

Projeto ESOF #Sprint 01 Python 2017-04-11 2017-04-25

1ª etapa realizada com o intuito de iniciar o contato com programação em Python.

Objetivos pré-determinados

- Realizar o curso de Python disponível no site Codecademy;
 - Entender a sintaxe e funcionamento de listas e dicionários.
- Utilizar o livro Automating the Boring Stuff with Python para aprofundar os conhecimentos adquiridos;
 - Capítulos 7 (*Pattern Matching*) e 8 (*File Operating*).
 - Escrever códigos para alguns dos projetos sugeridos nos capítulos.

Desenvolvimento

Codecademy

A atividade iniciou-se com o curso de Python disponível no site Codecademy, no qual foi possível ter um contato inicial com as seguintes características da linguagem:

- Sintaxe;
- Operação com strings;
 - Funções e métodos próprios.
- Uso de condicionais e controle de fluxo;
 - **if**, **else**, **elif**.
- Funções;
 - **def**.
- Estruturas de repetição;
 - **for**, **while**.
- Operações **bitwise**;
- Classes;
- Operação de arquivos.

Automating de Boring Stuff with Python

Após o contato inicial proporcionado pelo curso, a segunda parte envolveu o uso do livro *Automating the Boring Stuff with Python*.

1. Capítulo 7 - *Pattern Matching with Regular Expressions* De forma geral, o capítulo 7 baseia-se no módulo **re** para realização de buscas baseadas numa string padrão pré-determinada. Alguns exemplos dos métodos utilizados são:

Método	Uso
<code>.compile()</code>	Transforma uma string em um regex , que é utilizado para as buscas.
<code>.search()</code>	Utiliza um 'regex' já existente para encontrar sua primeira ocorrência em uma string.
<code>.findall()</code>	Utiliza um 'regex' já existente para encontrar todas as suas ocorrências em uma string.

Após o estudo do assunto, foi realizado o projeto sugerido ***Strong Password Detection***, que envolveu escrever o código para uma função que determinasse se uma dada senha era considerada "forte" (Ao menos 8 caracteres, um caixa alta, um caixa baixa e um dígito). O código-fonte encontra-se abaixo:

```
import re

def StrongPassword (password):

    if len(password)<8:
        return False

    passwordRegexLower=re.compile(r'[a-z]')
    passwordRegexUpper=re.compile(r'[A-Z]')
    passwordRegexDigit=re.compile(r'[0-9]')

    moForPassword=passwordRegexLower.search(password)
    if moForPassword==None:
        return False

    moForPassword=passwordRegexDigit.search(password)
    if moForPassword==None:
        return False

    moForPassword=passwordRegexUpper.search(password)
    if moForPassword==None:
        return False
```

```

        return True

userPass=input("Type the password: ")

if StrongPassword(userPass):
    print("Valid password.\n")
else:
    print("Password not strong enough.\n")

```

2. Capítulo 8 - *Reading and Writing Files* O capítulo 8 envolveu o estudo de funções e métodos do módulo **os**, além de **built-ins**, utilizados para a operação de arquivos e diretórios, como criá-los, ler, escrever, copiar, etc, sendo alguns assuntos já vistos no curso do Codecademy. Exemplos de algumas funções e métodos utilizados:

Função ou Método	Uso
<code>.getcwd()</code>	Retorna o caminho do diretório corrente.
<code>.listdir()</code>	Retorna uma lista com os nomes de todos os diretórios e arquivos presentes no diretório corrente.
<code>open()</code>	Abre um buffer para um arquivo disponível no sistema ou cria um novo.
<code>.close()</code>	Fecha o buffer previamente aberto.
<code>.read()</code>	Faz a leitura do conteúdo do arquivo, sendo possível armazená-lo em uma variável, por exemplo.

Deste capítulo, foi realizado o projeto sugerido "Regex Search", que envolveu escrever o código para um programa que busca em todos os arquivos presentes no diretório corrente linhas que contenham uma expressão determinada pelo usuário e imprime o que foi encontrado, fazendo-se uso do que foi visto no capítulo anterior. O código-fonte encontra-se abaixo:

```

import re
import os

fileList=os.listdir()
userRegex=input("Enter the expression to be searched: ")

# creates the regex that searches for the line that contains the user expression
searchRegex=re.compile('(.*)?(%s)(.*)?' % userRegex)
stringList=[]
textInFile=""

```

```

for i in range(len(fileList)):
    if os.path.isfile(fileList[i]):
fileOP=open(fileList[i], 'r')
textInFile=fileOP.read()

# searches for all the lines and stores them in stringList
stringList=searchRegex.findall(textInFile)

print("In file %s:" % fileList[i])
if stringList==[]:
    print("  No text found.")
else:
    for j in range(len(stringList)):
print('  '+'.join((stringList[j])).strip()) # formatting of the lines found

print("\n")

fileOP.close()

```

Considerações finais

O desenvolvimento dessa atividade possibilitou uma introdução eficiente à programação em Python, com o estudo de funções, módulos e métodos que serão extremamente úteis e até indispensáveis para a realização de projetos futuros. As dificuldades encontradas envolvem o fato de que existe uma quantidade expressiva de comandos a serem utilizados, sendo necessário consultas constantes a definições e exemplos de uso, algo que, sem dúvida, será superado ao longo do tempo, adquirindo-se costume de programar nessa linguagem.