```
mkdir web-crawler
cd web-crawler
python -m venv env
```

```
  File "<ipython-input-1-96f40c77535b>", line 1
    mkdir web-crawler
          ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
import requests
from bs4 import BeautifulSoup
```

```
python crawler.py
```

```
  File "<ipython-input-3-94bfee25412b>", line 1
    python crawler.py
           ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
# Imports
import requests
import numpy as np
import pandas as pd
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
import re
import os
%matplotlib inline
riturl="https://purdue.edu"
webpage = requests.get(riturl)
ritsoup = BeautifulSoup(webpage.content, "html.parser")
print(ritsoup)
```

```
<!DOCTYPE html>

<html class="is-fullheight" lang="en-US">
<head>
<title>Purdue University</title>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<link href="http://gmpg.org/xfn/11" rel="profile"/>
<!-- Google Tag Manager for WordPress by gtm4wp.com -->
<script data-cfasync="false" data-pagespeed-no-defer="">
        var gtm4wp_datalayer_name = "dataLayer";
        var dataLayer = dataLayer || [];
</script>
<!-- End Google Tag Manager for WordPress by gtm4wp.com --><link href="//use.typekit.net" rel="dns-prefetch">
<link href="//fonts.googleapis.com" rel="dns-prefetch"/>
<link href="//use.fontawesome.com" rel="dns-prefetch"/>
<script type="text/javascript">
window._wpemojiSettings = {"baseUrl":"https:\/\/s.w.org\/images\/core\/emoji\/14.0.0\/72x72\/","ext":".png","svgUrl":"https:\/\/s
/*! This file is auto-generated */
!function(i,n){var o,s,e;function c(e){try{var t={supportTests:e,timestamp:(new Date).valueOf()};sessionStorage.setItem(o,JSON.st
</script>
<style type="text/css">
img.wp-smiley,
img.emoji {
        display: inline !important;
        border: none !important;
        box-shadow: none !important;
        height: 1em !important;
        width: 1em !important;
        margin: 0 0.07em !important;
        vertical-align: -0.1em !important;
        background: none !important;
        padding: 0 !important;
}
</style>
<link href="https://www.purdue.edu/home/wp-includes/css/dist/block-library/common.min.css?ver=6.3.2" id="wp-block-library-css" me
<link href="https://www.purdue.edu/home/wp-includes/css/classic-themes.min.css?ver=6.3.2" id="classic-theme-styles-css" media="al
<link href="https://use.typekit.net/ghc8hdz.css?ver=6.3.2" id="brandfonts-css" media="all" rel="stylesheet" type="text/css"/>
<link href="https://www.purdue.edu/home/wp-content/mu-plugins/boilerup-wp/unitedsans.css?ver=6.3.2" id="unitedsans-css" media="al
<link href="https://fonts.googleapis.com/css2?family=Source+Serif+Pro%3Awght%40400%3B600%3B700&amp;display=swap&amp;ver=6.3.2" id
<link href="https://www.purdue.edu/home/wp-content/plugins/wp-accessibility/css/wpa-style.css?ver=2.0.1" id="wpa-style-css" media
<style id="wpa-style-inline-css" type="text/css">
:root { --admin-bar-top : 7px; }
</style>
```

```
<link href="https://use.fontawesome.com/releases/v6.4.2/css/all.css?ver=6.3.2" id="load-fa-css" media="all" rel="stylesheet" type
<link href="https://www.purdue.edu/home/wp-content/themes/purdue-home-theme/style.css?ver=6.3.2" id="purdueBrand-style-css" media
<link href="https://www.purdue.edu/home/wp-content/themes/purdue-home-theme/build/app.css?ver=00e70e7c92ef88f9a348" id="purdueBra
<script id="jquery-core-js" src="https://www.purdue.edu/home/wp-includes/js/jquery/jquery.min.js?ver=3.7.0" type="text/javascript
<script id="jquery-migrate-js" src="https://www.purdue.edu/home/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.4.1" type="text
<link href="https://www.purdue.edu/home/wp-json/" rel="https://api.w.org/"/><link href="https://www.purdue.edu/home/wp-json/wp/v2
<link href="https://www.purdue.edu/home/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fwww.purdue.edu%2Fhome%2F" rel="alternate" type
<link href="https://www.purdue.edu/home/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fwww.purdue.edu%2Fhome%2F&amp;format=xml" rel="
<link href="https://www.purdue.edu/home/wp-content/mu-plugins/boilerup-wp/favicon/favicon.ico" rel="shortcut icon" type="image/x-
<link href="https://www.purdue.edu/home/wp-content/mu-plugins/boilerup-wp/favicon/apple-icon-57x57.png" rel="apple-touch-icon" si
<link href="https://www.purdue.edu/home/wp-content/mu-plugins/boilerup-wp/favicon/apple-icon-60x60.png" rel="apple-touch-icon" si
```

```
print("Title of the parsed page : ",ritsoup.title)
print()
print("All the links : ")
links = [link.get('href') for link in ritsoup.find_all('a')]
print(links,"\n")
print("Accessing elements of the parsed page : ")
print("Accessing heading element : ",ritsoup.h1)
print(ritsoup.head)
print(ritsoup.head.meta)
paragraphs = ritsoup.find_all("p")
print()
print("Get all <p> elements : ",paragraphs)
print()
print("Gets all the p elements with a class attribute with value hide: ",ritsoup.find_all("p",
attrs={"class": "hide"}))
print()
# We can use regular expression too!
ritsoup.find_all(re.compile("^(p|a)$"))[: 3]
```

```
<link href="https://www.purdue.edu/home/wp-includes/css/classic-themes.min.css?ver=6.3.2" id="classic-theme-styles-css" media="al
<link href="https://use.typekit.net/ghc8hdz.css?ver=6.3.2" id="brandfonts-css" media="all" rel="stylesheet" type="text/css"/>
<link href="https://www.purdue.edu/home/wp-content/mu-plugins/boilerup-wp/unitedsans.css?ver=6.3.2" id="unitedsans-css" media="al
<link href="https://fonts.googleapis.com/css2?family=Source+Serif+Pro%3Awght%40400%3B600%3B700&amp;display=swap&amp;ver=6.3.2" id
<link href="https://www.purdue.edu/home/wp-content/plugins/wp-accessibility/css/wpa-style.css?ver=2.0.1" id="wpa-style-css" media
<style id="wpa-style-inline-css" type="text/css">
:root { --admin-bar-top : 7px; }
</style>
<link href="https://use.fontawesome.com/releases/v6.4.2/css/all.css?ver=6.3.2" id="load-fa-css" media="all" rel="stylesheet" type
<link href="https://www.purdue.edu/home/wp-content/themes/purdue-home-theme/style.css?ver=6.3.2" id="purdueBrand-style-css" media
<link href="https://www.purdue.edu/home/wp-content/themes/purdue-home-theme/build/app.css?ver=00e70e7c92ef88f9a348" id="purdueBra
<script id="jquery-core-js" src="https://www.purdue.edu/home/wp-includes/js/jquery/jquery.min.js?ver=3.7.0" type="text/javascript
<script id="jquery-migrate-js" src="https://www.purdue.edu/home/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.4.1" type="text
<link href="https://www.purdue.edu/home/wp-json/" rel="https://api.w.org/"/><link href="https://www.purdue.edu/home/wp-json/wp/v2
<link href="https://www.purdue.edu/home/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fwww.purdue.edu%2Fhome%2F" rel="alternate" type
<link href="https://www.purdue.edu/home/wp-json/oembed/1.0/embed?url=https%3A%2F%2Fwww.purdue.edu%2Fhome%2F&amp;format=xml" rel="
```

```
print("Obtaining strings : ",ritsoup.h1.string)
print()
#The contents method is similar but always returns a list:
print("Obtaining strings using contents method : ", ritsoup.h1.contents)
print()
#If the element contains any tags, then string will return None
print(paragraphs[2],"\n")
print(paragraphs[2].string,"\n")
#However, contents will return a list as before, mixing different kinds of elements:
para2 = paragraphs[2].contents
print(para2,"\n")
para7=paragraphs[5].contents
print(para7,"\n")
#You can also use stripped_strings, which is a generator over all the strings (tags
#removed)
#inside the element; this is a fast way to extract the raw texts, with all tag soup strained
#off:
for s in paragraphs[3].stripped_strings:
  print("="*50)
print(s)
```

```
Obtaining strings :  Every Giant Leap Starts with One Small Step

Obtaining strings using contents method :  ['Every Giant Leap Starts with One Small Step']

<p class="purdue-home-cta-card__cta-text">Give the Gift of Purdue</p>

Give the Gift of Purdue

['Give the Gift of Purdue']

['See the Boilermakers – long known as the “Cradle of Quarterbacks” – take on other Big Ten Conference teams.']

==================================================
Ross-Ade Stadium
```