

In [1]:

```
###IMPORT DAS BIBLIOTECAS QUE SERÃO USADAS

import requests
import json
import hashlib
```

In [2]:

```
###REQUISIÇÃO HTTP

response = requests.get('https://api.codenation.dev/v1/challenge/dev-ps/generate-data?token=2a7971a2f9916754246c74daca9c608f53247c2')
```

In [3]:

```
###VERIFICAÇÃO DO QUE TEM DENTRO DA REQUISIÇÃO

response.content
```

Out[3]:

```
b'{"numero_casas":7,"token":"2a7971a2f9916754246c74daca9c608f53247c2","cifrado":"wshu av aoyvd vul (ptwsltluahapvu) hdhf; fvb dpss, huf ovd. mylk iyvvrz","decifrado":"","resumo_criptografico":""}\n'
```

In [4]:

```
###SALVANDO O ARQUIVO OBTIDO

a = response.json()

with open('resp_text.json', 'w') as jsonFile:
    json.dump(a, jsonFile)
```

In [5]:

```
###CARREGAMENTO DO ARQUIVO

with open('resp_text.json', 'r') as jsonFile:
    answer = json.load(jsonFile)
```

In [6]:

```

###CÓDIGO PARA OBTER O TEXTO DECIFRADO

class Caesar:
    def __init__(self):
        self.__letters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'

    def decrypt(self, texto_cifrado, key = a['numero_casas']):
        ''' (Caesar, str, int) -> str

        Retorna em texto plano o texto decifrado com a cifra de Cesar, utilizando a chave
        '''
        plain_text = ''
        texto_cifrado = texto_cifrado.upper()
        for ch in texto_cifrado:
            if ch in self.__letters:
                idx = self.__letters.find(ch) - key
                plain_text += self.__letters[idx]
            else:
                idx = ch
                plain_text += idx
        return plain_text.lower()

```

In [7]:

```

###OBTENÇÃO DO TEXTO DECIFRADO

decrypt = Caesar().decrypt(a['cifrado'])
print(decrypt)

```

plan to throw one (implementation) away; you will, anyhow. fred brooks

In [8]:

```

###ATUALIZAÇÃO DO JSON

answer['decifrado'] = decrypt

```

In [9]:

```

###CONFERINDO A ATUALIZAÇÃO

answer

```

Out[9]:

```

{'numero_casas': 7,
 'token': '2a7971a2f9916754246c74daca9c608f53247c2',
 'cifrado': 'wshu av aoyvd vul (ptwsltluahapvu) hdhf; fvb dpss, hufo vd. mylk iyvvrz',
 'decifrado': 'plan to throw one (implementation) away; you will, anyhow. fred brooks',
 'resumo_criptografico': ''}

```

In [10]:

```
###CÓDIGO PARA OBTENÇÃO DO RESUMO ATRAVÉS DO SHA1

import hashlib
hash_object = hashlib.shal(b'plan to throw one (implementation) away; you will,
    anyhow. fred brooks')
```

In [11]:

```
###OBTENÇÃO DO RESUMO

resumo = hash_object.hexdigest()
print(hash_object.hexdigest())

2dab381e8f3df45c68a6fe7b59254267c439e842
```

In [12]:

```
###ATUALIZAÇÃO DO JSON

answer['resumo_criptografico'] = resumo
```

In [13]:

```
###CONFERINDO A ATUALIZAÇÃO

answer
```

Out[13]:

```
{'numero_casas': 7,
 'token': '2a7971a2f9916754246c74daca9c608f53247c2',
 'cifrado': 'wshu av aoyvd vul (ptwsltluahapvu) hdhf; fvb dpss, hufo
vd. mylk iyvvrz',
 'decifrado': 'plan to throw one (implementation) away; you will, an
yhow. fred brooks',
 'resumo_criptografico': '2dab381e8f3df45c68a6fe7b59254267c439e842'}
```

In [14]:

```
###SALVANDO O CONTEÚDO FINAL EM UM JSON

with open('answer.json', 'w') as jsonFile:
    json.dump(answer, jsonFile)
```

In [15]:

```
###URL DA API PARA A SUBMISSÃO DO ARQUIVO ATUALIZADO

url = 'https://api.codenation.dev/v1/challenge/dev-ps/submit-solution?token='+an
swer['token']
```

In [16]:

```
###SETANDO OS PARAMETROS EM FILES

files = {
    'answer': ('answer', open('answer.json', 'rb'), 'application/json'),
    'Content-Type': 'multipart/form-data; boundary=-----665956835435877517162732'
}

###OBS: BOUNDARY OBTIDO ATRAVÉS DO PROGRAMA POSTMAN
```

In [17]:

```
###SUBMETENDO O ARQUIVO ATUALIZADO VIA POST PARA A API

r = requests.post(url, files=files)
```

In [18]:

```
###RESULTADO DA SUBMISSÃO

if r.status_code == 200:
    print('Parabéns! A requisição foi um sucesso :')
    if r.text[-5:-2] == '100':
        print('O seu score foi: '+r.text[9:12]+'%')
    else:
        print('O seu score foi: '+r.text[9:11]+'%')
elif r.status_code == 429:
    print('Espere um pouco antes de enviar uma nova tentativa!')
else:
    print('Cuidado! Existe algum erro no código enviado :(')
```

```
Parabéns! A requisição foi um sucesso :)
O seu score foi: 100%
```