ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Python WWW API

Python WWW API

Nas duas aulas anteriores vimos:

- **Fundamentos da Web, incluindo URL, HTTP e HTML**
- **Intercâmbio de dados por meio de padronizações (HTML e JSON)**

Nesta aula, veremos alguns módulos da biblioteca padrão Python para acessar e processar recursos da Web.

Módulo urllib.request

O módulo urllib.request permite requisitar e receber recursos da Web, de modo similar a um navegador.

A função urlopen():

- recebe como parâmetro uma URL
- formula uma requisição HTTP que será enviada ao servidor especificado na URL
- obtém e retorna uma resposta HTTP completa do servidor.

Módulo urllib.request

Exemplo:

```
from urllib.request import urlopen

def getSource(url):
    response = urlopen(url)
    html = response.read()
    return html.decode()

html = getSource('http://www.uol.com.br')
html
```

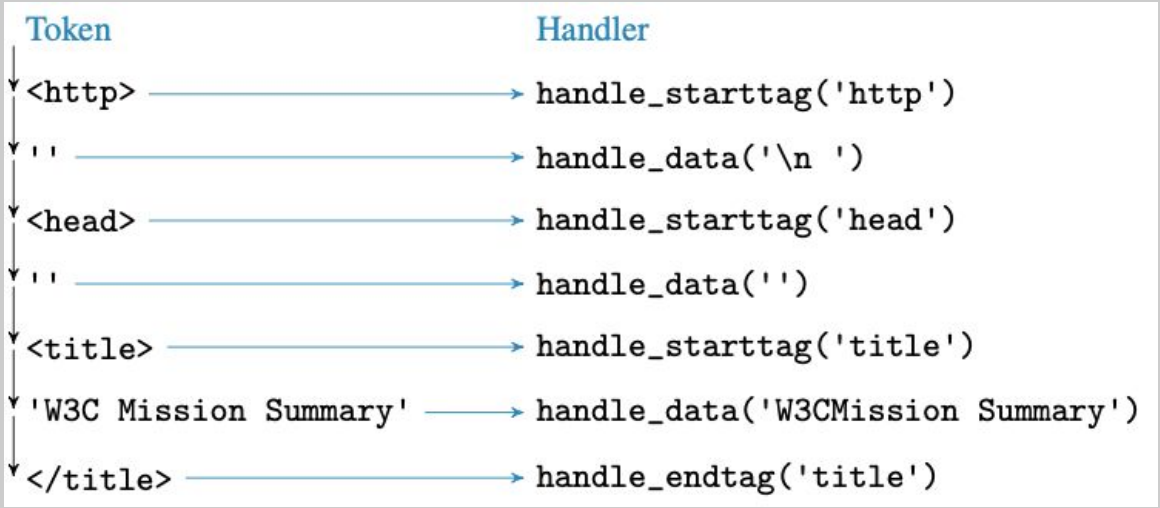
```
'<!DOCTYPE html> <html> <head lang="pt-br"> <link rel="prelo
ad" href="https://c.jsuol.com.br/assets/?loadComponent=media
&contentType=js&tpl=assets/dist/libs/duckslake-sdk.min" as
="script" crossorigin="anonymous"> <link rel="dns-prefetch"
href="//conteudo.jsuol.com.br"> <link rel="dns-prefetch" hre
f="//stc.uol.com"> <link rel="dns-prefetch" href="//conteud
o.imguol.com.br"> <link rel="dns-prefetch" href="//conteudo.
imguol.com.br"> <link rel="dns-prefetch" href="//conteudo.im
guol.com.br"> <link rel="dns-prefetch" href="//e.imguol.co
m"> <link rel="dns-prefetch" href="//click.uol.com.br"> <lin
k rel="dns-prefetch" href="//smetrics.uol.com.br"> <link rel
="dns-prefetch" href="//securepubads.g.doubleclick.net"> <li
nk rel="dns-prefetch" href="//www.googletagservices.com"> <l
ink rel="dns-prefetch" href="//tt-10162-1.seg.t.tailtarget.c
om"> <link rel="dns-prefetch" href="//tm.uol.com.br"> <link
```

Módulo html.parser

O módulo html.parser, por meio da classe HTMLParser, permite processar elementos HTML de uma página Web.

O método feed() da classe HTMLParser recebe como entrada uma página HTML no formato string, e para cada 'token' lido (tags de início, tags de fim, texto, etc.), executa um handler correspondente.

Módulo html.parser



Fonte: Perkovic, 2015

Módulo html.parser

Inicialmente, cada handler é implementado para não fazer nada. Assim, executamos o código abaixo, o qual não gera nenhuma saída visível:

```
from html.parser import HTMLParser

parser = HTMLParser()
parser.feed(html)
```

Para trazer funcionalidade a um determinado handler, precisamos sobrescrever o método correspondente, estendendo a classe HTMLParser.

Módulo html.parser

Exemplo:

```
from html.parser import HTMLParser

class MyParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        if tag == 'a':
            for attr in attrs:
                if attr[0] == 'href':
                    print(attr[1])
```

```
from urllib.request import urlopen

def getSource(url):
    response = urlopen(url)
    html = response.read()
    return html.decode()

html = getSource('http://www.uol.com.br')
```

```
parser = MyParser()
parser.feed(html)
```

```
#versao-mobile
http://batepapo.uol.com.br/
http://clicklogger.rm.uol.com.br/?prd=16&grp=src:210;chn:0;creative:barrauol;thm:barrauol-host&msr=Cliques%20de%20Origem:1&oper=11&redir=http://www.uolhost.com.br/
https://www.portaleducacao.com.br/portal-play?utm_source=uol&utm_medium=menu&utm_campaign=portalplay&utm_term=menu
https://play.uol.com.br/?utm_source=uol.com.br&utm_medium=barrauol&utm_campaign=linkfixo_barrauol&utm_term=barrauol-uolplay&utm_content=barrauol
http://clicklogger.rm.uol.com.br/?prd=32&grp=src:210;chn:0;creative:barrauol;thm:barrauol-pagseguro&msr=Cliques%20de%20Origem:1&oper=11&redir=https://pagseguro.uol.com.br/
https://email.uol.com.br/
https://sac.uol.com.br/#/
https://www.uol.com.br/
https://economia.uol.com.br/cotacoes/cambio/
https://economia.uol.com.br/cotacoes/cambio/euro-uniao-europeia/
https://clicklogger.rm.uol.com.br/?prd=11&grp=src:13;chn:539;cpg:link
```

Outro exemplo

Retornar o número de polos de um determinado curso da UNIVESP.

```
from html.parser import HTMLParser
from urllib.request import urlopen, Request

class MyParser(HTMLParser):
    def __init__(self):
        HTMLParser.__init__(self)
        self.n_polos = 0

    def handle_starttag(self, tag, attrs):
        if tag == 'p':
            for attr in attrs:
                if attr[0] == 'class' and attr[1] == 'item-polos':
                    self.n_polos += 1

    def num_polos(self):
        return self.n_polos
```

Outro exemplo

Retornar o número de polos de um determinado curso da UNIVESP.

```
def getSource(url):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 6.1)
        AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0
Safari/537.3'}
    reg_url = "https:XXXXOOOO"
    req = Request(url=url, headers=headers)
    html = urlopen(req).read()
    return html.decode()

html = getSource('https://univesp.br/cursos/engenharia-de-computacao')

parser = MyParser()
parser.feed(html)
parser.num_polos()
```

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Python WWW API