

RYKA|CBT

پدرام شاه صفی
pd.Shahsafi@gmail.com



python



به نام پروردگار دانایی

برنامه نویسی به سبک پایتون

پدرام شاه صفی

تابستان ۱۳۹۴

مدیریت استشنا

فصل هفتم

فصل ششم

فصل پنجم

فصل چهارم

فصل سوم

فصل دوم

فصل اول

برنامه نویسی در

سطح اول

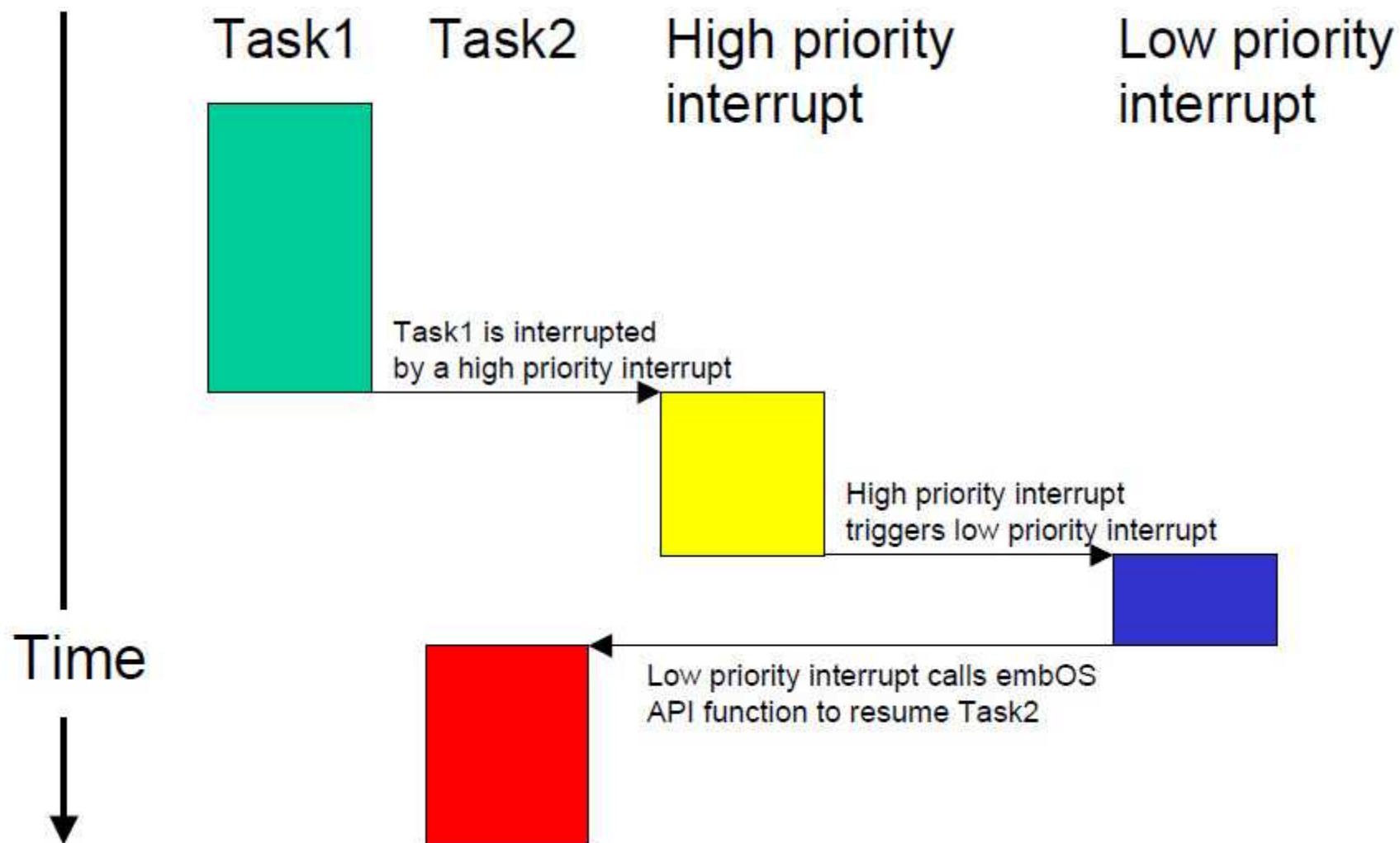
بخش اول

برنامه نویسی به

سبک پایتون



وقفه



VS

• **Interrupt :**

• یک سیگنال از
یا
است که موجب **تغییر** **ترتیب** اجرای برنامه ها
میشود.

• **Exception :**

• یک **event** است که موجب **تغییر** **اجرای** برنامه میشود.

• **Error :**

- **Syntax errors**.
- **Logic errors (bug)**.
- **run-time error**.

از دیدگاه پایتون:

- **Exception** یک **object**.
- **Error** ها از کلاس **Exception** هستند.
- **Exception** ها از کلاس **BaseException** هستند.

```
10 * (1/0)
4 + spam*3
'2' + 2
```

نگاهی به عقب

```
with open("file") as file_name:  
    do_something()  
  
assert always_true()
```

Metaprogramming



"Metaclasses are deeper magic than 99% of users should ever worry about. If you wonder whether you need them, you don't (the people who actually need them know with certainty that they need them, and don't need an explanation about why)."

— Tim Peters

شکل کلی

```
try :  
    do_something()  
except Exception as e:  
    print('Error.\n error type= {0}'.format(e))  
    print(type(e))
```

import coffee

اجرای خطا

```
raise NameError('HI dear!')

f=open("e:/this.txt",'r')
for i in f:
    raise IOError

with open("e://this.txt",'r') as f:
    for i in f:
        raise IOError

f.closed
```

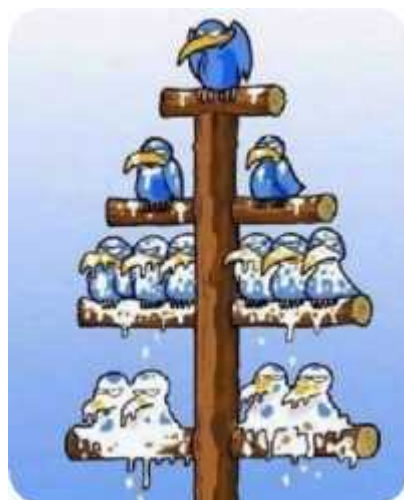
```
if 'variable' in locals().keys():
    pass
else:
    raise NameError
    print (e.__doc__)
assert 'variable' in locals().keys()
```

```
while(alive)
{
    eat( );
    sleep( );
    code( );
}
```


<u>Exception Name</u>	<u>Description</u>	<u>Exception Name</u>	<u>Description</u>
Exception	Base class for all exceptions.	StopIteration	Raised when the next() method of an iterator does not point to any object.
SystemExit	Raised by the sys.exit() function.	StandardError	Base class for all built-in exceptions except StopIteration and SystemExit.
ArithmeticError	Base class for all errors that occur for numeric calculation.	BufferError	Raised when a buffer related operation cannot be performed.
OverflowError	Raised when a calculation exceeds maximum limit for a numeric type.	AssertionError	Raised in case of failure of the Assert statement.
KeyboardInterrupt	Raised when the user interrupts program execution, usually by pressing Ctrl+c.	NameError	Raised when an identifier is not found in the local or global namespace.

مدیریت استثناء

```
try:
    input()
except IOError as e:
    print('Error= ', e)
except NameError as e:
    print('Error= ', e)
except Exception:
    print('there is Exception.')
except : # BaseException
    print('something wrong!')
else:
    print('there is no exception.')
finally:
    print('bye!')
```



صفحه کلید

```
import time
def main():
    run_while()
    make_delay()
def make_delay():
    time.sleep(3)
    print('\nready to exit...')
    print('\nclosing all programs',end='')
    for i in range(10):
        time.sleep(1)
        print('.',end='')
    print('\nall done!\ngoodbye... :)')
    time.sleep(3)
    exit(0)
def run_while():
    try:
        while True :
            print('\ntry again...')
    except KeyboardInterrupt): # the parens are necessary for Python 3
        print('\nstopped by KeyboardInterrupt!')
if __name__ == '__main__':
    main()
```

کار با فایل

```
def dec_open(func):
    def open_file(file_obj=None, **kwargs):
        try:
            f=open(kwargs['file_name'],kwargs['file_mode'])
            ret=func(f)
        except Exception as e:
            print('Error= ',e)
        finally:
            if f and not f.closed:
                f.close()
    return open_file

@dec_open
def working_with_file(file_obj=None, **kwargs):
    file_obj.write('simulate with by decorator\n')
    return 0

working_with_file(file_name='e:/this.txt',file_mode='w')
```



working_with_file(file_name='e:/this.txt',file_mode='w')

working_with_file

شکستن کلمه عبور

```
import zipfile
zFile = zipfile.ZipFile('e:/evil.zip','r')
passFile = open('e:/dictionary.txt','rb')
counter=0
for line in passFile:
    password = line.strip()
    try:
        zFile.extractall(pwd=password)
        break
    except Exception:
        counter+=1
        print("{0} check!".format(counter))

print ('[+] Password = ' , password )
```



```
print ('[+] Password = ' , password )
```

```
print ('[+] Password = ' , password )
```

تمرین

- با استفاده از روش brute force رمز یک فایل زیپ را کرک کنید.
- برنامه ای بنویسید که از کاربر یک سری ورودی از پیش مشخص بگیرد. کاربر سعی میکند به طور عمدی ورودی ها را اشتباه وارد کند. با error handling جلوی این کار او را بگیرید و او را مجبور کنید مقدار صحیح را وارد کند.
- برنامه ای که پسورد تولید میکند. از کاربر بخواهید میزان قوی بودن پسورد را مشخص کند و برا اساس آن تعدادی کاراکتر که از حروف کوچک و بزرگ و اعداد و کاراکتر ها بطور تصادفی تولید شده را به او برگردانید.
- برنامه ای بنویسید که تاریخ دیروز یا امروز یا فردا را از شما بگیرد و متناسب با آن پیغامی را چاپ کند اگر ورودی غیر ازین سه روز باشد خطا برگرداند.
- برنامه ای بنویسید که تمام خطاهای استاندارد پایتون را غیر فعال کند.
- برنامه ای بنویسید که هر آنچه که کاربر از طریق کیبرد در هر برنامه ای وارد میکند ذخیره کند.
- کد زیر چه کاری انجام میدهد ؟

```
def f(x):
    return {
        'a': 1,
        'b': 2,
    }.get(x, 9)
```


پاسخ

```
import random

def make_pass(user_choice):

    if user_choice=='low':
        passlen=random.randrange(3,5)
    elif user_choice=='medium':
        passlen=random.randrange(6,10)
    elif user_choice=='high':
        passlen=random.randrange(11,20)
    else:
        return("user_choice = low/medium/high")

    s = "abcdefghijklmnopqrstuvwxyz01234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()?"
    p = "".join(random.sample(s,passlen ))
    return p
```

any questions?



RANKA | CBT