Hossein Seilani
Designer, Developer,SysAdmin

Complete Reference Books
2025
https://predator-store.ir/

Book 2\100

# 1001 Python Practice Examples

1st Edition

**Complete Python Examples Reference**

## Introduction

This book is a complete, practical, and well-structured guide to learning the Python programming language. Whether you're a beginner taking your first steps or an experienced developer looking to strengthen your Python skills, this book is designed to walk you through every essential concept — from the basics to more advanced topics — in a step-by-step manner. Rather than simply listing commands and syntax, the book focuses on real understanding through clear explanations, practical examples, helpful tips, and hands-on exercises. It begins by exploring the history of Python, the reasons behind its creation, and the unique features that make Python one of the most popular languages in the world today.


باسمه تعالی

# ۱۰۰۱ مثال تمرینی پایتون

# 1001 Python Practice Examples

**یادگیری پایتون با مثال**

**نویسندگان**

**حسین سیلانی**

شهریور ۱۴۰۴

تقیم به تو

یگانه فرزانه دلم

## پیشگفتار مولفان

پایتون نمونه ای از یک زبان همه کاره است. این زبان با امتیاز برنامه نویسان در حال حاضر زبان چهارم در جهان است. نکته جالب‌تر در مورد پایتون این است که به راحتی می‌توانید یک برنامه را در یک پلت‌فرم بنویسید و آن را در هر پلت‌فرم دیگر استفاده کنید. پایتون امکان استقلال را در سایر پلت فرم‌ها برای برنامه نویسان فراهم کرده است. قدرت پایتون علاوه برقابلیت خواندن کد و نحو مختصر، این اجازه را به برنامه نویس می‌دهد که برنامه‌های کاربردی را با استفاده از خطوط کمتری بنویسند. بعضی افراد پایتون را به عنوان یک زبان اسکریپت می‌شناسند،درحالی که پایتون چیزی فراتر از آن است. این کتاب در سطح مقدماتی و پیش نیاز سایر مباحث کاربردی پایتون نوشته شده است. تا حد امکان از مثال‌های فراوان برای آموزش بهتر مطالب استفاده شده است.

**شهریور ۱۴۰۴**

پیشنهادات و انتقادات خود را از طریق ایمیل زیر با مولف در میان بگذارید:

hosseinseilany@gmail.com

| | |
|---|---|
| سرشناسه | سیلانی، حسین، ۱۳۶۰، |
| عنوان و نام پدیدآور | برنامه نویسی<br>حسین سیلانی؛ ویراستاری حسین سیلانی. |
| مشخصات نشر | تهران: کیان دانش، ۱۴۰۴. |
| مشخصات ظاهری | ۲۵۰ ص. |
| شابک | ۹۷۸،۶۲۲،۳۰۱،۰۴۵،۶ |
| وضعیت فهرست نویسی | فیپا |
| موضوع | برنامه نویسی پایتون |
| موضوع | برنامه نویسی (کامپیوتر) |
| شناسه افزوده | سیلانی، حسین، ۱۳۶۰، |
| رده بندی کنگره | ۷۶/۷۶QA |
| رده بندی دیویی | ۴۳۲/۰۰۵ |
| شماره کتابشناسی ملی | ۹۶۷۳۵۹۳ |
| اطلاعات رکورد کتابشناسی | فیپا |

نشر کیان دانش: ناشر کتاب‌های دانشگاهی

عنوان: ۱۰۰۱ مثال تمرینی پایتون

تالیف: حسین سیلانی

صفحه آرایی و ویراستاری: حسین سیلانی

چاپ و صحافی: موسسه مهراد

طراحی جلد: حسین سیلانی

چاپ اول: ۱۴۰۴

شمارگان: ۲۰۰ جلد

شابک: ۶،۰۵۵،۴۰۱،۶۲۲،۷۸۶




خیابان ۱۲ فروردین، خیابان شهدای ژاندارمری، شماره ۱۲۶، طبقه چهارم

# فهرست مطالب

## تاریخچه زبان

زبان پایتون سال ۱۹۹۰-۱۹۸۵ در موسسه تحقیقاتی بین المللی هلند به نام CWI توسط جیودو ون راسوم[1] طراحی و نام آن را از نام یک مار به نام پایتون قرار داد. زبان پایتون ترکیبی از زبان های :

ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell

همچنین پایتون تحت مجوزGPL) GNU General Public License) است.

## علت اصلی ایجاد پایتون

طراح زبان پایتون در ابتدا یک زبان اسکریپتی را برای سیستم Amoeba ایجاد کرده بود. پایتون از ابتدا برای استفاده ساده و یادگیری سریع ایجاد شد. به راحتی قابل خواندن، نوشتن و فهمیدن برای یک زبان برنامه نویسی است. همچنین این زبان به صورت رایگان و متن باز[2] ارائه شد.

## ویژگی‌های پایتون

- کاربرد عمومی[3]
- ارتباط متقابل[4]
- شی‌گرا[5]
- قدرتمند برای برنامه نویسی پیشرفته
- نگارش و تایپ ساده تر
- پیچیدگی بسیار کم و تعداد خط کد کمتر
- دارای کتابخانه کد بسیار وسیع
- رفع عیب راحت‌تر و سریع تر
- نزدیک به زبان محاوره سطح بالا[6]
- زبانی قابل حمل و قابل اجرا در همه جا[7]
- زبان کاربرپسند[8]

## سایرویژگی‌ها :

1 Guido Van Rossum
2 Open Source
3 general-purpose
4 interactive
5 object-oriented
6 high-level language
7 portable and native
8 user-friendly

## قابلیت‌ها

همچنین اکثر تهدیدات علیه سازمان‌ها از طریق زبان پایتون صورت می‌گیرد. پایتون تمرکز خوبی بر روی خوانایی، کیفیت نرم افزار، انسجام نسبت به سایر زبان‌های برنامه نویسی دارد.کد پایتون برای قابلیت خواندن طراحی شده است و از این رو قابل استفاده مجدد و نگهداری بهتر نسبت به خیلی از زبان‌های برنامه نویسی سنتی را دارا است. یکنواختی کد پایتون آن را برای درک، آسان می‌کند، زبانی که نیاز به اشکال یابی debug و نگهداری کمتر است و به سرعت و مستقیم اجرا می‌شوند.

## پشتیبانی از کتابخانه‌ها

پایتون با یک مجموعه بزرگی از توابع و کتابخانه‌های کد همراه است و هم چنین کتابخانه‌های استانداردآن شناخته شده است. امروزه پایتون می‌توان آن را با کتابخانه‌های C و جاوا و C ++ و دات نت ادغام کرد.

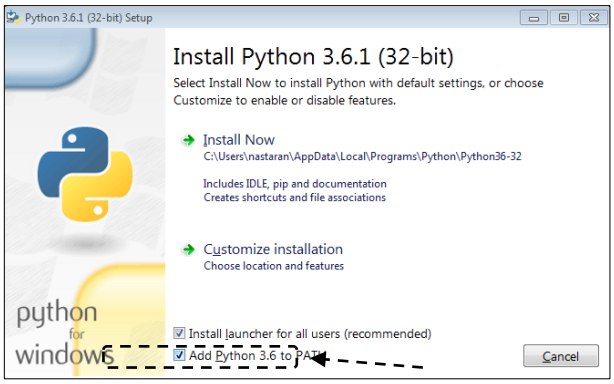
تعداد بیشماری از کتابخانه‌های کد پایتون برای دانلود وجود دارد که برنامه نویس می‌تواند برنامه‌هایی را برای:

- خواندن فایل‌ها
- پردازش داده‌ها
- ارتباطات از طریق وب و شبکه
- ساخت وب سرورها
- طراحی و توسعه وب
- برنامه نویسی شبکه
- برنامه‌های گرافیکی
- برنامه نویسی اسکریپتی برای سیستم عامل
- ایجاد بازی
- ساخت برنامه‌های تجاری
- ساخت نمودارهای آماری

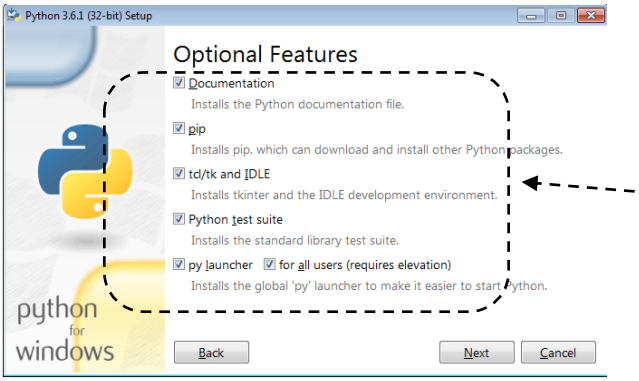
و همچنین می‌تواند با اجزای دات نت، می‌تواند مثل ارتباط با COM و رابط با دستگاه‌ها با پورت‌های سریال در شبکه‌های با رابط‌های تعامل مثل :

SOAP ، XML-RPC، و CORBA.

در پایین پنجره گزینه add python to path را انتخاب نمایید تا خود برنامه متغییرهای محلی را برای سیستم به صورت پیش فرض نصب و تنظیم نماید.

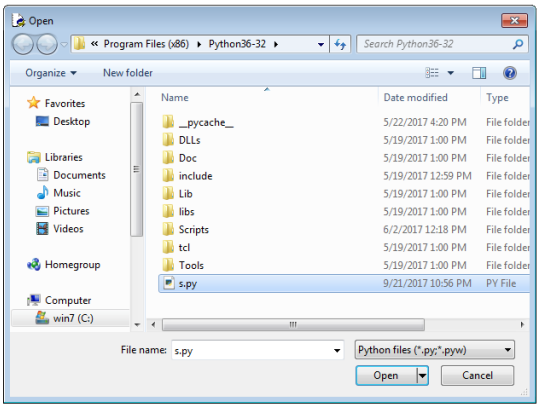



بعد انتخاب گزینه‌ها بر روی گزینه customize installationکلیک نمایید در پنجره بازشده بعدی گزینه‌هایی وجود دارد که می‌توان تعیین نمایید چه پارامترهایی همراه برنامه نصب شود. ازقبیل : راهنما، سندها، محیط توسعه و غیره

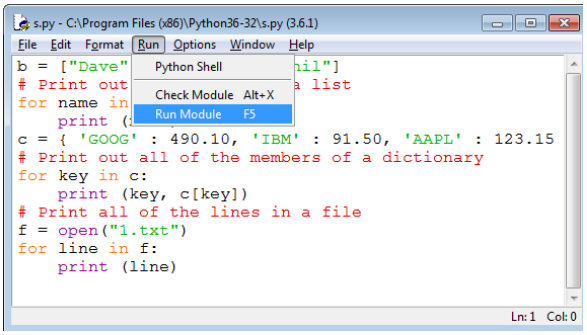



در ادامه بر روی گزینه next کلیک نمایید.

در پنجره باز شده بعدی گزینه‌های از قبیل نصب برای همه کاربران، اجرا شدن همه فایل‌های وابسته با پایتون با برنامه پایتون،ایجاد شدن میانبربرنامه بر روی میزکار، اضافه کردن متغییرهای محلی،کمپایل شدن کتابخانه‌های استاندارد و غیره وجود دارد که می‌توان آن‌ها را انتخاب نمایید.




پس از باز کردن آن می‌توان از طریق منوی run یا با فشار دادن کلید F5 آن را اجرا کرد.




پنجره اسکریپتی دارای سیستم رنگ بندی برای قسمت‌های کدمی‌باشد و به خوبی می‌توان خوانایی برنامه را مشاهده کرد.

رنگ های پیش فرض کدهای برنامه مفسر IDLE

| رنگ | کاربرد | مثال |
|---|---|---|
| مشکی | داده و متغییر ها | ali ۶,۳ |
| سبز | رشته ها | "test" |
| صورتی | توابع | Len |
| نارنجی | دستورات | If for else |
| آبی | توابع کاربر | Get_age |
| قرمز تیره | توضیحات | this is comments# |
| قرمز روشن | پیام های خطا | Syntax |

محل تنظیم و تغییر رنگ دلخواه IDLE

## شروع برنامه نویسی

همانگونه که می‌دانید یکی از اجزای زبان برنامه نویسی، متغییرها و ثوابت در آن هستند. متغییر مقادیری هستند که در طول اجرای برنامه تغییر می‌کنند و ثوابت در طول اجرای برنامه بدون تغییر هستند. همچنین متغییرها و ثوابت در خانه‌های حافظه موقت سیستم ذخیره می‌شوند. هر متغییر اشاره به یک مقدار در خانه‌های حافظه را دارد.


به کمک متغییرها می‌توان مقادیر خود را در خانه‌های حافظه برای مدت زمان دلخواه نگهداری کرد. درجدول زیر نحوه تعریف متغییر هادر سایر زبان‌ها و زبان پایتون نشان داده شده است. همان طور که مشاهده می‌کنید تعریف متغییر به سادگی هرچه تمام رخ داده است. در زبان پایتون نیاز نیست نوع متغییر ها[1] را بیان کنید.

1 loosely typed

**مثال** از عملوندهای ریاضی:

```
a = 10
b = 6
```

```
print('a + y =',a+b)
# Output: a + b = 16
```

```
print('a - b =',a-b)
# Output: a - b = 5
```

```
print('a * b =',a*b)
# Output: a * b = 50
```

```
print('a / b =',a/b)
# Output: a / b = 1.666
```

```
print('a // b =',a//b)
# Output: a // b = 1
```

```
print('a ** b =',a**y)
# Output: a ** b = 1000000
```

## عملوندهای مقایسه ای

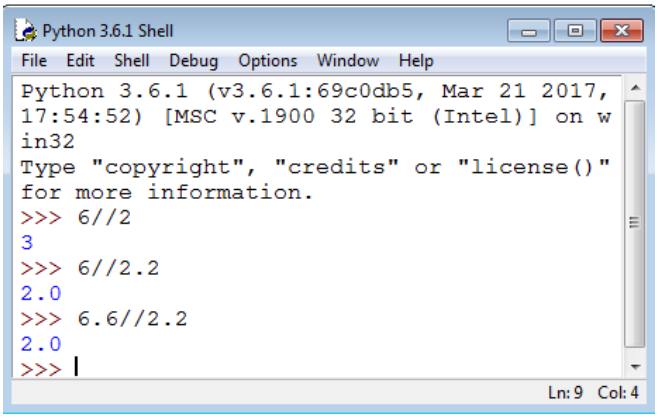
| عملوندهای مقایسه ای در پایتون | | |
|---|---|---|
| عملوند | توضیح | مثال |
| > | بزرگ تر از( عملگر سمت چپ اگر بزرگ باشد مقدار true برمی‌گرداند) | x > y |
| < | کوچک تر از( عملگر سمت چپ اگر کوچک باشد مقدار true برمی‌گرداند) | x < y |
| == | مساوی با( مقدار true بر می گردد، اگر هردو برابر باشند) | x == y |
| != | مساوی نیست با( مقدار true بر می گرداند، اگر مساوی نباشند) | x != y |
| >= | بزرگ ترمساوی با( عملگر سمت چپ اگر بزرگ باشد مقدار true برمی‌گرداند) | x >= y |
| <= | کوچک تر مساوی با( عملگر سمت چپ اگر کوچک باشد مقدار true برمی‌گرداند) | x <= y |

```
# XOR بیتی
bitwise_xor = a ^ b
print(bitwise_xor)  # Output: 1011 (Binary)

# NOT بیتی
bitwise_not = ~a
print(bitwise_not)  # Output: 0101 (Binary)

# Left shift
left_shift = a << 2
print(left_shift)  # Output: 40 (Binary)

# Right shift
right_shift = a >> 2
print(right_shift)  # Output: 2 (Binary)
```

۶. عملوندهای انتصابی:

```
Python
x = 10

# انتساب ساده
x = 20
print(x)  # Output: 20

# انتساب با جمع
x += 5
print(x)  # Output: 25

# انتساب با تفریق
x -= 3
print(x)  # Output: 22

# انتساب با ضرب
x *= 2
print(x)  # Output: 44
```

در این نوع تقسیم خروجی به صورت تقسیم گرد شده[1] نمایش داده می‌شود. به عبارتی آن را خلاصه و کوتاه[2] می‌کند. البته خروجی برحسب نوع مقدار ورودی متفاوت است.




**مثال:**

```
>>> 10 / 4
2.5
>>> 10 / 4.0
2.5
>>> 10 // 4
2
>>> 10 // 4.0
2.0
```

### تفاوت floor , truncate

این دو دستور برای گرد کردن خروجی اعداد به‌کار برده می‌شوند با این تفاوت که truncate خروجی را به مقدار پایین ترین مقدار خودگرد می‌کند که مقداری پایین تراز تابع floor است. مثلا با تابع floor عدد منفی ۲٫۵ به صورت منفی ۳ نشان داده می‌شود و با تابع truncate به صورت منفی ۲.

```
>>> import math
>>> math.floor(2.5)
2
>>> math.floor(-2.5)
-3
```

1 truncating division
2 truncate

```
import math   # This will import math module

print "math.floor(-45.17) : ", math.floor(-45.17)
print "math.floor(100.12) : ", math.floor(100.12)
math.floor(-45.17) :  -46.0
math.floor(100.12) :  100.0
```

### متد cmp()

این متد مخفف compare است و برای مقایسه دو مقدار ورودی به‌کار می‌رود.

### متد exp

برای محاسبه مقدار $e^x$ از متد exp استفاده می‌نماییم.

```
>>>import math
>>>math.exp( x )
>>>  import math
>>>  math.exp(-45.17)
2.4150062132629406e-20
```

### متد fabs

با متد fabs() مقدار مثبت یک عدد را بر می‌گردانیم.

```
>>>import math
>>>math.fabs( x )
>>>math.fabs(-45.17) :  45.17
>>>math.fabs(100.12) :  100.12
```

### متدlog

**از این متد برای گرفتن مقدار لگاریتم یک عدد استفاده می‌شود.**

```
>>>import math
>>>math.log( x )
>>>  math.log(100)
4.605170185988092
>>>  math.log(10)
2.302585092994046
```

**توابع مثلثاتی**

برای ساخت چند عدد راندوم از یک لیست ورودی، کافی است که تعداد عدد دلخواه خود را در انتهای متد وارد نمایید.

```
>>> random.sample(values, 2)
[6, 2]
>>> random.sample(values, 2)
[4, 3]
>>> random.sample(values, 3)
[4, 3, 1]
>>> random.sample(values, 3)
[5, 4, 1]
```

**ایجاد لیست راندم**

گاهی یک لیست وردی داریم و قصد داریم این لیست را به صورت راندوم تغییر دهیم. این تغییر برای هربار استفاده متفاوت خواهد بود. برای این امر از متد shuffle() استفاده می‌کنیم.

```
>>> values = [1, 2, 3, 4, 5, 6]
>>> random.shuffle(values)
>>> values
[2, 4, 6, 5, 3, 1]
>>> random.shuffle(values)
>>> values
[3, 5, 2, 1, 6, 4]
```

**متد randint**

از این متد برای ایجاد عدد راندوم از نوع صحیح استفاده می‌شود. در این متد دامنه اعداد را تعریف می‌کنیم.

```
>>> random.randint(0,10)
2
>>> random.randint(0,10)
5
>>> random.randint(0,10)
0
>>> random.randint(0,10)
7
>>> random.randint(0,10)
10
```

**متد random**

```
youraddress= "iran"
```

## تعریف رشته

برای تعریف متغییر رشته‌ای هم از تک و جفت گیومه می‌توان استفاده نمود. متغییرهای رشته‌ای در خانه‌های حافظه به صورت آدرس دهی شده قرار می‌گیرند. این خانه‌های حافظه با عدد صفر شروع می‌شوند. یعنی آدرس آفست[1] آن از صفر است. علت این که متغییرهای رشته‌ای در خانه‌های حافظه با شماره ایندکس، شماره گذاری می‌شوند این است که بتوان به محتوای خانه‌های رشته با استفاده از شماره ایندکس خانه، دسترسی پیدا کرد.

**مثال**:

```
>>> myname="hossein"
```

| H | o | s | s | e | i | n |
|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Index(0) | Index(1) | Index(2) | Index(3) | Index(4) | Index(5) | Index(6) |

## دسترسی به محتوا

فرض کنید می‌خواهیم به محتوای خانه شماره ۵ دسترسی پیدا نماییم:

```
>>> myname="hossein"
>>> myname[5]
>>> i
```

### دسترسی به محتوای آخرین خانه

```
>>> myname[-1]
>>> n
```

### برگرداندن چند مقدار از محتوای خانه

بدین منظورکافی است با علامت: دامنه آدرس‌های ایندکس را وارد نمایید:

```
>>> myname[1:3]
>>> 'os'
```

1 offset

```
>>>print('Updated animal list: ', animal)
Updated animal list:  ['cat', 'dog', 'guinea pig']
```

علاوه بر یک مقدار، این متد عناصر تکراری را نیز از لیست حذف می‌نماید.

```
# animal list
>>>animal = ['cat', 'dog', 'dog', 'guinea pig', 'dog']

# 'dog' element is removed
>>>animal.remove('dog')

#Updated Animal List
>>>print('Updated animal list: ', animal)
Updated animal list:  ['cat', 'dog', 'guinea pig', 'dog']
```

فرض کنید خواهان حذف یک عنصر از لیست هستیم که این چنین عنصری در لیست وجود نداشته باشد. پس از استفاده از متد، برنامه مفسر خطایی را مبنی بر نبوده عنصر بر می‌گرداند.

```
 # animal list
>>>animal = ['cat', 'dog', 'rabbit', 'guinea pig']

# Deleting 'fish' element
>>>animal.remove('fish')

print('Updated animal list: ', animal)
Traceback (most recent call last):
  File ".. .. ..", line 5, in <module>
    animal.remove('fish')
ValueError: list.remove(x): x not in list
 >>>  mycar =['reno','bmw','benz']
>>>  mycar.remove('mazda')
>>>  mycar
['reno', 'bmw', 'benz']
```

### حذف تمامی عناصر لیست

برای حذف همه عناصر یک لیست کافی است از متد )del( استفاده کنید. بعد از متد del باید نام لیست را وارد نمایید.

```
# Defining a list
>>>list = [{1, 2}, ('a'), ['1.1', '2.2']]
# clearing the list
```

```
>>>del list[:]
>>>print('List:', list)
```

علاوه بر متد del از متد clearنیز به همین صورت استفاده می‌نماییم.

```
The clear() method only empties the given list. It doesn't return any value.
# Defining a list
>>>list = [{1, 2}, ('a'), ['1.1', '2.2']]

# clearing the list
>>>list.clear()

>>>print('List:', list)
List: []
```

### حذف کردن یک عنصر

با متد pop() و با بیان موقعیت عنصرمی‌توان عنصر مورد نظر را حذف کرد. توجه داشته باشید در صورتی که محل عنصر در متد pop بیان نشود، مقدار پیش فرض ۱- را در نظر می‌گیرد که عدد منفی یک به آخرین عنصر اشاره دارد و آن را بر می‌گرداند.

```
 # programming language list
>>>language = ['Python', 'Java', 'C++', 'French', 'C']

# Return value from pop()
# When 3 is passed
>>>return_value = language.pop(3)
>>>print('Return Value: ', return_value)

# Updated List
>>>print('Updated List: ', language)
Return Value:  French
Updated List:  ['Python', 'Java', 'C++', 'C']

>>>  mycar =['reno','bmw','benz']
>>>  mycar.pop(3)
>>>  mycar
['reno', 'bmw', 'benz']
```

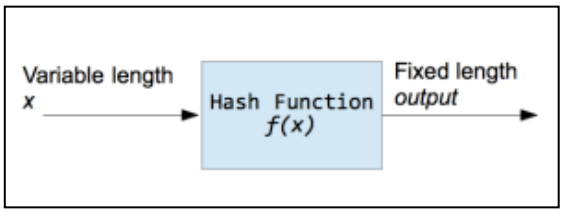
در صورتی که پارامتر وارد شده در متد pop در لیست وجود نداشته باشد با خطای زیر روبرو خواهید شد.

**تغییر مقدار شمارشگر:**

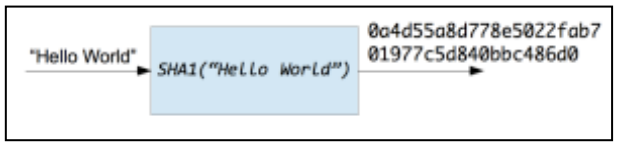
```
>>>enumerateGrocery = enumerate(grocery, 10)
>>>print(list(enumerateGrocery))
[(0, 'bread'), (1, 'milk'), (2, 'butter')]
[(10, 'bread'), (11, 'milk'), (12, 'butter')]
```

**تابع hash()**

از این تابع برای رمز کردن یک مقدار ورودی و گرفتن مقدار رمزشده در خروجی مورد استفاده قرار می‌گیرد.




به عنوان **مثال** :




**تابع hash دارای یک آرگومان ورودی (شی) از نوع integer, string, float می‌باشد.**

```
# hash for integer
print('Hash for 181 is:', hash(181))
# hash for decimal
print('Hash for 181.23 is:',hash(181.23))
# hash for string
print('Hash for Python is:', hash('Python'))
```

**Hash for 181 is: 181**
**Hash for 181.23 is: 530343892119126197**
**Hash for Python is: 2230730083538390373**

```
vowels = ('a', 'e', 'i', 'o', 'u')
>>>print('The hash is:', hash(vowels))
```

**The hash is: -695778075465126279**

### ماژول پسوردgetpass

**با کمک این ماژول می‌توان از کاربر درخواست رمزعبور کرد. درخواست رمز به گونه ای است که نیاز به داشتن نام کاربری نخواهد بود. این ماژول دارای دو متد است:**

متد اول در ساختار ماژول یک متن برای درخواست رمزعبور است.

متد دوم در ساختار ماژول مربوط به کنترل ترمینال های مجازی /dev/tty در سیستم های لینوکسی می‌باشد.

getpass.getpass([prompt[, stream]])

**مثال: دریافت رمز از ورودی**

```
import getpass
p = getpass.getpass()
print(p)
```

**مثال:**

```
import getpass

p = getpass.getpass(prompt='What is your favorite color:  ')
if p.lower() == 'blue':
   print ('Right.')
else:
   print( 'bad !')
```

### ماژول platform

با کمک این ماژول می‌توان دریافت اطلاعات سیستم را دریافت کرد. از آنجایی که پایتون یک زبان چند سکویی است و در زمان کارکردن با مفسر برنامه گاهی نیاز است که اطلاعاتی درباره برنامه پایتون داشته باشیم. این ماژول قابلیت دریافت اطلاعات در هر سیستم را داراست. این ماژول دارای ۴ تابع است:

python_version()و python_version_tuple(): این تابع برای بازیابی نسخه مفسر به‌کار برده می‌شود.

python_compiler() : این تابع برای بازیابی کامپایلر که در مفسر به‌کار گرفته شده است.

python_build() : برای بازیابی رشته نسخه مربوط به مفسرپایتون است.

```
        print("write hello word to file")
else:
    print('File not exists!')
```

### خواندن فایل های فشرده

به منظور خواندن فایل‌های فشرده باید نام ماژول همان فایل را وارد نمایید و سپس با متد open و متد read آنها را بخوانیم.

```
import gzip
with gzip.open('myfile.gz', 'rt') as f:
    text = f.read()
    print(text)
```

نوشتن در فایل

```
import gzip
with gzip.open('myfile.gz', 'wt') as f:
   f.write(text)
```

مثال:

```
import bz2
with bz2.open('myfile.bz2', 'rt') as f:
   text = f.read()
```

```
import bz2
with bz2.open('somefile.bz2', 'wt') as f:
   f.write(text)
```

```
import gzip
f = open('somefile.gz', 'rb')
with gzip.open(f, 'rt') as g:
   text = g.read()
```

### ایجاد فایل موقت

منظور از موقت همان temporary است. فایل و فهرستی برای استفاده کوتاه مدت در زمان اجرای برنامه دلخواه ایجاد می‌شود. ابتدا ماژول مربوط به ایجاد فایل موقت را فراخوانی کرده سپس یک فایل موقت ایجاد می‌نماییم. در ادامه مقادیری را در این فایل موقت قرار می‌دهیم.

```
from PIL import  Image, ImageEnhance

myimg = Image.open("original-image.png")
enhancer = ImageEnhance.Sharpness(myimg)

count = 0.05
new_myimg = enhancer.enhance(count)
new_myimg.save('new-image-1.png')
```

## روشن و تاریک کردن تصویر

برای روش کردن تصویر از متد ImageEnhance.Brightness استفاده می‌نماییم.

```
from PIL import Image, ImageEnhance

im = Image.open("sample-image.png")
enhancer = ImageEnhance.Brightness(im)
```

### تیره ترشدن تصویر

```
count = 0.5
im_output = enhancer.enhance(count)
im_output.save('darkened-image.png')
```

### روشن ترشدن تصویر

```
count = 2
im_output = enhancer.enhance(count)
im_output.save('brightened-image.png')
```

## افزایش کنتراست تصویر

با استفاده از متد Contrast می‌توان کنتراست(وضوح تصویر) را افزایش یا کاهش داد.

```
from PIL import Image, ImageEnhance

im = Image.open("sample-image.png")
enhancer = ImageEnhance.Brightness(im)
```

### کاهش کنتراست

```
count = 0.5
```

```
myregex = r"({2})"
```

بین دو تا پنج مورد را مورد تطبیق قرار می‌دهد.

```
myregex = r"({2,5})"
```

دو یا موارد بیشتر مورد تطبیق قرار می‌دهد.

```
myregex = r"({2,})"
```

بیشتراز ۵ مورد تطبیق قرار می‌دهد.

```
myregex = r"({,5})"
```

یکی از موارد a,b,c,d را مورد تطبیق قرار می‌دهد.

```
myregex = r"([ab-d])"
```

همه موارد به جز موارد a,b,c,d را مورد تطبیق قرار می‌دهد.

```
myregex = r"([^ab-d])"
```

همه موارد با a و A شروع شود تطبیق می‌دهد.

```
myregex = r"( [aA] li )"
```

همه کلمات که با ali یا aly تطبیق پذیر باشد نشان می‌دهد.

```
myregex = r"( al[iy] )"
```

همه اعداد بین ۰ تا ۹ را تطبیق می‌دهد.

```
myregex = r"( [0-9] )"
```

همه اعداد بین a تا z را تطبیق می‌دهد.

```
myregex = r"( [a-z] )"
```

همه اعداد بین a تا z و ۰ تا ۹ را تطبیق می‌دهد.

```
myregex = r"( [a-zA-Z0-9] )"
```

همه موارد را به غیر اعداد ۰ تا ۹ را تطبیق می‌دهد.

```
myregex = r"( [^0-9] )"
```

تطبیق به صورت گروهی صورت می دهد.تطبیق بر روی al و fr انجام می‌دهد.

```
myregex = r"(al..).(fr..)"
```

یک یا موارد بیشتر مورد تطبیق قرار می‌دهد.

```
myregex = r"al +"
```

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
a.sort()
print("Largest element is:",a[n-1])
```

مثال ۲۹:پیدا کردن دومین عدد بزرگ از لیست ورودی

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=int(input("Enter element:"))
    a.append(b)
a.sort()
print("Second largest element is:",a[n-2])
```

مثال ۳۰ :مرتب کردن یک لیست ورودی:

```
a=[]
n=int(input("Enter number of elements:"))
for i in range(1,n+1):
    b=input("Enter element:")
    a.append(b)
a.sort(key=len)
print(a)
```

مثال ۳۱:جایگزین کردن حرف دلخواه در یک رشته

```
string=input("Enter string:")
string=string.replace('a','$')
string=string.replace('A','$')
print("Modified string:",string)
print(string)
```

مثال ۳۲:بررسی مشابه بودن دو رشته ورودی:

```
s1=input("Enter first string:")
s2=input("Enter second string:")
if(sorted(s1)==sorted(s2)):
    print("The strings are similar.")
else:
```

## برنامه مدیریت IP سیستم

نمایش IP محلی(Local IP)

نمایش IP عمومی(Public IP)

نمایش نام سیستم(Hostname)

نمایش همه‌ی آدرس‌های آی‌پی کارت‌های شبکه

```
import socket
import requests
import psutil

def get_hostname():
    return socket.gethostname()

def get_local_ip():
    return socket.gethostbyname(socket.gethostname())

def get_public_ip():
    try:
        return requests.get("https://api.ipify.org").text
    except:
        return "Unable to fetch public IP"

def get_all_ips():
    addrs = psutil.net_if_addrs()
    ips = []
    for iface, addr_list in addrs.items():
        for addr in addr_list:
            if addr.family == socket.AF_INET:
                ips.append((iface, addr.address))
    return ips

while True:
    print("\n\033[95m--- IP Management Menu ---\033[0m")
    print("1. Show Hostname")
    print("2. Show Local IP")
    print("3. Show Public IP")
    print("4. Show All Network Interfaces")
    print("5. Exit")

    choice = input("\033[93mChoose an option (1-5): \033[0m")

    if choice == "1":
```

**MY PROJECTS**

*seilany.ir*
*learninghive.ir*
*artystone-os.ir*
*emperor-os.ir*
*predator-os.ir*
*little-psycho.ir*

**CONTACT ME**

*telegram: @seilany*

LEARNING HIVE

# HosseinSeilani

## Designer,Developer and Linux sysadmin

I am Hossein Seilani, a Master of Science in Computer Science, and the founder and developer of Emperor-OS, Little-Psycho, and Predator-OS Linux, with extensive experience and certifications in Linux/Windows System Administration, UX/UI Design, Front-End Web Development, SEO, Graphic Design, Data Science, and Machine Learning.

LITTLE-PSYCHO
DESTRUCTIVE LINUX

Emperor-OS

Predator-OS