Folha de Conclusões: Extraindo Dados de Recursos Online

Prática

```
# Extrair informação de uma página da web usando um URL
import requests

req = requests.get(URL)
print(req.text) # imprimindo o conteúdo da página
print(req.status_code) # imprimindo o código de status
```

```
# Pesquise uma string pela primeira substring que corresponda a uma expressão regular

import re
print(re.search(pattern, string).group())
```

```
# Divida uma string em substrings por ocorrências do padrão
# maxsplit - número máximo de divisões, maxsplit=0 por padrão

import re
print(re.split(pattern, string, maxsplit=num_split))
```

```
# Encontre uma substring e a substitua pela substring repl
import re
print(re.sub(pattern, repl, string))
```

```
# Encontra todas as substrings correspondentes

import re
print(re.findall(pattern, string))
```

```
# Gera uma estrutura de árvore para uma página da web
```

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(req.text, parser)
```

```
# Encontre a primeira tag 'tag'
# Retorna uma string com tag, atributos e conteúdo
# attrs - dicionário de atributos de tags

tag_content = soup.find(tag, attrs={"attr_name": "attr_value"})
print(tag_content.text) # conteúdo sem tag
```

```
# Operações com todas as tags
# attrs - dicionário de atributos de tags

for tag_content in soup.find_all(tag, attrs={"attr_name": "attr_value"}):
  # faça alguma coisa
```

Teoria

Mineração da Web — o processo de procurar recursos online e extrair dados deles

HTML — Hypertext Markup Language, uma linguagem usada para criar páginas da web

HTTP — protocolo de transferência usado para transmitir informação online

HTTPS — uma versão segura do HTTP

Tag HTML — um elemento de linguagem de marcação de hipertexto

Atributo de tag — um recurso que permite o ajuste fino ao trabalhar com tags