



Criando cursos

Transcrição

Agora temos um dataframe que, além dos nomes dos alunos e alunas, possui também uma identificação única (ID) e um e-mail (que foi criado juntando os nomes dos alunos a domínios fictícios que criamos e selecionamos aleatoriamente). Porém, ainda falta cadastrarmos uma lista de cursos, afinal estamos trabalhando com uma escola de programação. De modo a mantermos a organização do projeto, criaremos uma nova seção com o título "Criando a tabela Cursos".

Para criarmos nosso dataframe de nomes, utilizamos a função `read_json()` do Pandas, que leu uma API do IBGE e nos trouxe os dados. Pensando nisso, poderíamos buscar uma página na internet com uma tabela semelhante, mas referente a uma lista de cursos.

De modo a facilitar o seu trabalho, nós da Alura preparamos essa tabela para você, e ela pode ser encontrada em <http://tabela-cursos.herokuapp.com/> (<http://tabela-cursos.herokuapp.com/>).

No primeiro acesso, a página pode levar algum tempo para carregar!

O link lhe dará acesso a um site fictício contendo somente a lista de cursos (os links "Agenda, "Quem somos?" e "Contato" não funcionam!).

Nome do curso

Lógica de programação

Java para Web

C# para Web

Ruby on Rails

Cursos de Python

PHP com MySql

A ideia agora é criarmos em nosso projeto um dataframe que utilize os nomes dessa lista. Para isso, precisaremos instalar duas bibliotecas com o auxílio do `!pip3 install :a html5lib e a lxml`.

```
!pip3 install html5lib
```

```
!pip3 install lxml
```

COPIAR CÓDIGO

Em seguida, importaremos a `html5lib`.

```
import html5lib
```

COPIAR CÓDIGO

Salvaremos o endereço <http://tabela-cursos.herokuapp.com/index.html> (<http://tabela-cursos.herokuapp.com/index.html>) em uma variável `url`, que leremos utilizando a função `pd.read_html()`, que nos permitirá carregar a tabela da página HTML. Armazenaremos o seu resultado em uma variável `cursos`.

```
url = 'http://tabela-cursos.herokuapp.com/index.html'  
cursos = pd.read_html(url)  
cursos
```

COPIAR CÓDIGO

[Nome do curso 0 Lógica de programação 1 Java para Web 2 C# para Web 3 Ruby on Rails 4 Cursos de Python 5 PHP com MySql 6 .NET para web 7 Novas integrações com Java 8 TDD com Java 9 Código limpo com C# 10 Preparatório para certificação Java 11 Hardware básico 12 Persistência com .NET 13 Desenvolvendo jogos 14 Análise de dados 15 Estatística básica 16 Internet das coisas 17 Programação funcional 18 Boas práticas em Java 19 Orientação objetos com Java]

Nosso retorno é uma lista, como podemos verificar utilizando a função `type(cursos)` . Se utilizarmos a instrução `cursos[0]` , teremos como retorno um tipo bem parecido com um dataframe, que armazenaremos na variável `cursos` sobrescrevendo o conteúdo anterior.

```
cursos = cursos[0]
```

[COPIAR CÓDIGO](#)

Ao verificarmos seu tipo com `type(cursos)` , veremos que nosso conteúdo é realmente um dataframe.

```
type(cursos)
```

[COPIAR CÓDIGO](#)

```
pandas.core.frame.DataFrame
```

```
cursos
```

[COPIAR CÓDIGO](#)

Nome do curso	
0	Lógica de programação
1	Java para Web
2	C# para Web
3	Ruby on Rails
4	Cursos de Python
5	PHP com MySql
6	.NET para web
7	Novas integrações com Java
8	TDD com Java
9	Código limpo com C#
10	Preparatório para certificação Java
11	Hardware básico
12	Persistência com .NET
13	Desenvolvendo jogos
14	Análise de dados
15	Estatística básica
16	Internet das coisas
17	Programação funcional
18	Boas práticas em Java
19	Orientação objetos com Java

Com `cursos.head()` , mostraremos as cinco primeiras entradas desse dataframe.

Nome do curso	
0	Lógica de programação
1	Java para Web
2	C# para Web
3	Ruby on Rails
4	Cursos de Python

No próximo vídeo melhoraremos esse novo dataframe de cursos!