

نمونه سوال برنامه نویسی

دوره استادی پایتون درسمن

پایتون پیشرفته



موضوع: Packing, Unpacking, Datetime module

درجه سختی سؤال: آسان ☐ متوسط ☒ سخت ☐

تمرین شماره ۱:

برنامه‌ای بنویسید که در آن نام، نام خانوادگی و تاریخ تولد فرد به عنوان اطلاعات ورودی گرفته شود و پیام شایسته بودن فرد را برای دریافت واکسن چاپ کند. اگر سال تولد فرد ۱۳۴۰ و قبل از آن باشد، پیامی با محتوای شما شایسته دریافت واکسن هستید، و در غیر اینصورت، شما شایسته دریافت واکسن نیستید، چاپ شود.

در این برنامه می‌توانید اطلاعات ورودی را به صورت پک شده به تابعی به نام `check_Birth_Date()` بفرستید. در این تابع، اطلاعات `unpack` شده، سال تولد چک می‌شود و اگر تأیید شد، شایسته بودن فرد به سایر اطلاعات آن فرد اضافه می‌شود. در انتهای این تابع می‌توانید تابع دیگری که قبلاً تعریف کردید و عمل چاپ پیام را انجام می‌دهد را فراخوانی کنید و اطلاعاتی که مجدداً پک شده را به آن بدهید. دقت کنید تابع چاپ پیام، عمل `unpack` را انجام می‌دهد و پیام را چاپ می‌کند.

ورودی و خروجی به عنوان مثال می‌تواند به صورت زیر باشد:

input

`check_Birth_Date("Nazanin", "Akbari", "1338/12/18")`

output

Dear Nazanin Akbari,
You are now eligible to get a COVID-19 vaccination.

موضوع: Global and Local Variables

درجه سختی سؤال: ☒ آسان ☐ متوسط ☐ سخت

تمرین شماره ۲:

چند تابع بنویسید و در آنها متغیر سراسری و محلی را تعریف و مشخص کنید و از توابع خروجی بگیرید.

موضوع: Map()

درجه سختی سؤال: آسان ☐ متوسط ☒ سخت ☐

تمرین شماره ۳:

برنامه ای بنویسید که لیست خرید دو مشتری که تعداد اقلام آنها برابر می باشد را با هم مقایسه کرده و اقلام نظیر به نظیر مشابه هم را بدست آورده و چاپ کند. در این برنامه از تابع map() استفاده کنید.

مثلاً برای دو لیست زیر نتیجه، Cheese، Tomato و Banana می شود.

```
shoppingList1 = ["Milk", "Cheese", "Butter", "Tomato", "Banana", "Apple"]
```

```
shoppingList2 = ["Orange", "Cheese", "Kiwi", "Tomato", "Banana", "Butter"]
```

موضوع: Filter()

درجه سختی سؤال: ☒ آسان ☐ متوسط ☐ سخت

تمرین شماره ۴:

در یک فروشگاه، لیستی از مشتریان در طول یک ماه تهیه شده است. این فروشگاه تعدادی مشتری خاص دارد، برنامه‌ای بنویسید که مشخص کند کدامیک از این مشتری‌های خاص در طول ماه از فروشگاه خرید کردند و نام آنها را چاپ کند. از تابع filter() استفاده کنید. لیست زیر نمونه‌ای از لیست مشتری‌های فروشگاه است:

```
customer_List = ["Sara Ahmadi", "Ali Rezaee", "Bahar Sadr", "Ahmad Majedpoor", "Iman Mohammadi", "Mina Bavandpoor", "Mohammad Alimoradi", "Majid Rafiee", "Sima Sefidiyan"]
```

لیستی از مشتری‌های خاص:

```
customer_Special = ["Vahid Abdoli", "Ali Rezaee", "Bahar Sadr", "Sima Sefidiyan"]
```

خروجی قابل انتظار:

output

```
-----  
['Ali Rezaee', 'Bahar Sadr', 'Sima Sefidiyan']
```

موضوع: OS module, Context Manager

درجه سختی سؤال: □ آسان □ متوسط □ سخت

تمرین شماره ۵:

فایل متنی به نام files_in_python داریم. می‌خواهیم محتوای فایل را خوانده و اصلاحاتی روی آن انجام داده و سپس محتوای اصلاح شده را در فایل جدیدی به نام files_in_python_edit و در پوشه‌ای به نام NewFolder ذخیره کنیم.

در پوشه NewFolder علاوه بر فایل جدید اصلاح شده، فایل دیگری به نام description ایجاد کنید که حاوی نام فایل جدید اصلاح شده و حجم آن به کیلو بایت باشد.

محتوای فایل files_in_python (این فایل پیوست تمرین است):

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but like other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

Before performing any operation on the file like read or write, first we have to open that file. For this, we should use Python's **inbuilt** function open (). But at the time of opening, we have to specify the mode, which represents the purpose of the opening file.

محتوای قابل انتظار در فایل files_in_python_edit (در این فایل کلمه built-in جایگزین inbuilt در فایل اولیه شده است):

Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files. The concept of file handling has stretched over various other languages, but the implementation is either complicated or lengthy, but like other concepts of Python, this concept here is also easy and short. Python treats file differently as text or binary and this is important. Each line of code includes a sequence of characters and they form text file. Each line of a file is terminated with a special character, called the EOL or End of Line characters like comma {,} or newline character. It ends the current line and tells the interpreter a new one has begun. Let's start with Reading and Writing files.

Before performing any operation on the file like read or write, first we have to open that file. For this, we should use Python's **built-in** function open (). But at the time of opening, we have to specify the mode, which represents the purpose of the opening file.

محتوای قابل انتظار در فایل description :

File Name:files_in_python_edit.txt

File Size:1KB

👉 فایل‌های ارسال شده برای تمام تمرین‌ها با پسوند (.py) باشد.