

RXKA|CBT

پدرام شاه صفی
pd.Shahsafi@gmail.com



python

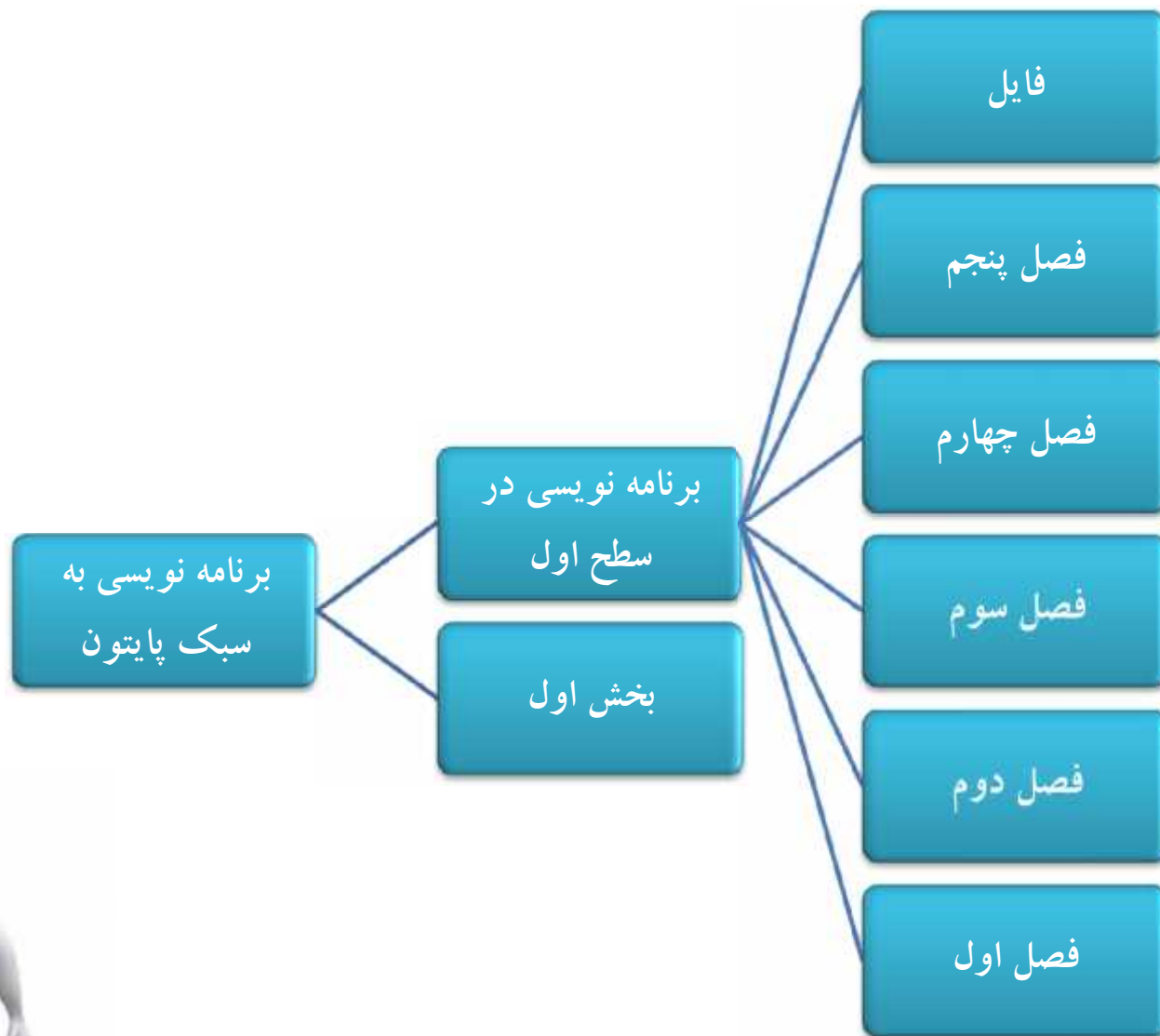


به نام پروردگار دانایی

برنامه نویسی به سبک پایتون

پدرام شاه صفی

تابستان ۱۳۹۴



فایل



نحوه کار

- ابتدا باید file را **باز** کنید. ← `file.open()`
- در این مرحله سه انتخاب وجود دارد.
- **خواندن** از file. ← `file.open(read)`
- **نوشتن** در file (از ابتدای فایل). ← `file.open(write)`
- **اضافه کردن** به محتوای file. ← `file.open(append)`
- file را **ببندید**. ← `File.close()`



تعریف فایل

```
file_name=open("e://this.txt")
file_name=open("e://this.txt",'w')
file_name=open("e://this.txt",'r')
file_name=open("e://this.txt",'a')
file_name=open('e://this.txt','r+')
file_name.readable()
file_name.writable()
```



بستن فایل

```
file_name=open('e://this.txt')

file_name.close()
```



| <u>Method</u> | <u>Description</u> | <u>Method</u> | <u>Description</u> |
|---------------|--|-------------------------|---|
| read(size) | returns it as a string or bytes object. <i>When size is omitted or negative, the entire contents of the file will be read and returned.</i> | tell() | returns an integer giving the file object's current position in the file. |
| readline() | reads a single line from the file. | seek(offset, from_what) | To change the file object's position. |
| readlines() | read all the lines of a file in a list. | close() | to close it and free up any system resources taken up by the open file. |
| readable() | <i>Returns True or False</i> | write(string) | writes the contents of <i>string</i> to the file, returning the number of characters written. |
| writable() | <i>Returns True or False</i> | writelines() | writes a sequence of strings to the file. The sequence can be any iterable object producing strings, typically a list of strings. |

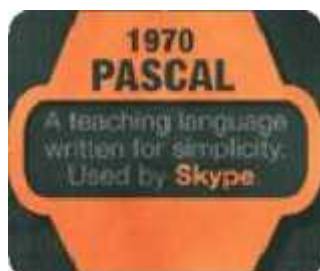
خواندن

```
file_name=open('e://this.txt','r')
file_name.read(1)
file_name.readline()
file_name.readlines()
```



خواندن یکجای تمام محتوا

```
file_name=open('e://this.txt','r')  
file_name.readlines()  
list(file_name)  
file_name.close()
```



نوشتن در فایل

```
poem = '''
Programming is fun
When the work is done
if you wanna make your work also fun:
use Python!
'''

f = open('poem.txt', 'w') # open for 'w'riting
f.write(poem) # write text to file
f.writelines(poem)
f.close() # close the file
```

**1983
C++**

One of the most common languages used today. Widely used in the software industry in systems software, application software, device drivers, embedded software and high-performance server and client applications such as

**Microsoft Office,
Adobe PDF Reader
and Firefox.**

خواندن و نوشتن فارسی

```
f=open('e://this.txt','r',encoding='UTF-8')  
f.readline()
```



نوشتن غیر رشته ها در فایل

```
import json
list_or_tuple= ('1','3',"this is string")
f = open('e://file.txt', 'w') # open for 'w'riting
json.dump(list_or_tuple,f)
f.write(list_or_tuple)#!
f.writelines(list_or_tuple) # write text to file
f.close() # close the file
```



راه حل

```
import json
f=open('e:\\this.txt','w')
x={1:1,2:2}
json.dump(x,f)
f.close()
f=open('e:\\this.txt','r')
x = json.load(f)
type(x)
```



ذخیره

```
message="Hello!"
f = open('e://file.txt', 'w') # open for 'w'riting
f.writelines(message)
#go to file
f.flush()
#go to file
```

**1991
JAVA**

Similar to C++ it allows for "write once, run anywhere" portability across the Net. JAVA is the language for the **Android** operating system.

مثلا

```
f=open("e://this.txt",'r+')
spaces, tabs= 0 ,0
for line in f:
    spaces+=line.count(' ')
    tabs+=line.count('\t')
f.write(str(spaces))
f.write(str(tabs))
f.close()
```

**1993
RUBY**

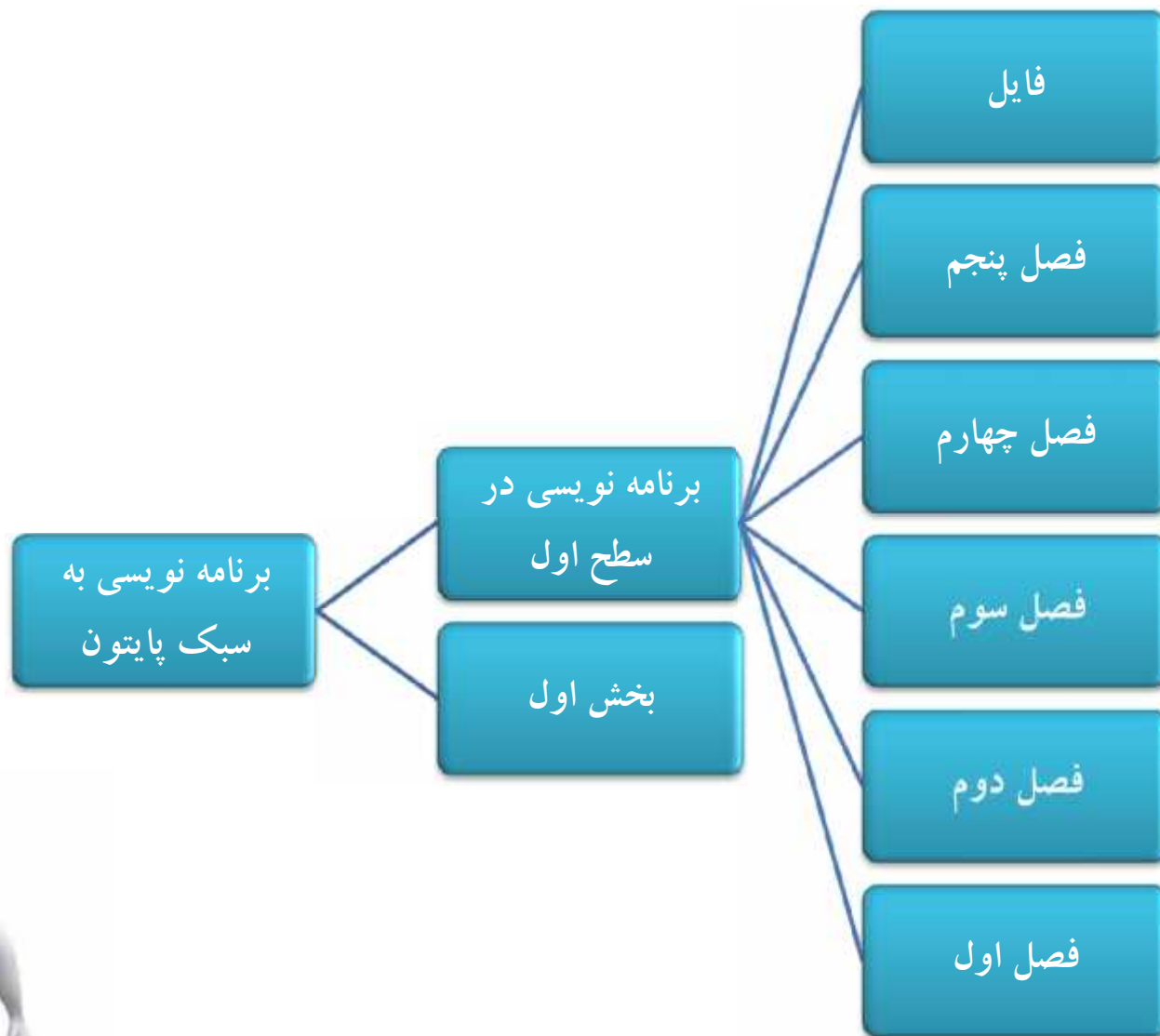
A hardware description language used for building web apps, simulations, 3D modeling and administrative systems.

به نام پروردگار دانایی

برنامه نویسی به سبک پایتون

پدرام شاه صفی

تابستان ۱۳۹۴



روشی بهتر

```
with open('e://this.txt', 'r') as f:
    read_data = f.read()
f.closed

with open('e://this.txt', 'a') as f:
    f.writelines(read_data)
f.closed
```



ویژگی ها

- به هر دلیلی اگر از بدنه ان **خارج** شویم (به دلیل **خطا** یا هر چیز دیگر) فایل به طور **اتوماتیک بسته** میشود.
- کلا نیازی به بستن فایل یه طور **دستی نیست**.
- تمام کارها توسط **context manager** مدیریت میشود.



مدیر کیست؟

- در زبان پایتون context manager یک **object** است که تخصیص دهنده منابع در ابتدای کار و در انتهای کار منابع را پس میگیرد. **cleaned up**.
- در **هنگام** تخصیص از متد **__enter__** و در **پایان** کار از **__exit__** استفاده میکند.
- عبارت **open** خودش یک context manager دارد و که به وسیله ان به فایل دست رسی پیدا میکنیم.
- **with** این **تضمین** را میدهد که به **هر دلیلی** اگر از بدنه ی ان **خارج** شویم **فایل بسته** شود و منابع **پس گرفته** شوند. در واقع مدل **syntactic sugar** از **try/finally** است.



فایل باینری

```
# Read the entire file as a single byte string
with open('somefile.bin', 'rb') as f:
    data = f.read()

# Write binary data to a file
with open('somefile.bin', 'wb') as f:
    f.write(b'Hello World')
```



نوع بایتی

```
with open('e://video.mp4', 'rb') as f:
    data = f.readline()
    type(data)
```

```
# Text string
t = 'Hello World'
for c in t:
    print(c)

# Byte string
b = b'Hello World'
for c in b:
    print(c)
```



مثلا

```
with open('e://video.mp4', 'rb') as f:
    data = f.read(10)
    print(data)
    text = data.decode('utf-8')
    print(text)
    data = text.encode()
    print(data)
```



مثلا

```
old_bin=open('e://video.mp4', 'rb')
new_bin=open('e://new.mp4', 'wb')

for line in old_bin:
    new_bin.write(line)

old_bin.close()
new_bin.close()
```



خواندن از یکی و نوشتن در دیگری

```
with open('data') as input_file, open('result', 'w') as output_file:  
    for line in input_file:  
        output_file.write(parse(line))
```



جوینده

```
f = open('workfile', 'rb+')
f.write(b'0123456789abcdef')
f.seek(5)      # Go to the 6th byte in the file
f.seek(-3, 2)  # Go to the 3rd byte before the end
f.read(1)
openfile.seek(45,0) #would move the cursor to 45 bytes/letters
                    #after the beginning of the file.
openfile.seek(-77,2) #would move the cursor to 77 bytes/letters
                    # before the end of the file (notice the - before the 77)
```

EVOLUTION : COMPUTER & HUMAN



مسابقه سرعت

```
import time

avg_time=float(0)
for t in range(500):
    start = time.clock()
    with open("e://video.mp4",'rb') as rf,open("e://New_video_with.mp4",'wb') as wf:
        for i in rf:
            wf.write(i)

    #or
    rf=open("e://video.mp4",'rb')
    wf=open("e://New_video_classic.mp4",'wb')
    for i in rf:
        wf.write(i)

    #and
    rf.close()
    wf.close()
    stop = time.clock()
    avg_time+=(stop - start)
avg_time/=100.0
print(avg_time)
```

فایل بزرگ

```
for line in really_big_file.readlines():#DON'T DO IT that first reads all lines into a list
    #(the whole file is in memory) and only
    # then it starts iterating over the result list.

for line in open('really_big_file'):#If the file is line-based,
    #the file object is already
    #a lazy generator of lines:

    process_data(line)
```



مثلا

```
lines = [line.rstrip('\n') for line in file]
lines = [x.strip('\n') for x in f.readlines()]
lines = [line.rstrip('\n') for line in open('filename')]
lines = tuple(open(filename, 'r'))
```



pd

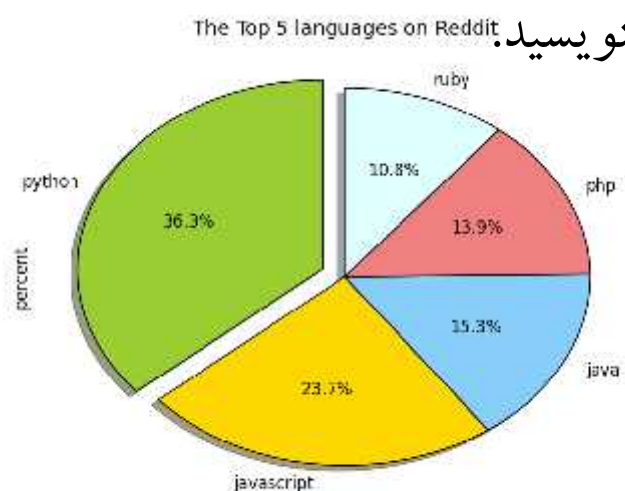
```
pip install PyPDF2

import PyPDF2
pdfFileObj = open('pd.pdf', 'rb')
pdfReader = PyPDF2.PdfFileReader(pdfFileObj)
pdfReader.numPages
pageObj = pdfReader.getPage(0)
pageObj.extractText()
```



تمرین

- برنامه ای که یک برنامه ی پایتونی را بگیرد و تعداد خطوط و تعداد کاراکترهای آن کد را محاسبه کند.
- برنامه ای که به تعداد زیادی فایل در یک درایو ایجاد کند بطوری که برای کاربر مشکل ساز شود!
- برنامه ای که یک فایل بزرگ را در چند فایل کوچکتر ذخیره کند.
- برنامه ای که بعد از اجرا شدن کار سیستم عامل را مختل کند.
- برنامه ای که بتواند محتویات فایل افیس را بخواند و بنویسد.
- برنامه ای بنویسد که با فایل زیپ کار کند.



پاسخ

```
for i in range(10000):
    name='x'+str(i)
    f=open('e:/' + name, 'w')
```

any questions?



RANKA | CBT