



Revisão Strings

fmasanori@gmail.com

Texto

- É difícil comunicar-se sem palavras
- Entre os vários tipos de dados, um dos mais importantes é o texto ou string
 - Obs.: não é tão fácil manipular strings em algumas linguagens
- Vamos procurar onde estão as informações num texto == scraping
- E iremos aprender um dos conceitos mais importantes de orientação à objetos: métodos

Starbuzz Café

Meu programador
sumiu! Você pode me
ajudar? Ele deixou o
seguinte código...



The Starbuzz CEO →

Código Starbuzz atual

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices.html')
texto = pagina.read().decode('utf8')
print (texto)
```

```
>>>
<html><head><title>Welcome to the Beans'R'Us Pricing Page</title>
<link rel="stylesheet" type="text/css" href="beansrus.css" />
</head><body>
<h2>Welcome to the Beans'R'Us Pricing Page</h2>
<p>Current price of coffee beans = <strong>$5.31</strong></p>
<p>Price valid for 15 minutes from Tue Mar 15 13:16:01 2011.</p>
</body></html>
```

O CEO quer apenas o preço



Você acha que
pode obter
apenas o preço?

O preço está embutido no HTML

- Este é um texto HTML “bruto”, que é o formato das páginas Web
- O preço está embutido no HTML

>>>

```
<html><head><title>Welcome to the Beans'R'Us Pricing Page</title>  
<link rel="stylesheet" type="text/css" href="beansrus.css" />  
</head><body>  
<h2>Welcome to the Beans'R'Us Pricing Page</h2>  
<p>Current price of coffee beans = <strong>$5.31</strong></p>  
<p>Price valid for 15 minutes from Tue Mar 15 13:16:01 2011.</p>  
</body></html>
```



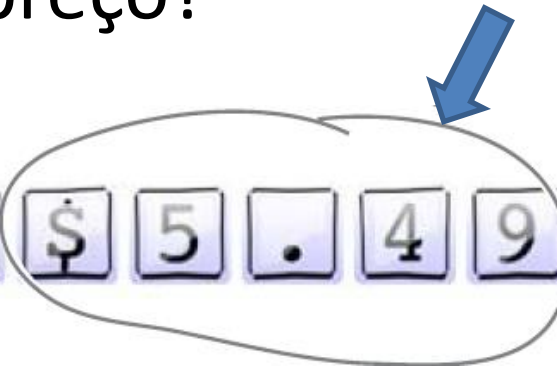
Strings

- Strings são seqüências de caracteres

< h t m l > < h e a d > < t i

- Como obter apenas o preço?

< s t r o n g > \$ 5 . 4 9 < /



Strings

Deslocamento

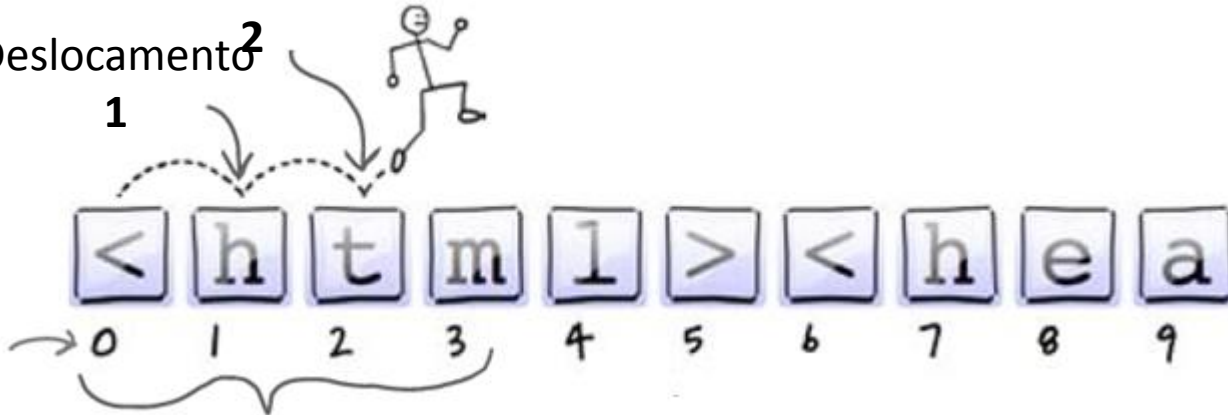
zero



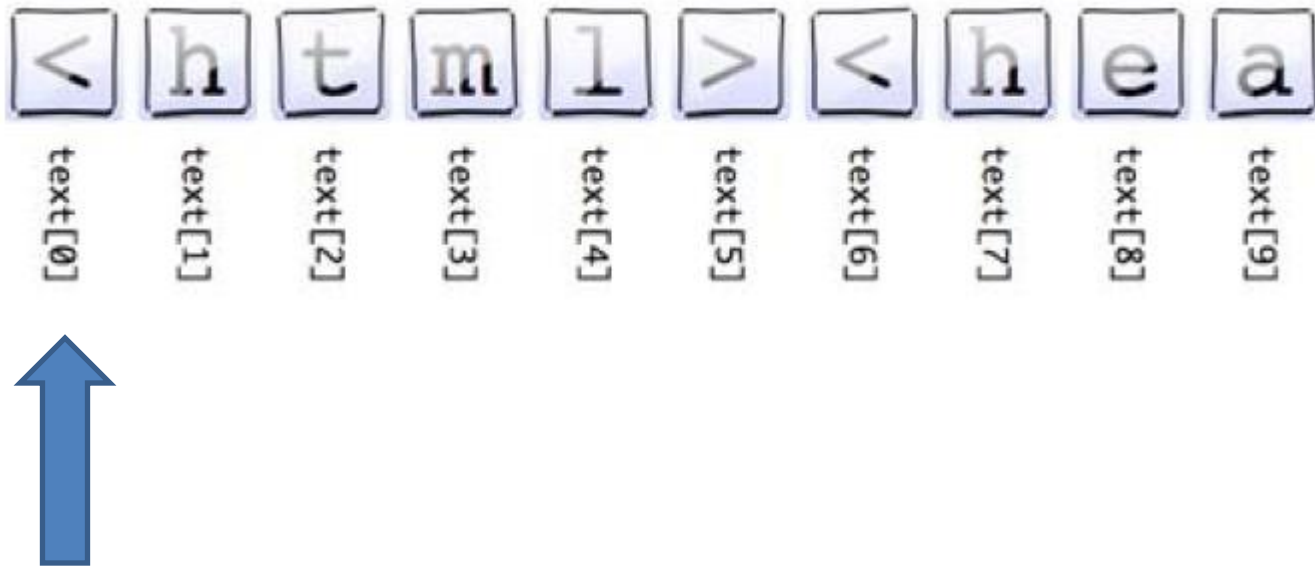
Deslocamento

Deslocamento²

1



Strings



O primeiro é zero!!

Fatiamento

```
>>> time = "Palmeiras"
>>> time[2:5]
'lme'
>>> time[0:3]
'Pal'
>>> time[4:6]
'ei'
>>>
```

"Palmeiras"
0 1 2 3 4 5 6 7 8

Fatia do primeiro número até antes do segundo
Não inclui o segundo número!

Fatiamento



Fatiamento

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices.html')
texto = pagina.read().decode('utf8')
preço = texto[234:238]
print (preço)

>>>
5.31
```

O CEO está feliz!

Exatamente o que preciso!
Você não sabe quanto
tempo e dinheiro irá
economizar para mim...



Não existem perguntas idiotas

- Posso colocar qualquer página web neste código?
 - Sim. Fique à vontade, mas não esqueça o decode
 - Por exemplo, o site abaixo usa iso8859
 - <http://www.ime.usp.br/~pf/algoritmos/dicios/br>
- O que urllib.request faz?
 - Permite conversar com a internet
- Posso acessar uma página diretamente no navegador?
 - Sim. Digite no modo interativo “import antigravity”

Descontos para clientes fiéis

Vocês poderiam ver o preço no programa de fidelidade? Acho que é simples mudar...

Clientes normais

↪ <http://www.beans-r-us.biz/prices.html>

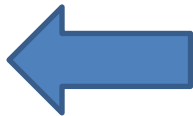
Clientes fiéis ↪ <http://www.beans-r-us.biz/prices-loyalty.html>



Programa de fidelidade

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
texto = pagina.read().decode('utf8')
preço = texto[234:238]
print (preço)
```

```
bean
>>>
```



Não deu certo! Apareceu “bean” no lugar do preço. Por quê será?

O preço se moveu

- As páginas são diferentes e o preço muda de posição na string

Welcome to the Beans'R'Us Pricing Page

Current price of coffee beans = **\$6.94**

Price valid for 15 minutes from Tue Mar 15 13:22:01 2011.

Welcome to the Beans'R'Us Pricing Page

Special Offer!!! Current price of coffee beans = **\$4.39**

Limited time only - get 'em while they're roasting!

Price valid for 15 minutes from Tue Mar 15 13:22:01 2011.

Os dados do Python são espertos

- As linguagens de programação fornecem uma **funcionalidade embutida** nos dados para ajudá-lo
- Os dados do Python são espertos: eles podem **fazer coisas**

```
>>> 'batatinha quando nasce'.upper()  
'BATATINHA QUANDO NASCE'  
>>> 'batatinha quando nasce'.split()  
['batatinha', 'quando', 'nasce']
```

Método find

- Métodos find para strings

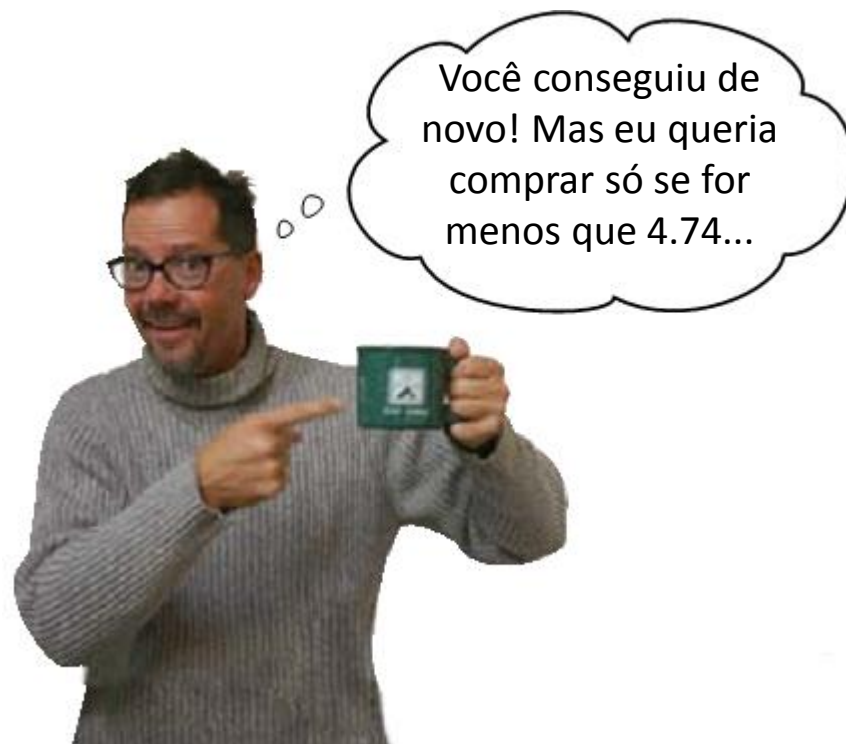
```
>>> "Palmeiras".find("P")
0
>>> "Palmeiras".find("lmei")
2
>>> "Palmeiras".find("Pa")
0
>>>
```

Para saber os métodos que possuo dar ctrl + espaço após ponto

Método find

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
texto = pagina.read().decode('utf8')
onde = texto.find('>$')
inicio = onde + 2
fim = inicio + 4
preço = texto[inicio:fim]
print (preço)
```

```
>>>
4.90
```



Só quando for menos que 4.74



Só quando for menos que 4.74

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
texto = pagina.read().decode('utf8')
onde = texto.find('>$')
inicio = onde + 2
fim = inicio + 4
preço = texto[inicio:fim]
if preço < 4.74:
    print (preço)

>>>
Traceback (most recent call last):
  File "C:/Python31/caneca05.py", line 9, in <module>
    if preço < 4.74:
TypeError: unorderable types: str() < float()
```

Strings são diferentes de números



Convertendo para float

```
import urllib.request
pagina = urllib.request.urlopen(
    'http://beans.itcarlow.ie/prices-loyalty.html')
texto = pagina.read().decode('utf8')
onde = texto.find('>$')
inicio = onde + 2
fim = inicio + 4
preço = float(texto[inicio:fim]) ←
if preço < 4.74:
    print ('Comprar! Preço: %5.2f' %preço)

>>>
Comprar! Preço:  4.58
```


Ele pode ficar testando o preço?



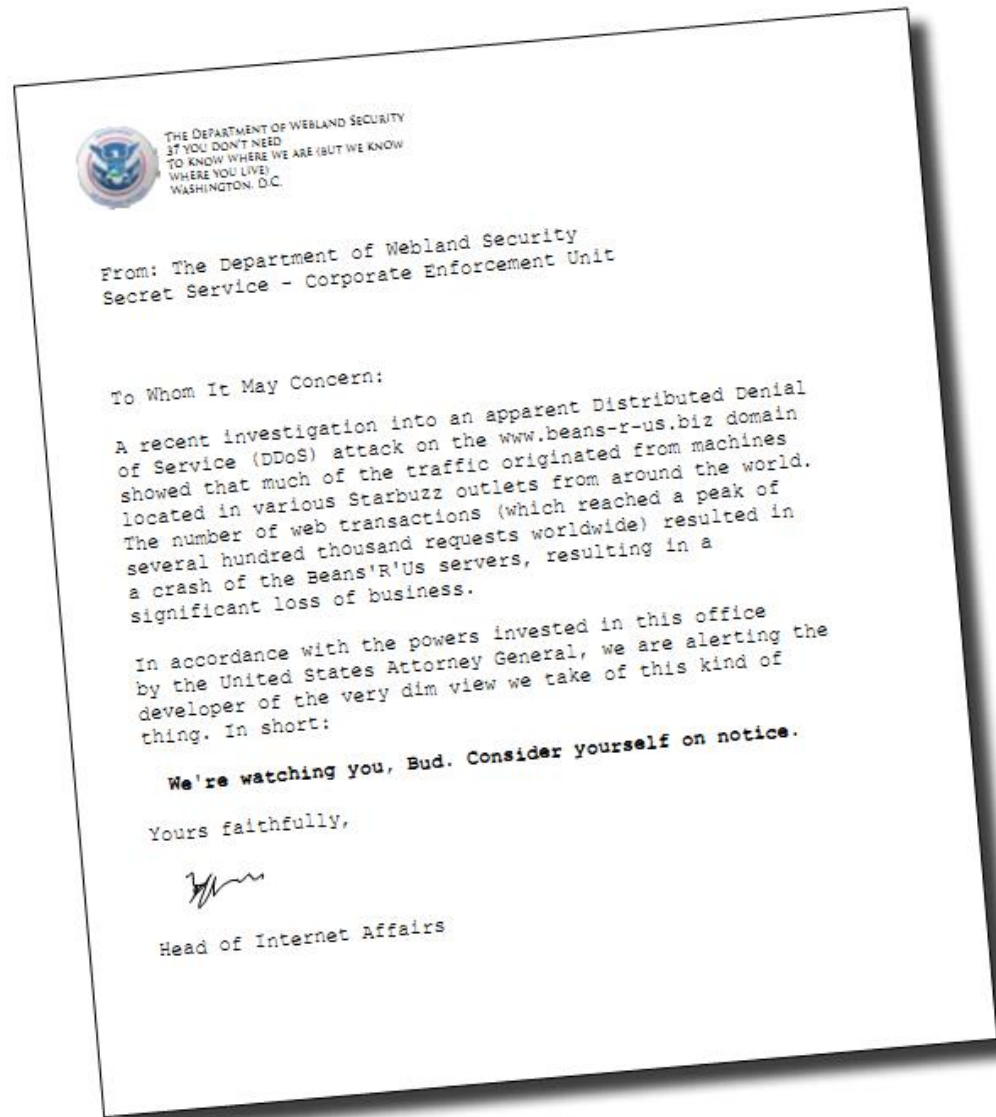
Ele pode ficar tentando?

```
import urllib.request
preço = 99.99
while preço >= 4.74:
    pagina = urllib.request.urlopen(
        'http://beans.itcarlow.ie/prices-loyalty.html')
    texto = pagina.read().decode('utf8')
    onde = texto.find('>$')
    inicio = onde + 2
    fim = inicio + 4
    preço = float(texto[inicio:fim])
print ('Comprar! Preço: %5.2f' %preço)
>>>
Comprar! Preço: 4.47
```

O CEO está muito feliz!



Aconteceu algum problema

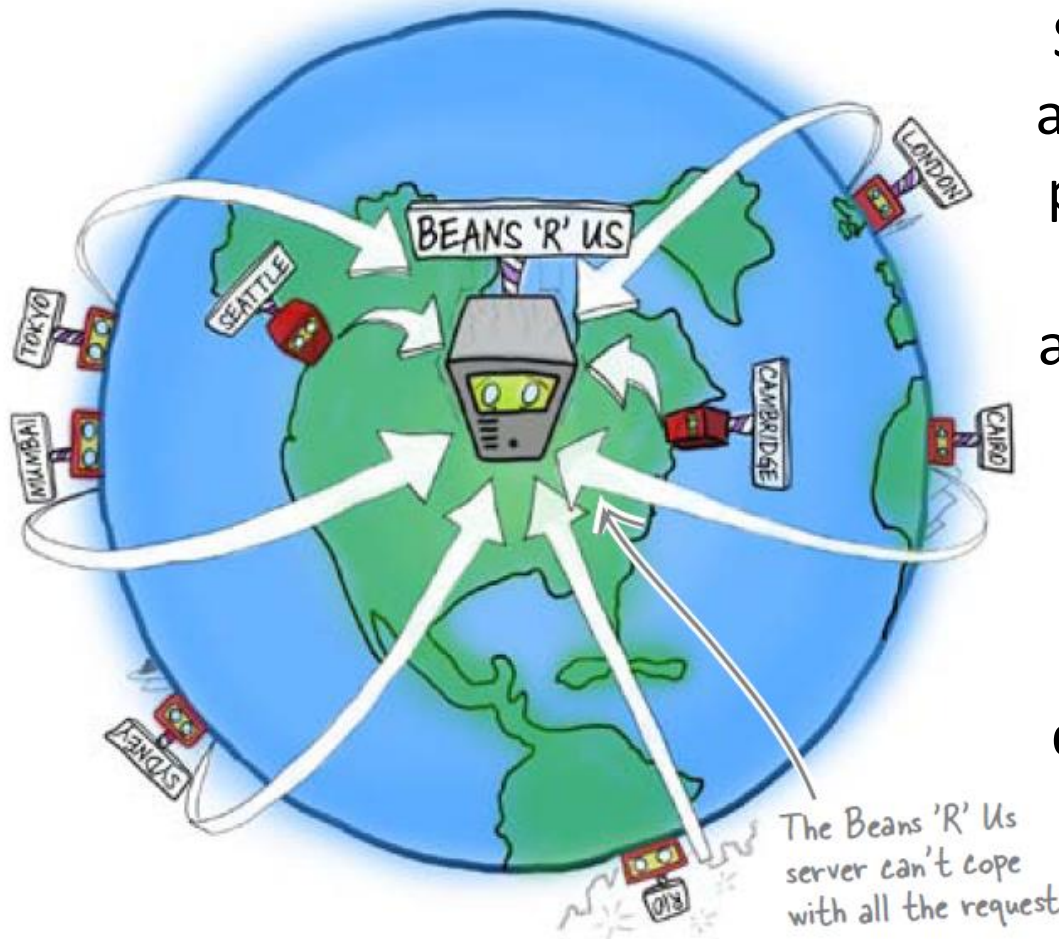


O servidor da
empresa de grãos
travou!

E fomos acusados
de ser hacker's!

Acusação de DDoS

- DDoS – Distributed Denial of Service



Se o valor está acima de 4.74 o programa NÃO espera e já acessa o site de novo!

Seu programa gera milhares de solicitações por hora em todas as filiais

Recebemos uma mensagem

Zkzzkkvkk... Desculpe, cara... Vvzzz...
Muita neve.... Ffzzkk... A ligação....
Pzzzkkvkk... Acho que você precisa....
Vzzzkkk.... da biblioteca time!



Biblioteca time

- Hora atual em segundos `time.clock()`
- Estou no horário de verão? `time.daylight()`
- Dormir alguns segundos `time.sleep(secs)`
- Fuso horário `time.timezone()`

10 minutos entre cada acesso

```
import urllib.request
import time
preço = 99.99
while preço >= 4.74:
    pagina = urllib.request.urlopen(
        'http://beans.itcarlow.ie/prices-loyalty.html')
    texto = pagina.read().decode('utf8')
    onde = texto.find('>$')
    inicio = onde + 2
    fim = inicio + 4
    preço = float(texto[inicio:fim])
    if preço >= 4.74:
        time.sleep(600)
print ('Comprar! Preço: %5.2f' %preço)
>>>
Comprar! Preço: 4.44
```


Resumo

- Strings são seqüências de caracteres
- Acessamos os caracteres individuais pelo índice, que começa com zero
- Métodos são funções embutidas nas variáveis
- Existem bibliotecas de programação com código pronto
- Os dados possuem um tipo, como int ou string

Ferramentas Python

- `texto[4]` acessa o 5º caracter
- `texto[4:9]` acessa do 5º ao 9º caracter
- O método `texto.find()` procura um substring
- `float()` converte algo para ponto flutuante
- Bibliotecas: `urllib.request` e `time`