

```

1 import requests
2 from bs4 import BeautifulSoup
3 import pandas as pd
4 import numpy as np
5 import plotly.express as px

```

```

1 # Localizamos uma cidade usando o próprio serviço de clima-tempo
2 url = "https://www.tempo.com/estreito_maranhao-1108203.htm"

```

```

1 # usamos a biblioteca requests para fazer o download da página web
2 page = requests.get( url )
3 page

```

```
<Response [200]>
```

```

1 # criamos um objeto beautiful soup para poder fazer a raspagem dos dados
2 soup = BeautifulSoup(page.content, 'html.parser')
3 #soup
4 print(soup.prettify())

```

```

<a href="https://www.microsoft.com/pt-br/store/apps/o-tempo-14-dias/9wzdnckrk" >
  
  <ul class="enlaces download_app">
    <li>
      <a href="https://play.google.com/store/apps/details?id=aplicacion.tiempo&ar
      Baixe Grátis o App Android tempo.com
      </a>
    </li>
    <li>
      <a href="https://itunes.apple.com/br/app/tiempo.com/id543364901?mt=8&pt
      Baixe Grátis o App iOS tempo.com
      </a>
    </li>
    <li>
      <a href="https://appgallery7.huawei.com/#/app/C101411075" title="Baixe Grát
      Baixe Grátis o App Huawei tempo.com
      </a>
    </li>
    <li>
      <a href="https://www.microsoft.com/pt-br/store/apps/o-tempo-14-dias/9wzdnckrk
      Baixe Grátis o App Windows 10 tempo.com
      </a>
    </li>
  </ul>
  <ul class="sociales">
    <li>
      <a class="boton" href="https://www.facebook.com/meteoredoficial" rel="noop
      <span class="icono">
        
      
      
       <ul> <li class="dia d1 activo"> <span class="cuando

```

```

1 forecast_items = datos_dos.find_all("li")
2 #forecast_items

```

```

1 morning = forecast_items[0]
2 morning
3

```

```

<li class="dia d1 activo"> <span class="cuando">Hoje<br/><span>29 Maio</span></span>

```

```

1 # Impressão na tela formatado
2 print(morning.prettify())

```

```

<li class="dia d1 activo">
  <span class="cuando">
    Hoje
    <br/>
    <span>
      29 Maio
    </span>
  </span>
  <span class="prediccion">

```

```

<span class="icono sD-2">
  <svg class="simbw" version="1.1" xmlns="http://www.w3.org/2000/svg">
    <use class="useS" data="sp3" xlink:href="#s2">
      </use>
    </svg>
  </span>
</span>
<span class="temperatura">
  <span class="maxima changeUnitT" data="33.04|0|">
    33°
  </span>
  <span class="barra">
    /
  </span>
  <span class="minima changeUnitT" data="20.08|0|">
    20°
  </span>
</span>
<span class="viento">
  <span class="icono">
    <svg class="wA-36" version="1.1" xmlns="http://www.w3.org/2000/svg">
      <use data="sp2" xlink:href="#windA36">
        </use>
      </svg>
    </span>
    <span class="velocidad">
      <span class="changeUnitW" data="7|1">
        7
      </span>
      -
      <span class="changeUnitW" data="20|1">
        20
      </span>
    <br/>
    <span class="changeUnitW" data="|2">
      km/h
    </span>
  </span>
</span>
</li>

```

```

1 afternoon = forecast_items[1]
2 print(afternoon.prettyfy())

```

```

<li class="dia d2">
  <span class="cuando">
    Amanhã
  <br/>
  <span>
    30 Maio
  </span>
</span>
<span class="prediccion">
  <span class="icono sD-1">
    <svg class="simbw" version="1.1" xmlns="http://www.w3.org/2000/svg">
      <use class="useS" data="sp3" xlink:href="#s1">
        </use>
      </svg>
    </span>

```

```

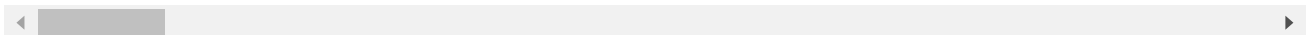
</span>
<span class="temperatura">
  <span class="maxima changeUnitT" data="34.28|0|">
    34°
  </span>
  <span class="barra">
    /
  </span>
  <span class="minima changeUnitT" data="20.56|0|">
    21°
  </span>
</span>
<span class="viento">
  <span class="icono">
    <svg class="wA-43" version="1.1" xmlns="http://www.w3.org/2000/svg">
      <use data="sp2" xlink:href="#windA43">
    </use>
    </svg>
  </span>
  <span class="velocidad">
    <span class="changeUnitW" data="13|1">
      13
    </span>
    -
    <span class="changeUnitW" data="32|1">
      32
    </span>
    <br/>
    <span class="changeUnitW" data="|2">
      km/h
    </span>
  </span>
</span>
</li>

```

## ▼ Extrair informações da página

1 morning

```
<li class="dia d1 activo"> <span class="cuando">Hoje<br/><span>29 Maio</span></span>
```



```
1 #morning.find(class_="cuando").get_text()
```

```
1 period = morning.find(class_="cuando").get_text()
```

```
2 period
```

```
'Hoje29 Maio'
```

```
1 changeUnitW = morning.find(class_="changeUnitW").get_text()
```

```
2 changeUnitW
```

```
'7'
```

```
1 temperatura = morning.find(class_="temperatura").get_ext()
2 temperatura
```

```
1 span = morning.find("span")
2 span
```

```
<span class="quando">Hoje<br/><span>29 Maio</span></span>
```

```
1 # Obtem o título (title) da descrição da img
2 desc = span['class']
3 desc
```

```
['quando']
```

## ▼ Extraindo toda a informação da página

```
1 dados_dos
```

```
<span class="dados-dos-semanas"> <ul> <li class="dia d1 activo"> <span class="quando
```

```
< <span class="quando">Hoje<br/><span>29 Maio</span></span>,
<span class="quando">Amanhã<br/><span>30 Maio</span></span>,
<span class="quando">Segunda<br/><span>31 Maio</span></span>,
<span class="quando">Terça<br/><span>1 Jun.</span></span>,
<span class="quando">Quarta<br/><span>2 Jun.</span></span>,
<span class="quando">Quinta<br/><span>3 Jun.</span></span>,
<span class="quando">Sexta<br/><span>4 Jun.</span></span>]
```

```
1 dados_dos.select(".quando")
```

```
[<span class="quando">Hoje<br/><span>29 Maio</span></span>,
<span class="quando">Amanhã<br/><span>30 Maio</span></span>,
<span class="quando">Segunda<br/><span>31 Maio</span></span>,
<span class="quando">Terça<br/><span>1 Jun.</span></span>,
<span class="quando">Quarta<br/><span>2 Jun.</span></span>,
<span class="quando">Quinta<br/><span>3 Jun.</span></span>,
<span class="quando">Sexta<br/><span>4 Jun.</span></span>]
```

```
1 # Usa o seletor CSS para extrair todos os period-name dentro do quando
2 quando_tags = dados_dos.select(".quando")
3
4 # Usando list comprehensions (não vimos ainda)
5 #periods = [pt.get_text() for pt in period_tags]
6
7 quando = []
8 for pt in quando_tags:
9     quando.append(pt.get_text())
10 quando
```

```
['Hoje29 Maio',
'Amanhã30 Maio',
'Segunda31 Maio',
'Terça1 Jun.',
'Quarta2 Jun.',
```

```
'Quinta3 Jun.',
'Sexta4 Jun.']
```

```
1 # Usando list comprehensions (não vimos ainda)
2 #short_descs = [sd.get_text() for sd in seven_day.select(".tombstone-containr
3
4 temperatura_tags = dados_dos.select(".temperatura")
5 temperatura = []
6 for sd in temperatura_tags:
7     temperatura.append(sd.get_text())
8 temperatura

[' 33° / 20° ',
 ' 34° / 21° ',
 ' 34° / 20° ',
 ' 34° / 22° ',
 ' 32° / 22° ',
 ' 33° / 23° ',
 ' 34° / 22° ']
```

```
1 # Usando list comprehensions (não vimos ainda)
2 # temps = [t.get_text() for t in seven_day.select(".tombstone-container .ter
3
4 changeUnitW_tags = dados_dos.select(".changeUnitW")
5 changeUnitW = []
6 for tp in changeUnitW_tags:
7     changeUnitW.append(tp.get_text())
8 changeUnitW

['7',
 '20',
 'km/h',
 '13',
 '32',
 'km/h',
 '7',
 '19',
 'km/h',
 '8',
 '23',
 'km/h',
 '7',
 '29',
 'km/h',
 '11',
 '32',
 'km/h',
 '10',
 '33',
 'km/h']
```

## ▼ Pandas

```
1 len(cuando)
```

```
7
```

```
1 len(temperatura)
```

```
7
```

```
1 len(changeUnitW)
```

```
21
```

```
1 df_clima_tempo_Estreito = pd.DataFrame({
2     "Período":cuando,
3 })
4 df_clima_tempo_Estreito
```

	Período
0	Hoje29 Maio
1	Amanhã30 Maio
2	Segunda31 Maio
3	Terça1 Jun.
4	Quarta2 Jun.
5	Quinta3 Jun.
6	Sexta4 Jun.

```
1 df_clima_tempo_Estreito = pd.DataFrame({
2     "Vento":changeUnitW,
3 })
4 df_clima_tempo_Estreito
```

Vento	
0	7
1	20
2	km/h
3	13
4	32
5	km/h
6	7
7	19
8	km/h
9	8
10	23
11	km/h
12	7

```

1 df_clima_tempo_Estreito = pd.DataFrame({
2     "Temperatura":temperatura,
3 })
4 df_clima_tempo_Estreito

```

Temperatura	
0	33° / 20°
1	34° / 21°
2	34° / 20°
3	34° / 22°
4	32° / 22°
5	33° / 23°
6	34° / 22°

## ▼ Média de temperatura em Estreito no Maranhão

```

1 df_clima_tempo_Estreito["Temperatura"]

0      33° / 20°
1      34° / 21°
2      34° / 20°
3      34° / 22°
4      32° / 22°
5      33° / 23°

```



```
6      34° / 22°  
Name: Temperatura, dtype: object
```

```
1 # Neste caso, a expressão regular para extrair os dígitos da string  
2 temperatura_nums = df_clima_tempo_Estreito["Temperatura"].str.extract(r"(\d-  
3 temperatura_nums
```

```
0      33  
1      34  
2      34  
3      34  
4      32  
5      33  
6      34  
Name: Temperatura, dtype: object
```

```
1 # O tipo resultante de Series.str.extract  
2 type(temperatura_nums)
```

```
pandas.core.series.Series
```

```
1 # Converte os valores da coluna temp_num em inteiro  
2 df_clima_tempo_Estreito["Temperatura"] = temperatura_nums.astype('int')  
3 df_clima_tempo_Estreito["Temperatura"]
```

```
0      33  
1      34  
2      34  
3      34  
4      32  
5      33  
6      34  
Name: Temperatura, dtype: int32
```

```
1 df_clima_tempo_Estreito["Temperatura"].mean()
```

```
33.42857142857143
```

```
1
```

---

✓ 0s conclusão: 22:00

● ✕