



همانند هر زبان برنامه‌نویسی دیگری، در شروع یادگیری پایتون باید کمی وقت صرف کنید تا به خوبی بتوانید با ترکیب نحوی این زبان ارتباط برقرار کرده و به شکل روانی کدنویسی کنید. برای نیل به این هدف، باید بتوانید در کنار مباحث آموزشی از راهنمای آفلاین پایتون نیز استفاده کنید.

برای مطالعه بخش سوم آموزش رایگان پایتون [اینجا](#) کلیک کنید

چهار دستور کاربردی که اطلاعاتی در ارتباط با پایتون در اختیارتان قرار می‌دهند

در کادر جست‌وجوی ویندوز 10 عبارت python را نوشته و کلید اینتر را فشار دهید تا محیط تعاملی خط فرمان پایتون اجرا شود.

(Copyright): این فرمان اطلاعاتی در ارتباط با لایسنس پایتون، سازمانی که مسئولیت پشتیبانی از پایتون را عهده‌دار است و همچنین تاریخ نگارش پایتونی که روی سیستم شما ثبت شده است ارائه می‌کند.

```
Python 3.7 (64-bit)
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 6
Type "help", "copyright", "credits" or "license" for more information.
>>> copyright()
Copyright (c) 2001-2018 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
>>> _
```

Credits(): اطلاعاتی در ارتباط با سازمان‌هایی که از پایتون پشتیبانی به عمل می‌آورند ارائه می‌کند.

License(): این فرمان اطلاعاتی در ارتباط با مفاد موافقت‌نامه‌ای را ارائه می‌کند که شما با قبول آن‌ها اجازه دارید به شکل حرفه‌ای از پایتون استفاده کنید.

Help(): این فرمان جزء آن گروه از فرمان‌هایی است که زیاد از آن استفاده خواهید کرد. با تایپ این فرمان در محیط خط فرمان پایتون و فشار کلید اینتر به محیط راهنمای پایتون وارد می‌شوید. (دقت کنید که پرانتزها را پس از پایان فرمان قرار دهید.)

Python 3.7 (64-bit)

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64-bit]
Type "help", "copyright", "credits" or "license" for more information.
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
'modules', 'keywords', 'symbols', or 'topics'.  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> _
```

زمانی که در محیط راهنما قرار دارید با تایپ دستورات کلیدی قادر هستید اطلاعات مفیدی به دست آورید. به طور مثال، اگر دوست دارید اطلاعات کلیدی در ارتباط با پایتون دریافت کنید با تایپ هر یک از کلیدواژه‌های `modules`، `keywords` و `topics` اطلاعاتی در ارتباط کلیدواژه‌ها، مازول‌ها و نقطه شروع به کار در **پایتون** به دست می‌آورید. به طور مثال با تایپ واژه `topics` در این محیط فهرستی همانند شکل زیر نشان داده می‌شود.

```

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> topics

Here is a list of available topics. Enter any topic name to get more help.

ASSERTION          DELETION           LOOPING            SHIFTING
ASSIGNMENT         DICTIONARIES      MAPPINGMETHODS    SLICINGS
ATTRIBUTEMETHODS  DICTIONARYLITERALS  MAPPINGS          SPECIALATTRIBUT
ATTRIBUTES         DYNAMICFEATURES   METHODS           SPECIALIDENTIFI
AUGMENTEDASSIGNMENT ELLIPSIS          MODULES           SPECIALMETHODS
BASICMETHODS       EXCEPTIONS        NAMESPACES        STRINGMETHODS
BINARY            EXECUTION         NONE              STRINGS
BITWISE           EXPRESSIONS       NUMBERMETHODS     SUBSCRIPTS
BOOLEAN          FLOAT             NUMBERS           TRACEBACKS
CALLABLEMETHODS   FORMATTING        OBJECTS           TRUTHVALUE
CALLS            FRAMEOBJECTS     OPERATORS         TUPELITERALS
CLASSES          FRAMES           PACKAGES          TUPLES
CODEOBJECTS      FUNCTIONS        POWER             TYPEOBJECTS
COMPARISON       IDENTIFIERS      PRECEDENCE        TYPES
COMPLEX          IMPORTING        PRIVATENAMES      UNARY
CONDITIONAL     INTEGER          RETURNING         UNICODE
CONTEXTMANAGERS LISTLITERALS     SCOPING           SEQUENCEMETHODS
CONVERSIONS     LISTS            SEQUENCES
DEBUGGING       LITERALS
help>

```

زمانی که عنوانی را مشاهده می‌کنید که در نظر دارید اطلاعاتی در ارتباط با آن به دست آورید، به طور مثال FUNCTIONS را تایپ کرده و کلید اینتر را فشار دهید. راهنمای پایتون به شما اعلام می‌دارد که واژه FUNCTIONS چیست و چگونه باید از آن استفاده کنید. (دقت کنید که باید عبارات نشان داده شده در تصویر بالا را با حروف بزرگ تایپ کنید.)

```

DEBUGGING          LITERALS          SEQUENCES

help> FUNCTIONS
Functions
*****

Function objects are created by function definitions. The only
operation on a function object is to call it: "func(argument-list)".

There are really two flavors of function objects: built-in functions
and user-defined functions. Both support the same operation (to call
the function), but the implementation is different, hence the
different object types.

See Function definitions for more information.

Related help topics: def, TYPES
help>

```

آن‌چنان‌که با مثال‌های این سری از آموزش‌های کار می‌کنید، به همان نسبت نیاز پیدا می‌کنید تا از راهنمای پایتون

برای دریافت اطلاعات بیشتر در ارتباط با هر یک از دستورات استفاده کنید. اگر به یاد داشته باشید در شماره‌های قبل ما از فرمان `print()` برای نمایش اطلاعات روی صفحه‌نمایش استفاده کردیم. اگر در راهنمای فوق فرمان `print` را تایپ کرده و کلید اینتر را فشار دهید، **پایتون** اطلاعات مفیدی در ارتباط با این فرمان و نحوه استفاده از این فرمان را ارائه می‌کند.

نکته: دقت کنید در این‌جا لزومی ندارد همراه با فرمان `print` از پرانتزها استفاده کنید، زیرا شما در حال تایپ یک فرمان نیستید، بلکه به دنبال کسب اطلاعات در ارتباط با یک فرمان هستید.

```
help> print
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout
    sep:  string inserted between values, default a space.
    end:  string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

help>
```

در حال حاضر راهنمای **پایتون** اطلاعات خیلی زیادی به شما منتقل نمی‌کند، زیرا اطلاعات به شکل حرفه‌ای ارائه می‌شوند و شما هنوز باید نکات زیادی را یاد بگیرید. اما باید شیوه استفاده از راهنمای **پایتون** را بلد باشید.

روش سریع‌تر دسترسی به راهنمای پایتون

ورود به محیط `help` و در ادامه تایپ فرمان‌ها ممکن است کمی خسته کننده و وقت‌گیر باشد. بر همین اساس **پایتون** روش دیگری در اختیاران قرار می‌دهد که به شکل سریع‌تری به راهنما دسترسی داشته باشید. در این روش زمانی که در محیط خط فرمان **پایتون** قرار دارید برای دریافت اطلاعات در ارتباط با یک فرمان می‌توانید از ترکیب نحوی فرمان `help()` استفاده کنید. در این روش باید فرمانی که قصد دارید اطلاعاتی در ارتباط با آن به دست آورید را درون پرانتز همراه با ویرگول‌ها وارد کنید. به‌طور مثال اگر فرمان زیر را در محیط خط وارد کنید فرمان اطلاعات کاربردی در ارتباط با این فرمان دریافت می‌کنید، بدون آن وارد محیط راهنمای اصلی شود. (به دو کاراکتر ' در تصویر زیر دقت کنید که به درستی وارد کرده باشید).

`help('print')`

Python 3.7 (64-bit)

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64-bit]
Type "help", "copyright", "credits" or "license" for more information.
>>> help('print')
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file: a file-like object (stream); defaults to the current sys.stdout
    sep:  string inserted between values, default a space.
    end:  string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.

>>>
```

این روش اجازه می‌دهد به شکل سریع‌تری به اطلاعات راهنما دست پیدا کنید. به‌طور مثال، برای دریافت اطلاعاتی در ارتباط با FUNCTIONS کافی است فرمان زیر را اجرا کنید.

`help('FUNCTIONS')`

Python 3.7 (64-bit)

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64-bit]
Type "help", "copyright", "credits" or "license" for more information.
>>> help('FUNCTIONS')
Functions
*****

Function objects are created by function definitions. The only
operation on a function object is to call it: "func(argument-list)".

There are really two flavors of function objects: built-in functions
and user-defined functions. Both support the same operation (to call
the function), but the implementation is different, hence the
different object types.

See Function definitions for more information.

Related help topics: def, TYPES

>>>
```

روش دیگر دسترسی به راهنما و مستندات پایتون از طریق اجرای محیط توسعه یکپارچه IDLE است که در ادامه با این محیط آشنا خواهید شد.

بستن محیط تعاملی خط فرمان پایتون

پس از آن‌که کار شما با محیط خط فرمان پایتون به اتمام رسید زمان بستن آن فرا می‌رسد. شما برای بستن پنجره اصلی پایتون می‌توانید روی آیکن X در سمت راست بالای پنجره کلیک کنید. در حال حاضر این کار مشکلی برای شما به وجود نمی‌آورد، اما به مرور زمان که روی فایل‌های بزرگ کار می‌کنید نباید از این روش استفاده کنید، زیرا ممکن است فایل‌های باز را ناخواسته ببندید. در نتیجه بهتر است از روش استاندارد استفاده کنید. پایتون دو فرمان `exit()` و `quit()` را ارائه کرده است. هر یک از روش‌های فوق باعث بستن محیط تعاملی پایتون

می‌شوند. هر دو این فرمان‌ها یک مقدار اختیاری را به عنوان ورودی دریافت می‌کنند. در حال حاضر نیازی نیست از آرگومان‌های اختیاری این فرمان استفاده کنید، فقط بدانید که مقدار عددی قرار گرفته درون این دو دستور مقدار متغیر محیطی ERRORLEVEL را تنظیم می‌کند. شما می‌توانید quit() یا exit() را تایپ و کلید اینتر را فشار دهید تا محیط تعاملی بسته شود.

زمانی که هر یک از فرمان‌های فوق را برای بستن محیط تعاملی استفاده می‌کنید، **پایتون** یکسری کارها در پشت صحنه انجام داده و پس از آن محیط تعاملی را می‌بندد. البته دو فرمان دیگر sys.exit() و os.exit() نیز وجود دارند که در مواقع خاص استفاده می‌شوند. در شماره‌های آتی درباره این دو فرمان اطلاعات بیشتری به دست خواهیم آورد.

یک برنامه کاربردی چگونه ساخته می‌شود؟

توسعه نرم‌افزارهای کاربردی فرآیندی دنباله‌دار است. به عبارت دیگر، یکسری فرآیندها پشت سرهم باید به سرانجام برسند تا برنامه‌ای ساخته شود. درست مشابه هر کاری، افراد برای نوشتن برنامه‌ها به ابزارهای مخصوصی نیاز دارند. در ارتباط با **پایتون** شما نیازی ندارید از یک ابزار خاص یا خارجی استفاده کنید، با این حال ابزارهای ثالث به شما کمک می‌کنند، برنامه‌های خود را در زمان کمتر و به بهترین شکل بنویسید. محیط توسعه یکپارچه IDLE ابزاری است که شما در زمان نصب **پایتون**، آن را روی سیستم خود نصب کرده‌اید. ابزارهای قدرتمند دیگری نیز وجود دارند که به جای IDLE می‌توانید برای کدنویسی **پایتون** از آن‌ها استفاده کنید، اما در این سری از آموزش‌ها ما به سراغ ابزارهای دیگر نمی‌رویم، زیرا IDLE به خوبی به نیازهای ما پاسخ می‌دهد. اما پس از پایان این سری از آموزش‌ها پیشنهاد می‌کنیم به شکل جدی به سراغ ابزارهایی همچون Komodo Edit بروید که قابلیت‌های کاربردی بیشتری در اختیاران قرار می‌دهند.

محیط توسعه یکپارچه IDLE

شما می‌توانید از یک ویرایشگر متنی ساده برای ساخت یک برنامه **پایتون** هستید، اما نباید از یک ابزار ساده با حداقل امکانات انتظار زیادی داشته باشید. محیط‌های توسعه یکپارچه تعاملی فرآیند توسعه نرم‌افزارها را بیش از پیش برای توسعه‌دهندگان ساده می‌کنند. IDLE محیط توسعه یکپارچه‌ای است که بیشتر قابلیت‌های اصلی که برای برنامه‌نویسی به آن‌ها احتیاج دارید را در اختیاران قرار می‌دهد. از جمله این قابلیت‌ها به موارد زیر می‌توان اشاره کرد:

- نوشتن کدهای **پایتون**
- تشخیص دستورات و های‌لایت کردن دستورات و متن‌های ویژه
- انجام یک ویراستاری ساده (کپی، برش، انتقال و...) و ویرایش کردن کدها (نشان دادن پرانتزهایی که پیرامون یک عبارت قرار دارند).
- ذخیره و باز کردن فایل‌های **پایتون**
- نمایش مسیر **پایتون** برای دسترسی سریع‌تر به فایل‌ها
- انجام یک اشکال‌زدایی ساده (نشان دادن کدهای خطای موجود در کدها)

اجرای IDLE

IDLE در پوشه‌ای که **پایتون** در آن نصب شده است قرار دارد. برای اجرای IDLE می‌توانید به منوی شروع رفته و درون پوشه **پایتون** گزینه IDLE را انتخاب کنید یا می‌توانید در کادر جست‌وجوی ویندوز 10 کلمه IDLE را نوشته و کلید اینتر را فشار دهید تا محیط یکپارچه اجرا شود.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

شما از محیط توسعه یکپارچه درست به همان شکلی می‌توانید استفاده کنید که از محیط خط فرمان تعاملی پایتون برای اجرای دستورات استفاده می‌کردید. به‌طور مثال اگر فرمان `help()` را در این محیط نوشته و کلید اینتر را بزنید به محیط راهنما وارد شده و همانند مطالبی که در بالا به آن اشاره کردیم قادر به استفاده از راهنمای پایتون هستید.

```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

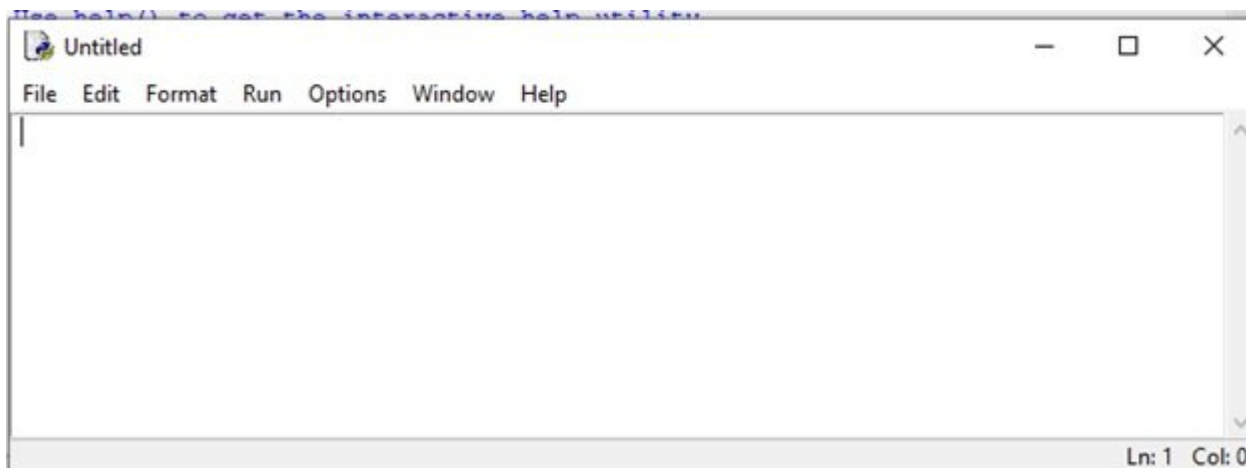
Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics". Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help>
```

اجرای پنجره تعاملی برای درج کدها

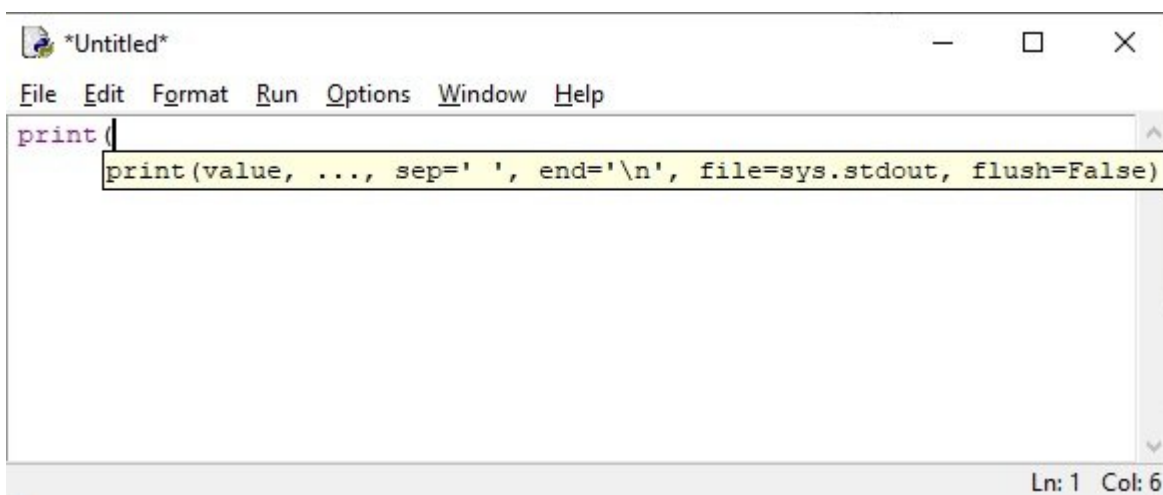
اکنون زمان آن رسیده است که اولین برنامه کاربردی خود را با پایتون ایجاد کنید. پنجره اصلی IDLE برای آشنایی اولیه مناسب است، اما برای تایپ کدها شما به یک پنجره تعاملی نیاز دارید. پنجره‌ای که پس از نوشتن دستورات درون آن بتوانید دستورات را ویرایش کرده یا کدهای خود را ذخیره‌سازی کنید. برای باز کردن یک پنجره تعاملی به مسیر `File → New File` بروید تا پنجره جدیدی ساخته شود. با اینکار پنجره‌ای همانند شکل زیر ظاهر می‌شود.



دقت کنید در عنوان پنجره ظاهر شده واژه Untitled به جای Python 3.3.4 Shell قرار گرفته است. هر دو پنجره اصلی و تعاملی نوار عنوان خاص خود را دارند. پنجره تعاملی به شما اجازه می‌دهد دستورات خود را درون آن تایپ کرده و تنظیمات پنجره را بنا بر نیاز خود تغییر دهید. در حال حاضر شما نیازی ندارید تنظیمات پیش‌فرض این محیط را تغییر دهید، بلکه مهم این است که با ترکیب نحوی و الگوی دستورات در **پایتون** آشنا شوید.

تایپ فرمان‌ها

همانند پنجره اصلی IDLE در پنجره تعاملی نیز شما به راحتی می‌توانید فرمان‌ها را در پنجره فوق تایپ کرده یا ویرایش کنید. این پنجره یک مزیت مهم دارد. هر زمان دستورات **پایتون** را درون آن تایپ می‌کنید، اطلاعات مفیدی در ارتباط با پارامترهایی که یک فرمان دریافت می‌کند در اختیارتان قرار می‌دهد. برای آن‌که ببینید این محیط چگونه کار می‌کند فرمان `print()` را درون این پنجره وارد کنید. دقت کنید پس از تایپ اولین پرانتز پیغامی ظاهر می‌شود که اطلاعات اولیه‌ای در ارتباط با فرمان `print` در اختیارتان قرار می‌دهد.



فرمان `print` را به شکل زیر کامل کنید.

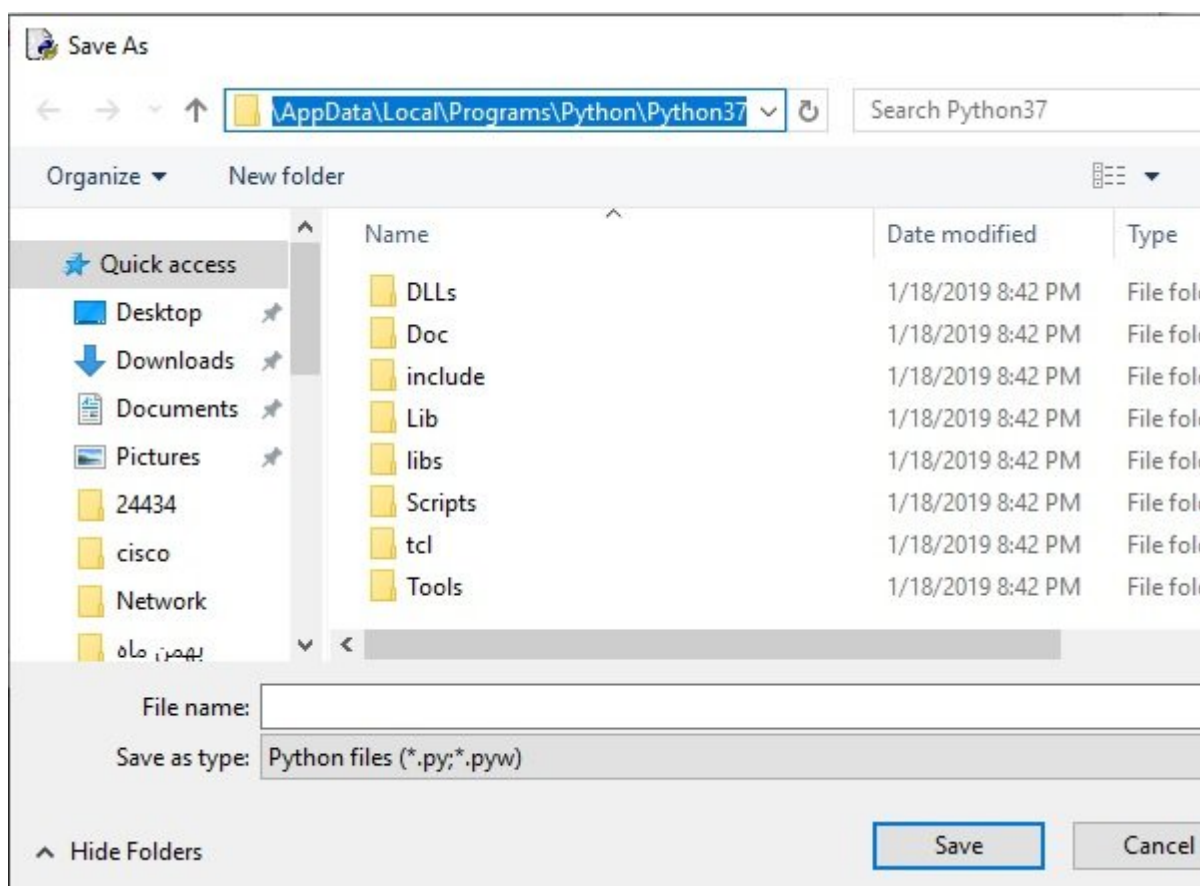
`print("یک مثال ساده از یک برنامه نوشته شده با پایتون")`

برنامه شما اکنون باید مشابه تصویر زیر باشد.



ذخیره کردن فایل

در این مرحله می‌توانید برنامه‌ای که نوشته‌اید را اجرا کنید. با این حال، توصیه می‌کنیم همیشه پیش از اجرای برنامه ابتدا آن را ذخیره کنید تا اگر اتفاق غیرمنتظره‌ای رخ داد، تلاش‌های شما از میان نروند. برای ذخیره کردن فایل به مسیر `File⇒Save` بروید. پنجره‌ای همانند شکل زیر ظاهر می‌شود.



پنجره فوق به‌طور خودکار مسیری که **پایتون** در آن نصب شده است را برای ذخیره کردن فایل پیشنهاد می‌کند. با توجه به این‌که کدهای اصلی **پایتون** در این پوشه قرار دارد، پیشنهاد می‌کنیم فایل‌های خود را در مکان دیگری ذخیره‌سازی کنید. به‌طور مثال، روی درایو اصلی پوشه‌ای به نام MyCodes ایجاد کرده و برنامه خود را به نام `FirstApp.py` درون این پوشه ذخیره کنید.

اگر دومرتبه به عنوان پنجره تعاملی دقت کنید به جای `untitled` نام و مسیر فایل شما قرار گرفته است.

```
print("یک مثال ساده از یک برنامه نوشته شده با پایتون")
```

اجرای برنامه

پایتون به شما اجازه می‌دهد به روش‌های مختلفی برنامه خود را اجرا کنید. ساده‌ترین راه برای اجرای برنامه‌ها رفتن به منوی Run و انتخاب گزینه Run Module است. (به جای این گزینه از میانبر F5 نیز می‌توانید استفاده کنید.) با اینکار برنامه شما اجرا شده و خروجی در پنجره اصلی IDLE همانند شکل زیر نشان داده می‌شود.

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\MyCodes\FirstApp.py =====
یک مثال ساده از یک برنامه نوشته شده با پایتون
>>>
```

دو خط بالایی این خروجی برای شما آشنا است. دو خطی که اطلاعاتی در ارتباط با پایتون، تاریخ اجرای برنامه و... می‌دهند. اما خطوط بعدی که با عبارت `RESTART =====` می‌دهند، مشخص شده‌اند، خروجی برنامه کاربردی شما هستند. هر بار که برنامه خود را اجرا کنید این خروجی را مشاهده خواهید کرد.

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
(AMD64) ] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\MyCodes\FirstApp.py =====
یک مثال ساده از یک برنامه نوشته شده با پایتون
>>>
===== RESTART: C:\MyCodes\FirstApp.py =====
یک مثال ساده از یک برنامه نوشته شده با پایتون
>>>
```

در این شماره آموزش رایگان **پایتون** یاد گرفتیم که چگونه از راهنمای **پایتون** استفاده کرده و اولین برنامه کاربردی

خود در پایتون را نوشته، ذخیره کرده و اجرا کردیم. در شماره آینده به سراغ مبحث تورفتگی‌ها در پایتون و معنای آن‌ها خواهیم رفت.

برای مطالعه تمام قسمت‌های منتشر شده آموزش‌های رایگان پایتون اینجا کلیک کنید
لطفاً نظرات خود در مورد این آموزش و ادامه آن را در بخش دیدگاه در انتهای صفحه اعلام
نمایید و نظارت سایر کاربران را نیز ببینید.

معرفی کانال آموزش برنامه‌نویسی و پایتون:

برنامه‌نویسی | پایتون: [@Python_0to100](#)
(کانال سری آموزش‌های رایگان پایتون)

آموزش پایتون: [@learnpv](#)

تاریخ انتشار:
04 بهمن 1397

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/14481/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D8%B1%D8%A7%DB%8C%DA%AF%D8%A7%D9%86-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86-python-%E2%80%93%D9%86%D8%AD%D9%88%D9%87-%D8%A7%D8%B3%D8%AA%D9%81%D8%A7%D8%AF%D9%87-%D8%A7%D8%B2-%D8%B1%D8%A7%D9%87%D9%86%D9%85%D8%A7%DB%8C-%D9%BE%D8%A7%DB%8C%D8%AA%D9%88%D9%86>