برنامه نویسی به زبان

پایتون

مرجع کامل

تالیف
مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرید

hadikiamarsi@gmail.com

فصل ششم

در این فصل مطالب زیر را خواهید آموخت

بازکردن و بستن فایل ها

باز کردن فایل با تابع open

صفت های شی فایل

بستن فایل

خواندن و نوشتن فایل

متد write

متد read

موقعیت اشاره گر در فایل

تغییر نام و پاک کردن فایل ها

تغییر نام فایل

پاک کردن فایل

دایرکتوری ها در پایتون (python)

ایجاد دایرکتوری

تغییر مسیر

دریافت آدرس دایرکتوری جاری

پاک کردن دایرکتوری

ورودی خروجی فایل ها در پایتون

بازکردن و بستن فایل ها

در فصل ها ی قبل خواندن و نوشتن در ورودی استاندارد و صفحه کلید توضیح داده شد و در این فصل خواندن و نوشتن در فایل ها توضیح داده می شود

در زبان برنامه نویسی پایتون (python) توابع و متدهای فراوانی برای دستکاری فایل ها وجود دارد که در این فصل به تفصیل توضیح داده می شوند

باز کردن فایل با تابع open

در زبان برنامه نویسی پایتون (python) پیش از آنکه بتوان خواندن و نوشتن فایل را شروع نمایید باید فایل را باز نمایید برای باز کردن فایل در زبان برنامه نویسی پایتون (python) از تابع open() استفاده می گردد این تابع بعد از باز کردن فایل یک اشاره گر از آن بر می گرداند که به اشاره گر فایل معروف می باشد .

ساختار نحوی

```
file object = open(file_name [, access_mode][, buffering])
```

در زیر توضیحات آرگومان های تابع open() آورده شده است

آرگومان file_name در این قسمت آدرس به همراه نام فایل نوشته می شود

آرگومان buffering در این قسمت اگر چیزی نوشته نشود یا عدد 0 قرار بگیرد به فایل بافر اختصاص داده نمی شود ولی اگر عدد 1 نوشته شود به فایل به اندازه یک خط بافر اختصاص داده می شود ولی اگر عدد بزرگتر از 1 نوشته شود به همان اندازه بافر اختصاص داده می شود و اگر عدد منفی قرار گیرد خود سیستم اندازه بافر را تعیین می نماید

آرگومان access_mode در این قسمت نحوه باز کردن فایل نوشته می شود که گزینه های این گزینه در جدول زیر آورده شده است

حالت های بازکردن فایل به همراه توضیحات
R باز کردن فایل در حالت خواندنی
Rb بازکردن فایل در حالت خواندنی باینری
r+ بازکردن فایل در حالت خواندنی نوشتنی
rb+ بازکردن فایل در حالت خواندنی نوشتنی به صورت باینری
W باز کردن فایل در حالت نوشتنی
Wb باز کردن فایل در حالت نوشتنی به صورت باینری
w+ باز کردن فایل در حالت نوشتنی و خواندنی
wb+ بازکردن فایل در حالت نوشتنی و خواندنی به صورت باینری
A بازکردن فایل به صورتی که بتوان به انتهای آن اطلاعات اضافه کرد
Ab بازکردن فایل به صورتی که بتوان به انتهای آن اطلاعات اضافه کرد به صورت باینری
a+ بازکردن فایل به صورت خواندنی و نوشتنی به صورتی که بتوان اطلاعاتی به انتهای آن اضافه نمود
ab+ بازکردن فایل به صورت خواندنی و نوشتنی به صورتی که بتوان اطلاعاتی به انتهای آن اضافه نمود به صورت باینری

صفت های شی فایل

لازم به ذکر است وقتی یک فایل را باز می نمایید شی فایل آن فایل دارای صفت هایی هست که اطلاعات زیادی را در اختیار برنامه نویس قرار می دهد در زیر لیست این صفت ها به همراه توضیحات آن آمده است

صفت های شی فایل به همراه توضیحات
file.closed اگر فایل بسته شده باشد `True` را بر می گرداند و در غیر این صورت `False` را بر می گرداند
file.mode حالتی را که فایل با آن باز شده است را بر می گرداند
file.name نام فایلی که باز شده است را باز می گرداند
file.softspace اگر فضای کافی برای نوشتن در فایل موجود نباشد `False` را بر می گرداند و در غیر این صورت `True` را بر می گرداند

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

# Open a file
fo = open("foo.txt", "wb")
print "Name of the file: ", fo.name
print "Closed or not : ", fo.closed
print "Opening mode : ", fo.mode
print "Softspace flag : ", fo.softspace
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Name of the file:  foo.txt
Closed or not :  False
Opening mode :  wb
Softspace flag :  0
```

بستن فایل

بستن فایل ها در زبان برنامه نویسی پایتون (python) بوسیله تابع close انجام می پذیرد با بستن فایل تمام اطلاعات نوشته نشده در فایل که هنوز تو بافر یا کچ ذخیره شده اند بر روی فایل نوشته می شوند

البته زبان برنامه نویسی پایتون (python) این قابلیت را دارد که با اختصاص اشاره گر فایل برای باز کردن یک فایل دیگر ابتدا به صورت خودکار آن را ببندد

ساختار نحوی

```
fileObject.close()
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

# Open a file
fo = open("foo.txt", "wb")
print "Name of the file: ", fo.name

# Close opend file
fo.close()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Name of the file:  foo.txt
```

خواندن و نوشتن فایل

تا اینجا فقط یاد گرفتیم فایل ها را باز و بسته نماییم و لی هنوز این قابلیت را نداریم که از آن ها چیزی بخوانیم یا در آن ها چیزی بنویسیم . در زبان برنامه نویسی پایتون (python) خواند و نوشتن در فایل ها با استفاده از تابع های read() و write() انجام می پذیرد

متد write

برای نوشتن اطلاعات در فایل ها در زبان برنامه نویسی پایتون (python) از متد write() استفاده می گردد بیاد داشته باشید این متد کاراکتر خط جدید ("\n") را اضافه نمی کند

ساختار نحوی

```
fileObject.write(string)
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

# Open a file
fo = open("foo.txt", "wb")
fo.write( "Python is a great language.\nYeah its great!!\n")

# Close opend file
fo.close()
```

در مثال بالا فایل `foo.txt` به صورت باینری باز شده است

```
Python is a great language.
Yeah its great!!
```

متد read

در زبان برنامه نویسی پایتون (`python`) خواندن اطلاعات از فایل ها بوسیله متد `read()` انجام می پذیرد

ساختار نحوی

```
fileObject.read([count])
```

این تابع اگر هیچ آرگومانی نداشته باشد تمام محتویات فایل را یکجا می خواند ولی اگر به عنوان آرگومان یک عدد دریافت نماید به اندازه عدد بایت از فایل می خواند

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

# Open a file
fo = open("foo.txt", "r+")
str = fo.read(10);
print "Read String is : ", str
# Close opend file
fo.close()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Read String is :  Python is
```

موقعیت اشاره گر در فایل

در زبان برنامه نویسی پایتون (python) برای تعیین مکان اشاره گر فایل در فایل از مند tell() استفاده می گردد برای مثال اگر 10 بایت از فایل را خوانده باشید تابع tell() موقعیت 10 را نشان خواهد داد و برای تعیین موقعیت اشاره گر فایل در فایل از متد seek استفاده می گردد با استفاده از این متد می توانید یک فایل را به صورت تصادفی بخوانید اگرآرگومان این متد را 0 قرار بدهید در واقع اشاره گر را به ابتدای فایل منتقل کرده و می توانید تابع را از ابتدا بخوانید

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

# Open a file
fo = open("foo.txt", "r+")
str = fo.read(10)
print "Read String is : ", str

# Check current position
position = fo.tell()
print "Current file position : ", position

# Reposition pointer at the beginning once again
position = fo.seek(0, 0);
str = fo.read(10)
print "Again read String is : ", str
# Close opend file
fo.close()
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Read String is :  Python is
Current file position :  10
Again read String is :  Python is
```

تغییر نام و پاک کردن فایل ها

در زبان برنامه نویسی پایتون (python) برای پردازش فایل ها از متدهای ماژول os استفاده می گردد برای استفاده از این ماژول در ابتدا باید آن را با دستور import به کد برنامه نویسی خود الحاق نمایید

تغییر نام فایل

برای تغییر نام یک فایل در زبان برنامه نویسی پایتون (python) از متد rename() استفاده می گردد

ساختار نحوی

```
os.rename(current_file_name, new_file_name)
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

import os

# Rename a file from test1.txt to test2.txt
os.rename( "test1.txt", "test2.txt" )
```

پاک کردن فایل

برای پاک کردن یک فایل در زبان برنامه نویسی پایتون (python) از متد remove() استفاده می گردد

ساختار نحوی

```
os.remove(file_name)
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

import os

# Delete file test2.txt
os.remove("text2.txt")
```

دایرکتوری ها در پایتون (python)

برای کار با دایرکتوری ها در زبان برنامه نویسی پایتون (python) از ماژول os استفاده می گردد این ماژول متدهای زیادی برای کار بر روی دایرکتوری ها در اختیار برنامه نویس قرار می دهد که در زیر بعضی از این مند توضیح داده می شوند

ایجاد دایرکتوری

برای ساخت دایرکتوری در زبان برنامه نویسی پایتون (python) از متد mkdir() استفاده می گردد

ساختار نحوی

```
os.mkdir("newdir")
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

import os

# Create a directory "test"
os.mkdir("test")
```

تغییر مسیر

برای تغییر مسیر جاری از متد chdir() استفاده می گردد این متد یک آرگومان بیشتر نداره و آن هم مسیر جدید می باشد که قصد دارید به آن تغییر مسیر بدهید

ساختار نحوی

```
os.chdir("newdir")
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

import os

# Changing a directory to "/home/newdir"
os.chdir("/home/newdir")
```

دریافت آدرس دایرکتوری جاری

برای بدست آوردن آدرس دایرکتوری جاری در زبان برنامه نویسی پایتون (python) از متد getcwd() استفاده می گردد

ساختار نحوی

```
os.getcwd()
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

```
#!/usr/bin/python

import os

# This would give location of the current directory
os.getcwd()
```

پاک کردن دایرکتوری

برای پاک کردن یا حذف یک دایرکتوری در زبان برنامه نویسی پایتون (python) از متد rmdir() استفاده می گردد لازم به ذکر است که این متد زمانی به صورت صحیح کار می کند که دایرکتوری خالی باشد به عبارتی از قبل باید محتوای دایرکتوری را پاک نمایید سپس دایرکتوری را پاک نمایید

ساختار نحوی

```
os.rmdir('dirname')
```

برای آشنایی بیشتر با این مبحث به مثال زیر توجه نمایید

در مثل زیر دایرکتوری "/tmp/test" پاک می گردد

```
#!/usr/bin/python

import os

# This would  remove "/tmp/test"  directory.
os.rmdir( "/tmp/test"  )
```