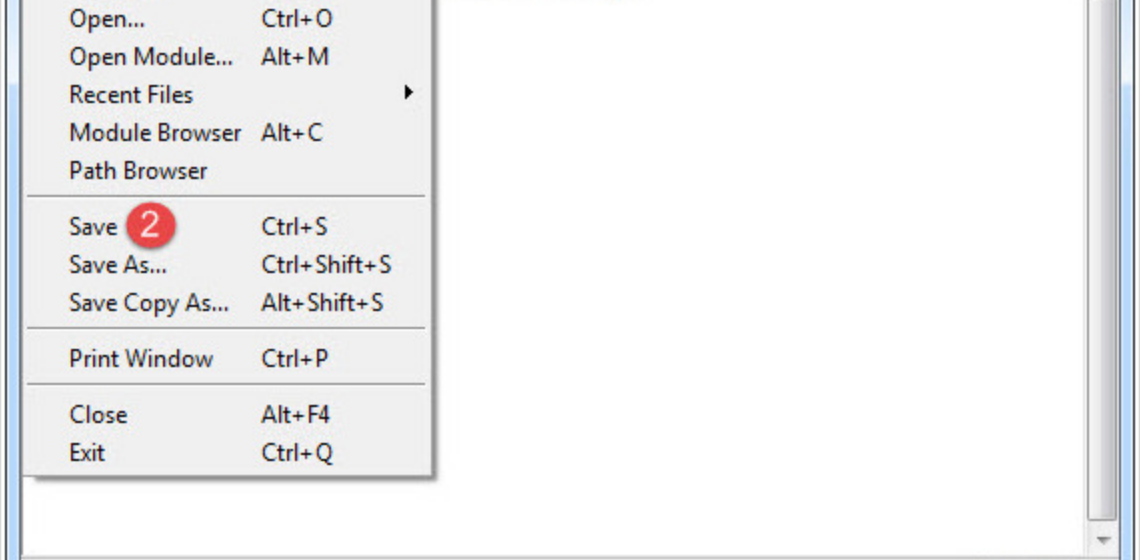


در پنجره بالا کدهای زیر را بنویسید:

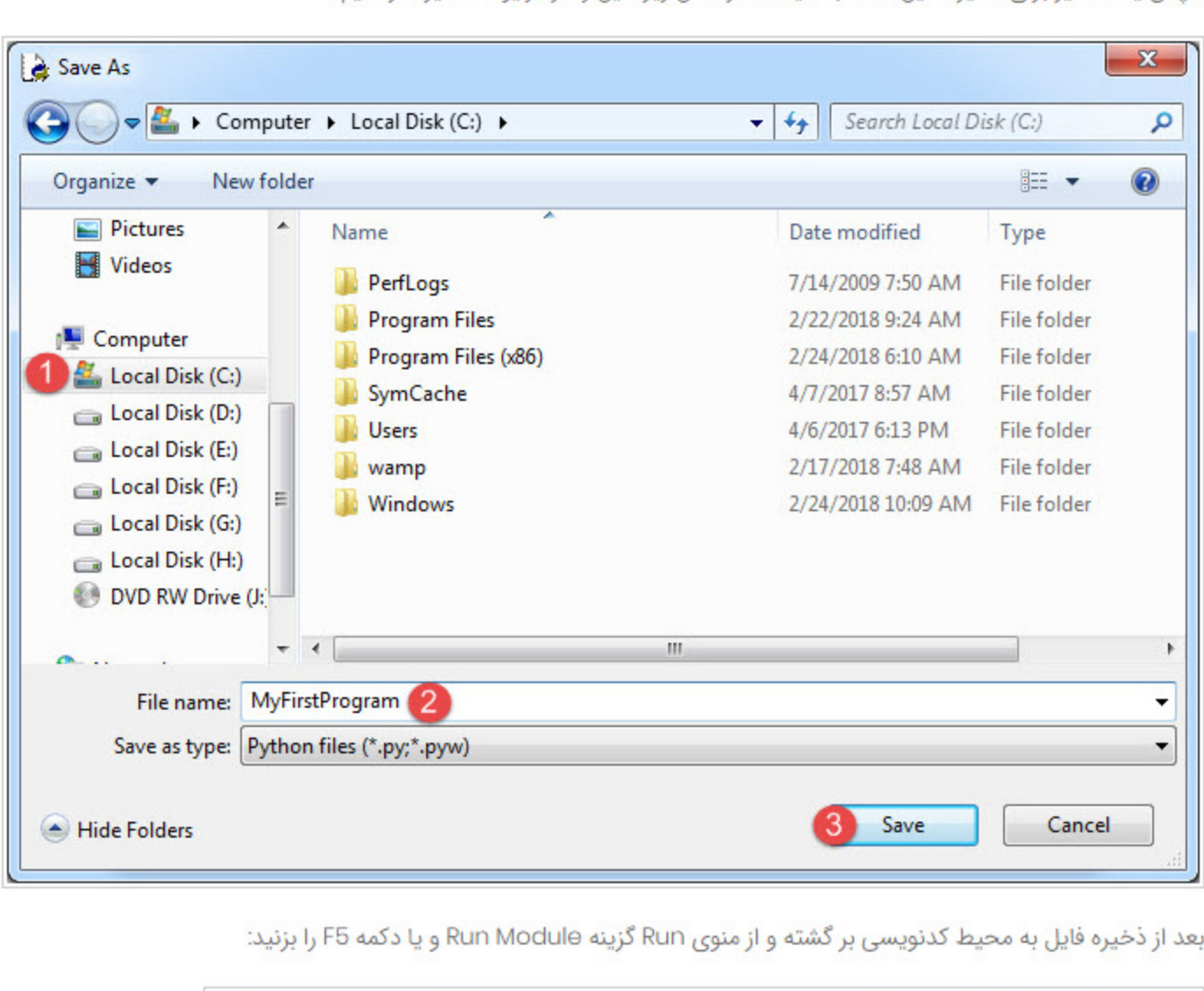
```
print("Welcome to Python Tutorials!")
```



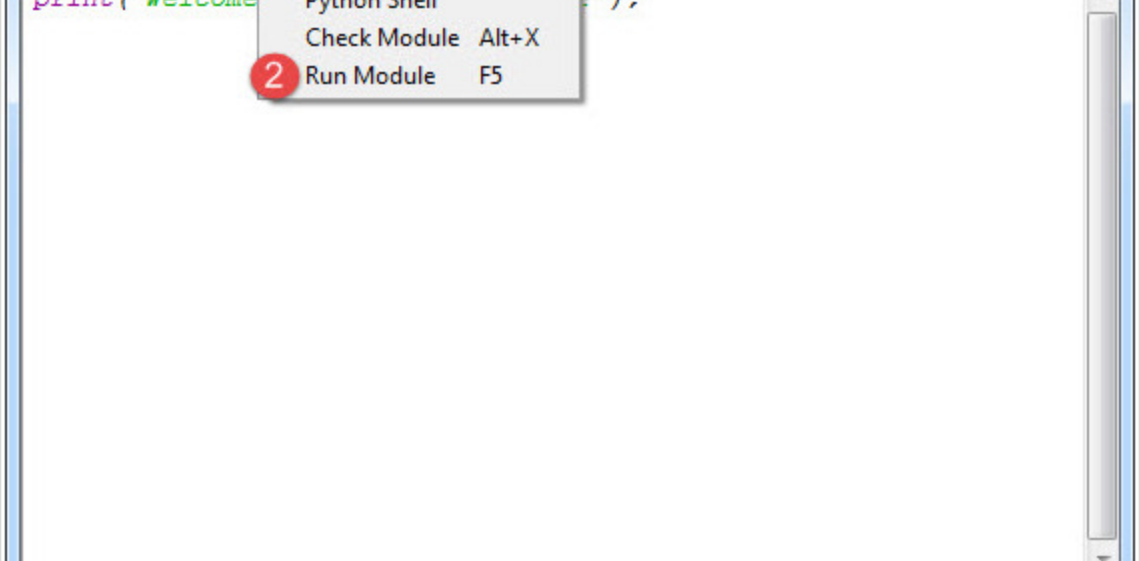
مانند شکل زیر از منوی File گزینه Save را بزنید:



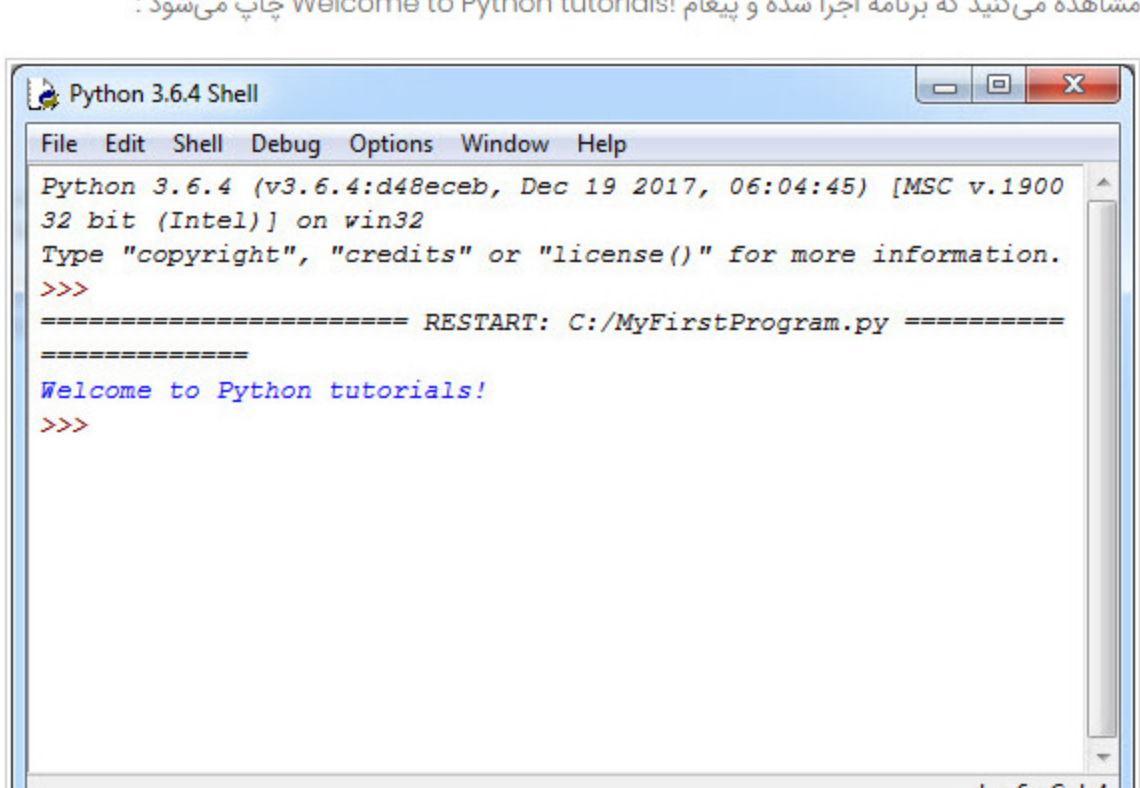
سپس یک مسیر برای ذخیره فایل انتخاب کنید. ما در شکل زیر فایل را در درایو C ذخیره کرده‌ایم :



بعد از ذخیره فایل به محیط کدنویسی بر گشته 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 برنامه نویسی 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 بین علامت‌های ()، {} و [] را نادیده می‌گیرد. مثلاً 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!")
PrintT("Welcome to Python Tutorials!")
```

تغییر به 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 دو نقطه به پایتون می‌گوییم که قصد داریم یک بلاک کد را آغاز کنیم و با تو رفتگی ابتدای خطوط دستورات آن بلاک را تعریف می‌کنیم. برای توفرتگی می‌توانیم از 3 یا 4 یا 10 فضای خالی استفاده کنیم. میزان این فضای خالی تا زمانی که در تمام کد رعایت شود، اهمیتی ندارد. در حالت استاندارد از 4 بار

زدن دکمه Space و یا یک بار زدن دکمه Tab برای تو رفتگی‌ها استفاده می‌شود. 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 1 تا 3 را شامل می‌شود. به این نکته توجه کنید که خطوط بعد از علامت: حتماً باید دارای تو رفتگی باشند. بلاک دوم (Block2) از خط 4 شروع و به خط 11 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 2 و 3 جز Block1 هستند و بدنه آن محسوب می‌شوند. چون تو رفتگی آنها از سمت چپ برابر است و اگر مثلاً فاصله‌های خط 3 از سمت چپ را حذف کنیم بگر جز بلاک محسوب نمی‌شود. یک بلاک را می‌توان زیر مجموعه بلاک دیگر کرد. مثلاً در خط 6، Block3 را زیر مجموعه Block2 و در خط 9، Block4 را زیر مجموعه Block3 کرده‌ایم. در نهایت خط 12 جز هیچکدام از بلاک‌ها نیست

مستقل اجرا می‌شود.