



Python lecture 5

جلسه قبل

- سازمان بندی کد و مخفی سازی جزئیات
- انتزاع
- تقسیم وظایف
- توابع
- محدوده متغیرها
- معرفی `python-nmap`
- معرفی `ftplib`

امروز

- مدیریت استثناء
- معرفی ماژول sys
- معرفی ماژول os
- معرفی ماژول cryptography
- معرفی ماژول zipfile

معرفی استثناء

- استثناء زمانی رخ می‌دهد که کد در حالت عادی به درستی کار می‌کند ولی زمانی که اتفاق غیر منتظره ای رخ دهد برنامه توانایی برخورد مناسب با آن را نداشته باشد


```
def check_ftp_connection():  
    f=FTP( 'p30download.com' )  
    f.login()  
    print(f)  
check_ftp_connection()
```

مثلاً اگر اتصال اینترنت برقرار نباشد
در این خط خطا رخ می‌دهد

مدیریت استثناء

- استفاده از بلاک try-except
- زمانی که خطایی رخ می‌دهد اجرا به بلاک except منتقل می‌شود

کلمه کلیدی تعریف یک بلاک مدیریت استثناء

try:  دستور ۱

دستور ۲

except class_name:
کدهای مدیریت خطا

مدیریت استثناء

```
def exception_handles_ftp():  
    try:  
        f=FTP( 'p30download.com' )  
        f.login()  
    except socket.gaierror:  
        print("check your connection")
```

ملژول sys

- امکان دریافت آرگومان‌های ورودی در هنگام اجرای برنامه
- امکان تغییر ورودی و خروجی استاندارد پایتون (به مثلاً یک فایل)
- امکان دریافت اطلاعاتی مربوط به مفسر پایتون
- امکان خروج از مفسر پایتون

دریافت آرگومان‌های ورودی

```
python3 test.py salam
```

آرگومان‌های ورودی که به مفسر پایتون ارسال می‌شوند

- دریافت آرگومان‌های ورودی

```
import sys  
print(sys.argv)
```

آرایه‌ای از آرگومان‌های ورودی

تغییر خروجی استاندارد پایتون

- می‌توانیم به جای نمایش خروجی بر روی کنسول خروجی‌های برنامه را در یک فایل یا هر چیز دیگری ذخیره کنیم

```
import sys  
sys.stdout=open( 'f' , 'w' )  
print( 'salam' )
```

دریافت اطلاعاتی راجع به مفسر پایتون

`sys.builtin_module_names` → لیست ماژول‌های استاندارد پایتون

`sys.executable` → مسیر فایل اجرایی پایتون

`sys.getsizeof(str)` → فضای رزرو برای نوع داده رشته 400 بایت

خروج از مفسر پایتون

```
sys.exit(10)
```

معرفی ماژول OS

- امکان استفاده از ویژگی‌های انحصاری هر سیستم عامل
 - اجرای یک دستور خاص در command line
 - تغییر مسیر جاری که برنامه در آن در حال اجرا است
 - دریافت مسیر جاری که برنامه در آن در حال اجرا است
 - دریافت نام کاربری که در حال اجرای برنامه است
 - دریافت تعداد هسته‌های پردازنده سیستم

مثال‌هایی از ماژول OS

```
import os
os.system('ls') # list the current files in current
directory
print(os.getcwd()) # shows the current working
directory
os.chdir('/') # change the current directory to /
print(os.getlogin()) # show the current user name
print(os.cpu_count()) #shows the number of cpus
```

معرفی ماژول cryptography

- استفاده جهت رمزنگاری متقارن با استفاده از الگوریتم AES 128
- ایجاد کلید جهت رمزنگاری و رمز گشایی متن
- تضمین امنیت و عدم تغییر متن رمز شده

```
from cryptography.fernet import Fernet
key = Fernet.generate_key()
f=Fernet(key)
token=f.encrypt(b"secret message")
print(token)
original_string=f.decrypt(token)
print(original_string)
```

معرفی مازول zipfile

- به منظور فشرده‌سازی یا از حالت فشرده خارج کردن یک فایل zip
- امکان استفاده از رمز در هنگام از حالت فشرده خارج کردن
- مثال کرک رمز یک فایل zip با استفاده از یک دیکشنری رمز عبور