

## Um pouco mais sobre o método `get_text()`

Conhecemos em nosso treinamento o método `get_text()`, ou `getText()`, que nos ajuda a acessar os conteúdos de uma *tag* HTML. O método `get_text()` permite a configuração de dois parâmetros, que podem ser úteis no tratamento dos resultados obtidos. O parâmetro `separator` permite a configuração de uma sequência de caracteres, para servir de separador entre os conteúdos das *tags* acessadas:

- In []:

```
soup = BeautifulSoup('<h1>Título 1</h1><h2>Título 1.1</h2><p>Conteúdo</p>')
soup.get_text(' # ')
```

- Out[]:

```
'Título 1 # Título 1.1 # Conteúdo'
```

O segundo parâmetro é o `strip`, que, se configurando como `True`, remove os espaços em branco do início e do fim de cada bloco de texto retornado.

Visto isso, analise o código abaixo:

```
from bs4 import BeautifulSoup

html = """
<html>
  <body>

    <div id="container-a">
      <h1>Título A</h1>
      <h2 class="ref-a">Sub título A</h2>
      <p>Texto de conteúdo A</p>
      <div id="container-a-1">
        <h1>Título A.1</h1>
        <h2 class="ref-a">Sub título A.1</h2>
        <p>Texto de conteúdo A.1</p>
      </div>
    </div>
  </body>
</html>
"""

soup = BeautifulSoup(html, 'html.parser')
soup
```

Considerando o código acima e as novas informações sobre o método `get_text()`, assinale a alternativa que indica o código que retorna todo o conteúdo das *tags* no seguinte formato:

- Out[]:

'Título A || Sub título A || Texto de conteúdo A || Título A.1 || Sub título A.1 || Texto



Selecione uma alternativa

**A** `soup.get_text(separator=' || ', strip=True)`

**B** `soup.getText().split()`

**C** `soup.get_text(separator=' || ')`

**D** `soup.get_text(strip=True)`