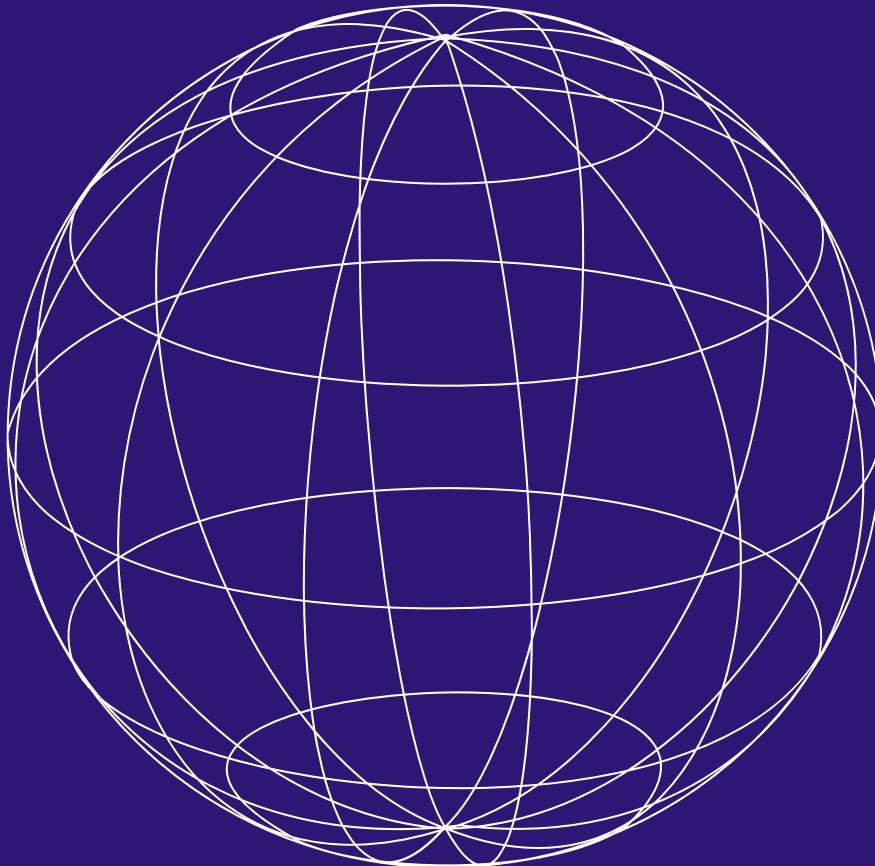


# ADAPTANDO LIVROS WEB PARA PDF COM WEB SCRAPING E PDFKIT

# Repositório no GitHub



Screenshot of a GitHub repository page for "convert-html-to-pdf".

**Repository Details:**

- Owner: gabriel-batistuta
- Name: convert-html-to-pdf
- Status: Public
- Last commit: 5a9e451 1 minute ago
- Commits: 9
- Branches: 1
- Tags: 0

**Commit History:**

Commit	Message	Time Ago
core	código funcional com a adição de funções especiais	1 minute ago
tests	código funcional com a adição de funções especiais	1 minute ago
.gitignore	código funcional com a adição de funções especiais	1 minute ago
LICENSE	Initial commit	2 weeks ago
README.md	Initial commit	2 weeks ago
main.py	código funcional com a adição de funções especiais	1 minute ago
requirements.txt	código funcional com a adição de funções especiais	1 minute ago

**README.md Content:**

```
convert-html-to-pdf

repositório de tutorial para uso de Web Scraping e pdfkit para transformar livros em pdf usando Python
```

**Repository Metadata:**

- Language: Python 100.0%
- License: MIT license
- Activity: 0 stars, 1 watching, 0 forks, 2 weeks old

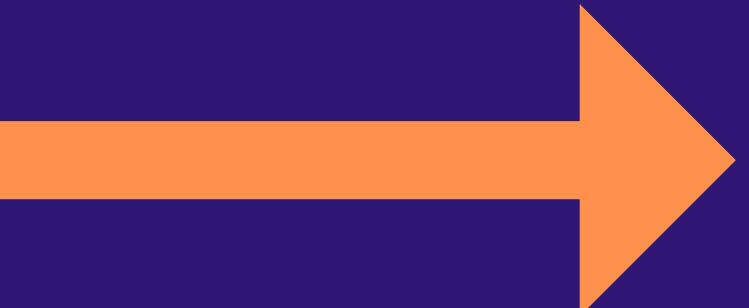
**Suggested Workflows:**

- Python application: Create and test a Python application.
- SLSA Generic generator: Generate SLSA3 provenance for your

<https://github.com/gabriel-batistuta/convert-html-to-pdf>

# O que é Web Scraping ?

é uma forma de mineração que permite a extração de dados de sites da web convertendo-os em informação estruturada para posterior análise ou outro uso específico.



# Inspecionar do navegador

Copy-paste the information manually into a new document. Here's where web scraping can help.

## What is Web Scraping?

Web scraping refers to the **extraction of data from a website**. This information is collected and then exported into a format that is more useful for the user. Be it a spreadsheet or an API.

figure.kg-card.kg-image-card | 600 x 337.5 | Flex Item

### What is web scraping?



```
Ferramentas de desenvolvimento — What Is Web Scraping and What Is It Used For? | ...
Inspetor Console Debugger Rede Editor de estilos
Q Pesquisar HTML
<!DOCTYPE html>
<html lang="en"> (event) (deslocamento)
  <head> ([])
  <body class="post-template tag-web-scraping" cz-shortcut-listen="true"> (transbordamento)
    <div class="site-wrapper" (flex)
      <header class="site-header outer" ([])
      <main id="site-main" class="site-main outer">
        <div class="inner" (transbordamento)
          <article class="post-full post tag-web-scraping featured" ([])
            <header class="post-full-header" ([])
            <figure class="post-full-image" ([])
            <section class="post-full-content">
              ::before
              <div class="post-content" (flex)
                <p> ([])
                <p> ([])
                <p> ([])
                <p> ([])
                <h2 id="what-is-web-scraping">What is Web Scraping?
                <p> ([])
                <!--kg-card-begin: image-->
                <figure class="kg-card kg-image-card" ([])
                  <!--kg-card-end: image-->
                <p> ([])
                <p> ([])
                <p> ([])
                <ul> ([])
                  <li> ([])
                <p> ([])
                <!--kg-card-begin: html-->
                <div class="fluid-width-video-container" ([])
                  <!--kg-card-end: html-->
                <h2 id="is-web-scraping-legal">Is web scraping legal?
                <p> ([])
                <p> ([])
                <p> ([])
                <h2 id="how-do-web-scrappers-work">How do Web Scrapers Work?
                <p> ([])
                <p> ([])
                <p> ([])
            
```

# Project Gutenberg

https://www.gutenberg.org

About Search and Browse Help

Quick search Go! Donation PayPal

## Welcome to Project Gutenberg

Project Gutenberg is a library of over 70,000 free eBooks

Choose among free epub and Kindle eBooks, download them or read them online. You will find the world's great literature here, with focus on older works for which U.S. copyright has expired. Thousands of volunteers digitized and diligently proofread the eBooks, for you to enjoy.



Rogue psi by James H. Schmitz  
Lettres de voyage by Rudyard Kipling  
Bits from Blackberry by John Strathesk  
Pakenvien parissa by Maria Salmela  
The willow weaver and seven other Latin maxims & rules, in law  
A collection of Latin maxims & rules, in law  
Influenza by Warren T. Vaughan  
The bedbug by Anonymous  
Peck's Bad Boy in an airship by An open verdict, Volume 2 (of)

Some of our latest eBooks [Click Here for more latest books!](#)

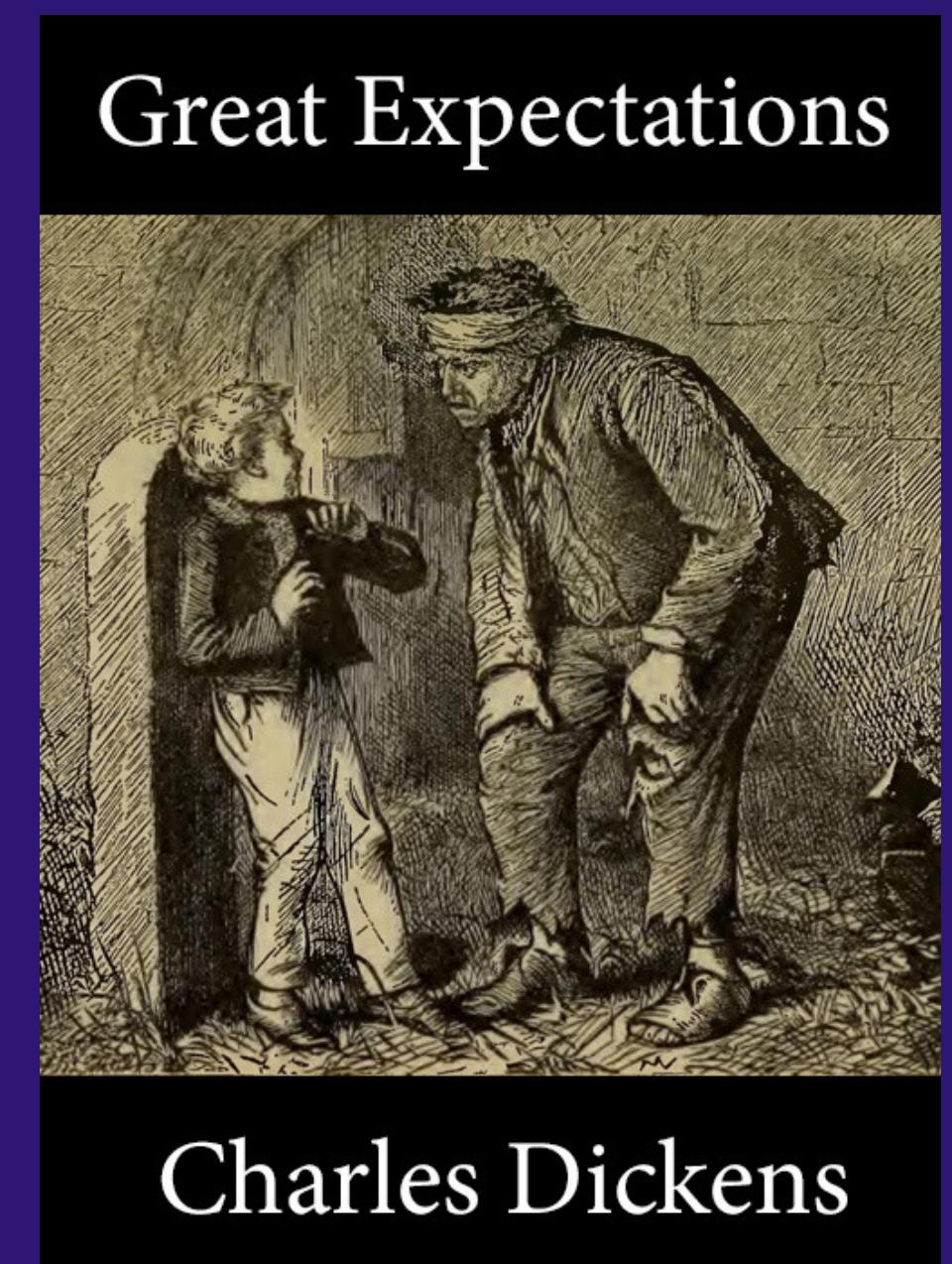
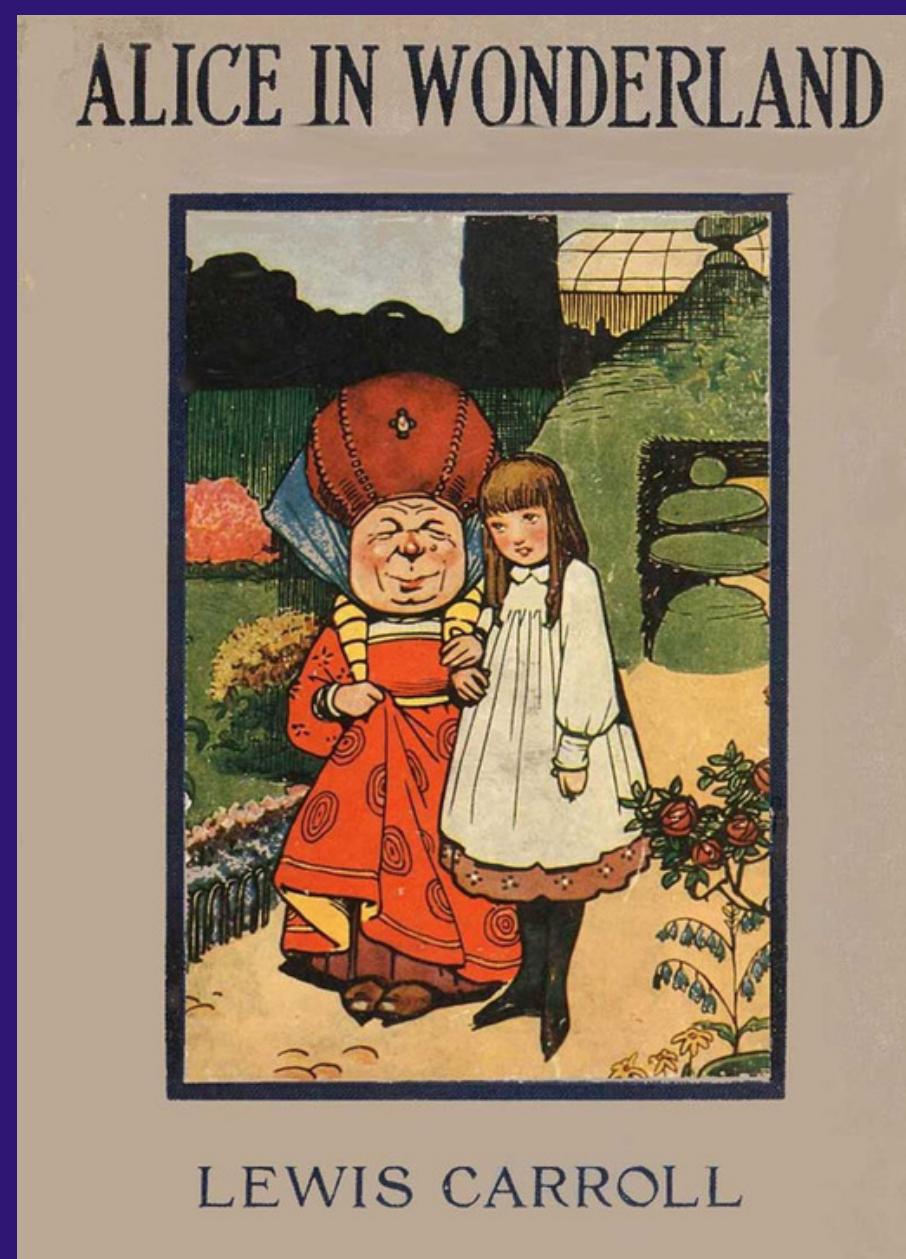
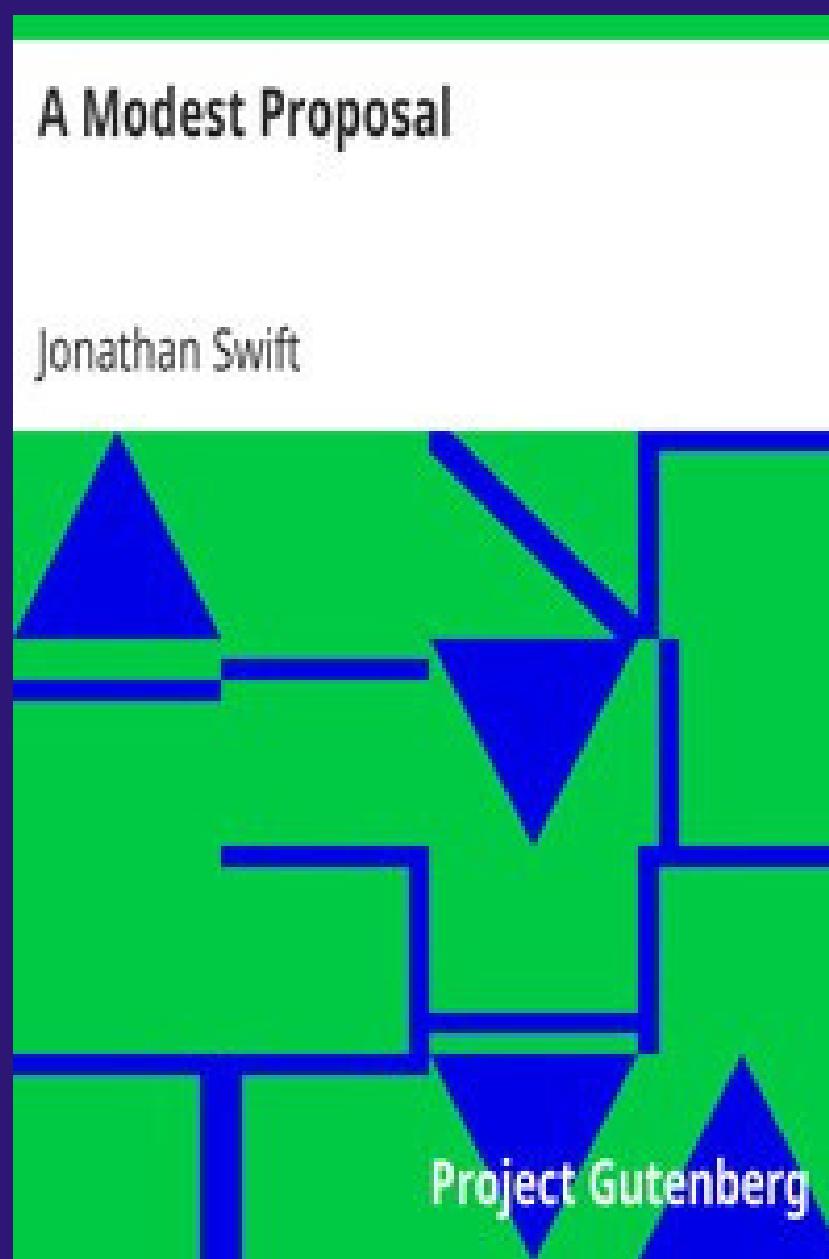
Trouble downloading or viewing books? [Let us know](#). We have made some fixes that seem to address some issues that had been reported, and we would like to hear if any troubles persist.

No fee or registration! Everything from Project Gutenberg is gratis, libre, and completely without cost to readers. If you find Project Gutenberg useful, please consider a small donation to help Project Gutenberg digitize more books, maintain its online presence, and improve Project Gutenberg programs and offerings. Other ways to help include digitizing, proofreading and formatting, or reporting errors.

No special apps needed! Project Gutenberg eBooks require no special apps to read, just the regular Web browsers or eBook readers that are included with computers and mobile devices. There have been reports of sites that charge fees for custom apps, or for the same eBooks that are freely available from Project Gutenberg. Some of the apps might have worthwhile features, but none are required to enjoy Project Gutenberg eBooks.

50 years of eBooks 1971-2021. In 2021, Project Gutenberg celebrated the [first eBook](#) for reading enjoyment and unlimited free redistribution. This eBook was created on July 4, 1971 by Project Gutenberg's founder, Michael

# Livros a serem adaptados



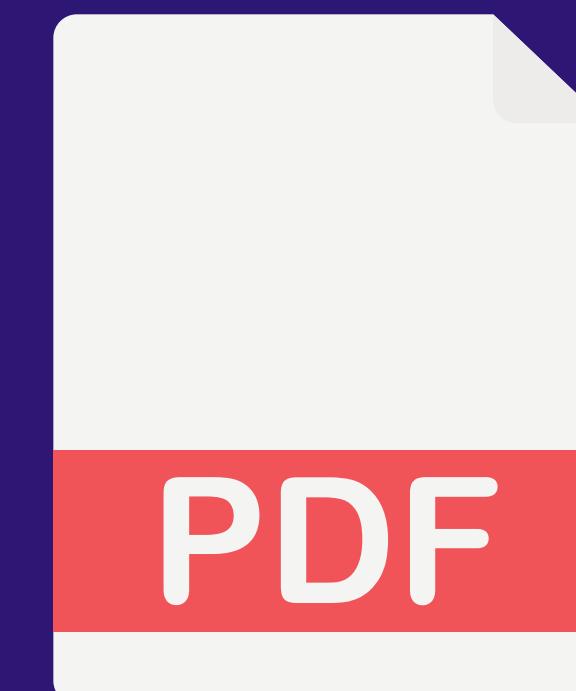
# Resultado final

```
→ books git:(dev) ✘ tree
```

```
.
├── Alice's Adventures in Wonderland
│   ├── Alice's Adventures in Wonderland.pdf
│   ├── cover - Alice's Adventures in Wonderland.jpg
│   ├── images
│   └── README.txt
├── A Modest Proposal
│   ├── A Modest Proposal.pdf
│   ├── images
│   └── README.txt
└── Great Expectations
    ├── cover - Great Expectations.jpg
    ├── Great Expectations.pdf
    └── images
        ├── image 10 - Great Expectations.jpg
        ├── image 11 - Great Expectations.jpg
        ├── image 12 - Great Expectations.jpg
        ├── image 13 - Great Expectations.jpg
        ├── image 14 - Great Expectations.jpg
        ├── image 15 - Great Expectations.jpg
        ├── image 1 - Great Expectations.jpg
        ├── image 2 - Great Expectations.jpg
        └── image 3 - Great Expectations.jpg
```



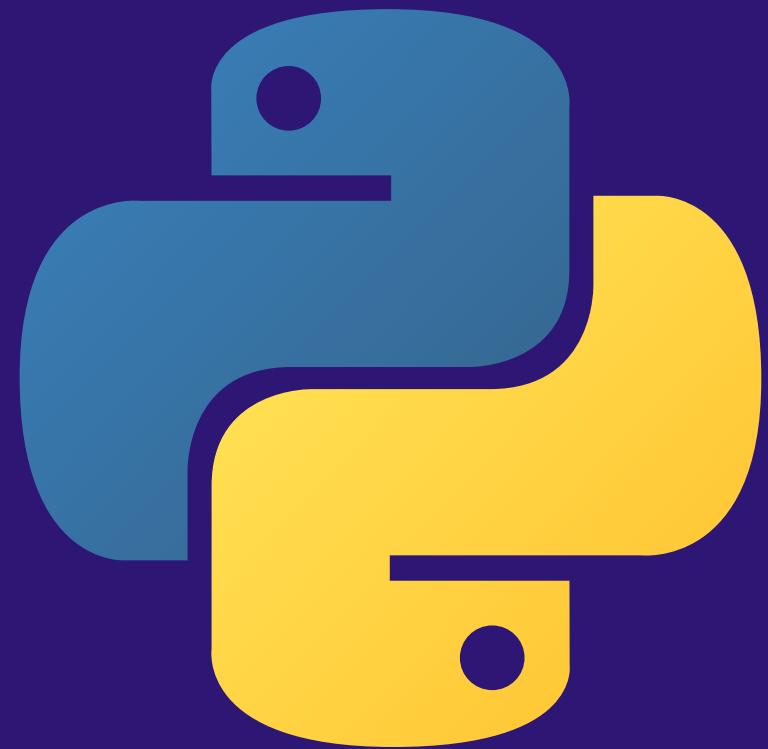
TXT



PDF



# Dependências



- requests
- bs4
- pdfkit

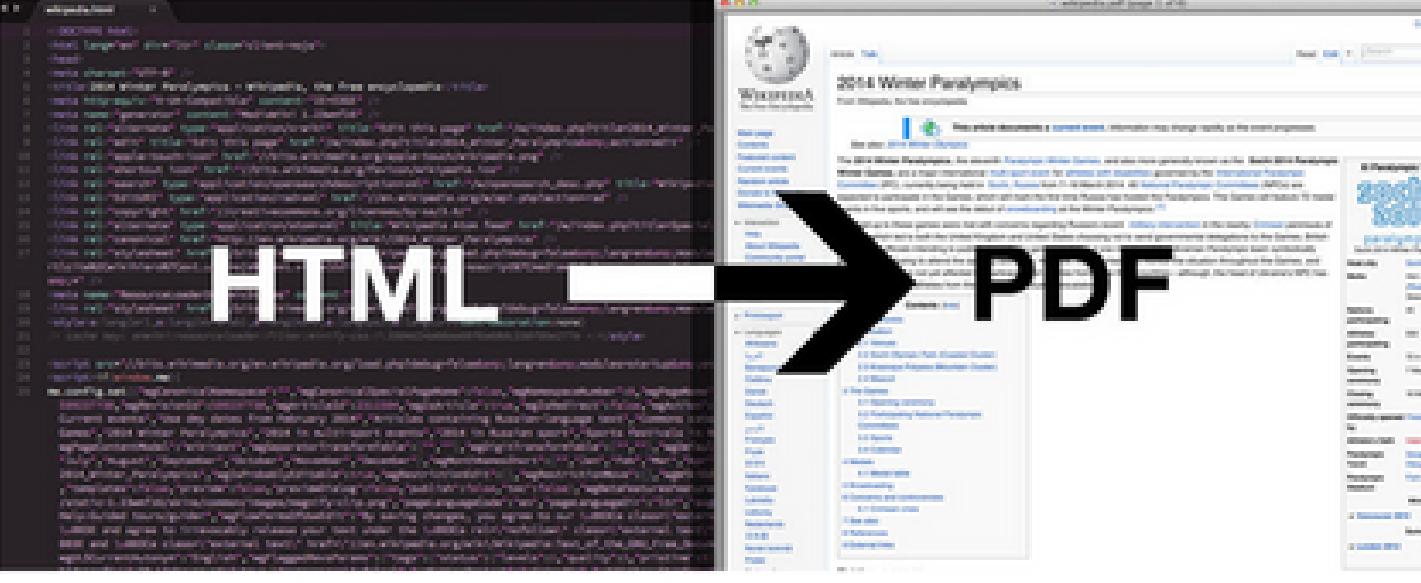
`pip install -r requirements.txt`

# Qt

# WKHTMLTOPDF

WK<html>TOPdf

Github Docs Status Support Downloads



**HTML → PDF**

**What is it?**

wkhtmltopdf and wkhtmltoimage are open source (LGPLv3) command line tools to render HTML into PDF and various image formats using the Qt WebKit rendering engine. These run entirely "headless" and do not require a display or display service.

There is also a C library, if you're into that kind of thing.

**How do I use it?**

1. Download a precompiled binary or build from source
2. Create your HTML document that you want to turn into a PDF (or image)
3. Run your HTML document through the tool.

For example, if I really like the treatment Google has done to their logo today and want to capture it forever as a PDF:

```
wkhtmltopdf http://google.com google.pdf
```

Additional options

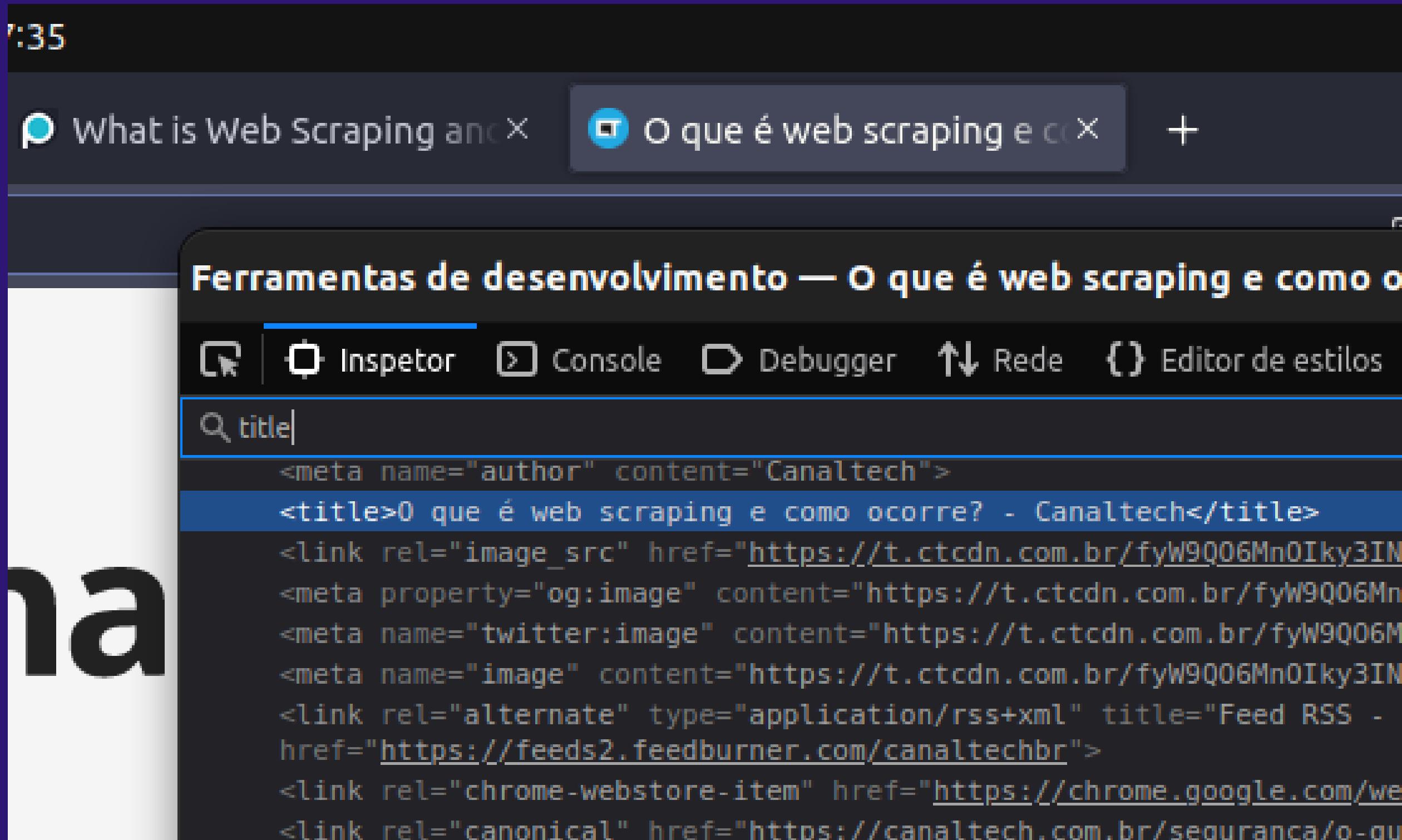
<https://wkhtmltopdf.org/index.html>

# BeautifulSoup Sintaxe



```
6 url = 'https://canaltech.com.br/seguranca/o-que-e-web-scraping/'  
7 response = requests.get(url)  
8 html_content = response.text  
9 soup = BeautifulSoup(html_content, 'html.parser')  
10  
11 print(type(soup))  
12 # <class 'bs4.BeautifulSoup'>  
13  
14 print(soup.title)  
15 # <title>O que é web scraping e como ocorre? - Canaltech</title>  
16  
17 with open('.html', 'w') as file:  
18     file.write(soup.prettify())
```

# soup.title



# Método pretty()

```
html > ...
369      </path>
370    </svg>
371    YT CT en Español
372  </a>
373 </li>
374 <li class="c-iWqbnf">
375   <a href="https://www.youtube.com/channel/UC_bXJnsgw0qEPA_-6N6faKw" rel="nofollow" target="_blank" title="YT
376   Canaltech">
377     <svg height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
378       <path d="M17 10.5V7a1 1 0 0 0-1H4a1 1 0 0 0-1v10a1 1 0 0 0 1h12a1 1 0 0 0 1-lv-3.5l4 4v-11l-4 42"
379         fill="currentColor">
380       </path>
381     </svg>
382   YT Canaltech
383   </a>
384 </li>
385 <li class="c-iWqbnf">
386   <a href="https://www.youtube.com/canaltechnews" rel="nofollow" target="_blank" title="YT Canaltech News">
387     <svg height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
388       <path d="M17 10.5V7a1 1 0 0 0-1H4a1 1 0 0 0-1v10a1 1 0 0 0 1h12a1 1 0 0 0 1-lv-3.5l4 4v-11l-4 42"
389         fill="currentColor">
390       </path>
391     </svg>
392   YT Canaltech News
393   </a>
394 </li>
395 <li class="c-iWqbnf">
396   <a href="https://www.youtube.com/c/MaravilhasdaTecnologia" rel="nofollow" target="_blank" title="YT Maravilhas da
397   Tecnologia">
398     <svg height="24" viewBox="0 0 24 24" width="24" xmlns="http://www.w3.org/2000/svg">
399       <path d="M17 10.5V7a1 1 0 0 0-1H4a1 1 0 0 0-1v10a1 1 0 0 0 1h12a1 1 0 0 0 1-lv-3.5l4 4v-11l-4 42"
400         fill="currentColor">
```

# BeautifulSoup Sintaxe



```
with open('.html', 'w') as file:  
    file.write(soup.prettify())  
  
div = soup.find('div', attrs={'class': 'c-bNTyLN'})  
print(type(div))  
# <class 'bs4.element.Tag'>  
  
sub_div = div.find('div', attrs={'class': 'c-leepeta'})  
print(sub_div)  
# <div class="c-leepeta">  
# ...  
# </div>  
  
links = soup.find_all('a')  
for link in links:  
    print(f'{link}')  
    print(f'{link["href"]}\n')  
    # <a href="https://especiais.magazineluiza.com.br/seguranca/" rel="norefferrer"  
    target="_blank" title="Segurança & Privacidade">Segurança & Privacidade</a>  
    # https://especiais.magazineluiza.com.br/seguranca/
```

# Arquivo main do código

```
> books
  < core
    > __pycache__
      __init__.py
      folder.py
      header.py
      image.py
      pdf.py
      source.py
      title.py
    > template
    > tests
      .gitignore
      LICENSE
      main.py
      README.md
      requirements.txt
      test.py
```

```
1 import core
2 # Scraping de livros do site: 'https://www.gutenberg.org/'
3
4 # livro: A Modest Proposal (sem imagens)
5 book_link1 = 'https://www.gutenberg.org/cache/epub/1080/pg1080-images.html'
6 # livro: Alice's Adventures in Wonderland (só tem a capa)
7 book_link2 = 'https://www.gutenberg.org/cache/epub/11/pg11-images.html'
8 # livro: Great Expectations (contém imagens)
9 book_link3 = 'https://www.gutenberg.org/cache/epub/1400/pg1400-images.html'
10
11 book_list = [book_link1, book_link2, book_link3]
12
13 def main(book_list):
14     for book_link in book_list:
15         source_html = core.get_source_html_from_url(book_link)
16         domain = core.get_domain(book_link)
17         title = core.get_title(source_html)
18         core.make_book_folder(title)
19         core.create_images_folder(title)
20         image_obj = core.get_images_url(source_html)
21         if image_obj['cover_link'] is not None and image_obj['image_links'] is not None:
22             core.download_images(domain, title, image_obj['cover_link'], image_obj
23             ['image_links'])
24         elif image_obj['cover_link'] is not None and image_obj['image_links'] is None:
25             core.download_images(domain, title, image_obj['cover_link'])
26         core.write_header(source_html, title)
27         core.create_pdf(source_html, domain, title)
28
29 if __name__ == '__main__':
30     main(book_list)
```

# Módulo source

```
import requests
from bs4 import BeautifulSoup

def get_source_html_from_url(url):

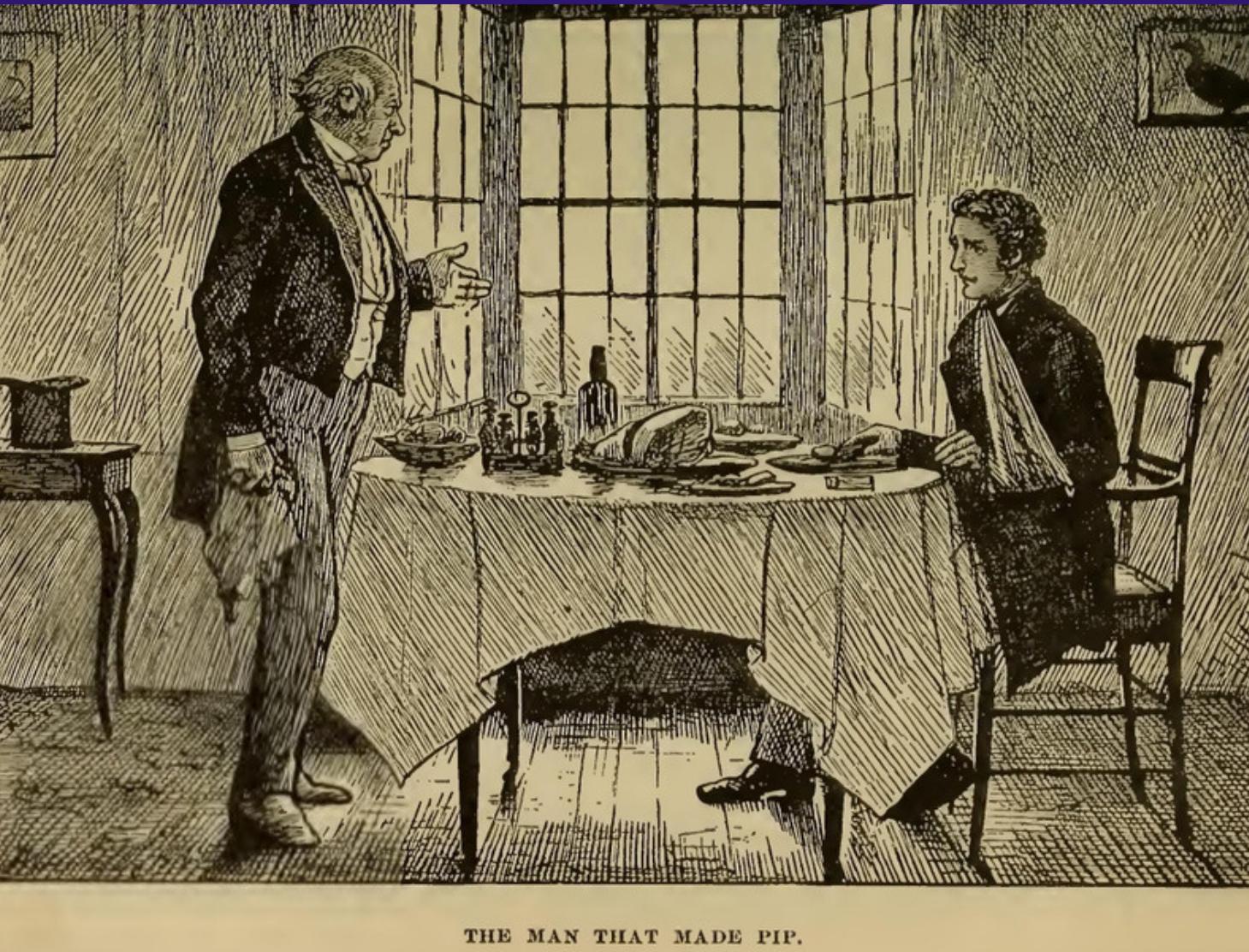
    response = requests.get(url)
    content = response.content
    source_html = BeautifulSoup(content, 'html.parser')

    return source_html
```



# Caminho relativo x absoluto

## “images/0393m.jpg”



caminhos utilizados em sistemas geralmente são relativos

# Módulo source

```
def get_domain(url):

    path_separated = url.split('/')
    # ['https:', '', 'www.gutenberg.org', 'cache', 'epub', '1400', 'pg1400-images.html']

    path_separated[-1] = ''
    # ['https:', '', 'www.gutenberg.org', 'cache', 'epub', '1400', '']

    domain_path = '/'.join(path_separated)
    # https://www.gutenberg.org/cache/epub/1400/

    return domain_path
```



# Módulo title

```
def get_title(source_html):
    # extraindo o texto da tag titulo
    title_tag = source_html.find('h1')
    print(title_tag)
    # <h1>Alice's Adventures in Wonderland</h1>

    title_text = title_tag.text.strip()
    title = title_text.replace('\n', ' ').replace(' ', ' ')
    print(f'título: {title} || {__name__}')
    # título: Alice's Adventures in Wonderland || core.title

    return title
```

# Módulo folder



```
import os

def make_book_folder(_title):
    if os.path.isdir('./books'):
        pass
    else:
        os.mkdir('./books')

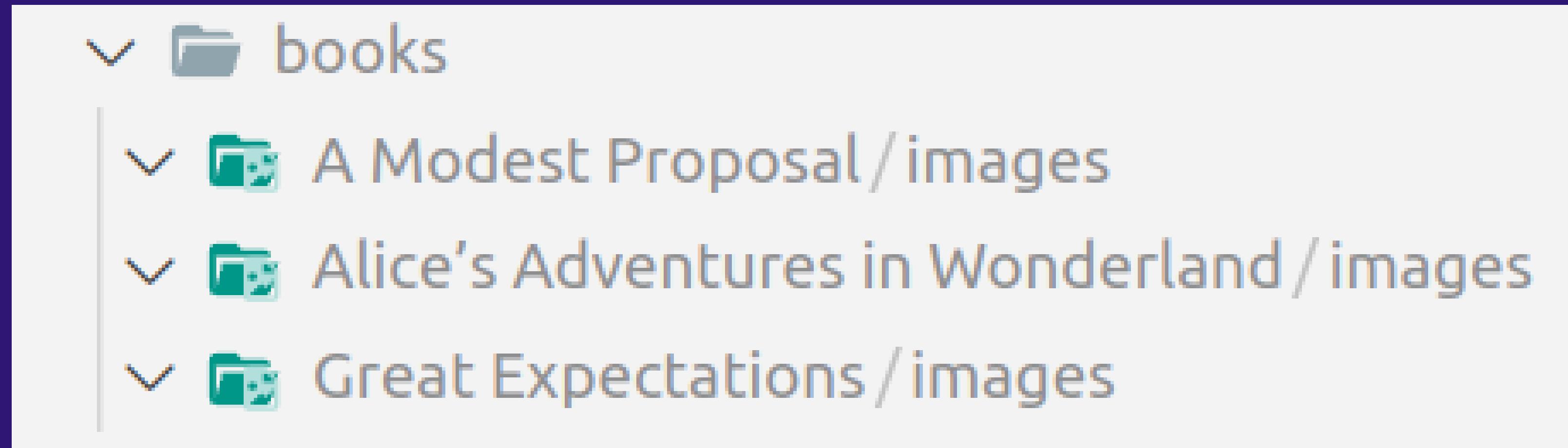
    if os.path.isdir(f'./books/{_title}'):
        pass
    else:
        os.mkdir(f'./books/{_title}'')
```

# Módulo folder

```
def create_images_folder(_title):
    if os.path.isdir(f'./books/{_title}/images'):
        pass
    else:
        os.mkdir(f'./books/{_title}/images')
```



# Pastas criadas



# função padrão para baixar imagens no módulo

```
import requests

def write_image(file_path, image_link):

    # pega o conteúdo da imagem
    # escreve seus binários em um arquivo jpg criado
    with open(file_path, 'wb') as file:
        data = requests.get(image_link)
        file.write(data.content)
        file.close()
```



# função de pegar links das imagens

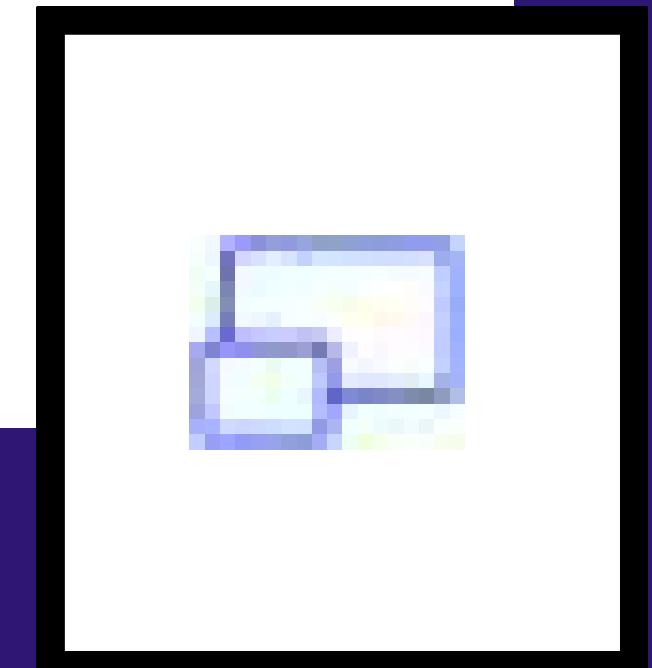
```
def get_images_url(source_html):
    image_links = []

    def remove_min_img_and_return_bs4(source_html:BeautifulSoup):
        pattern = re.compile(r'<img[^>]*src="images/enlarge\.jpg"[^>]*?>')

        # Substituir as correspondências pela string vazia (remover as tags)
        source_html_string = re.sub(pattern, '', str(source_html))

    return BeautifulSoup(source_html_string)

source_html = remove_min_img_and_return_bs4(source_html)
imgs_tags = source_html.find_all('img')
```



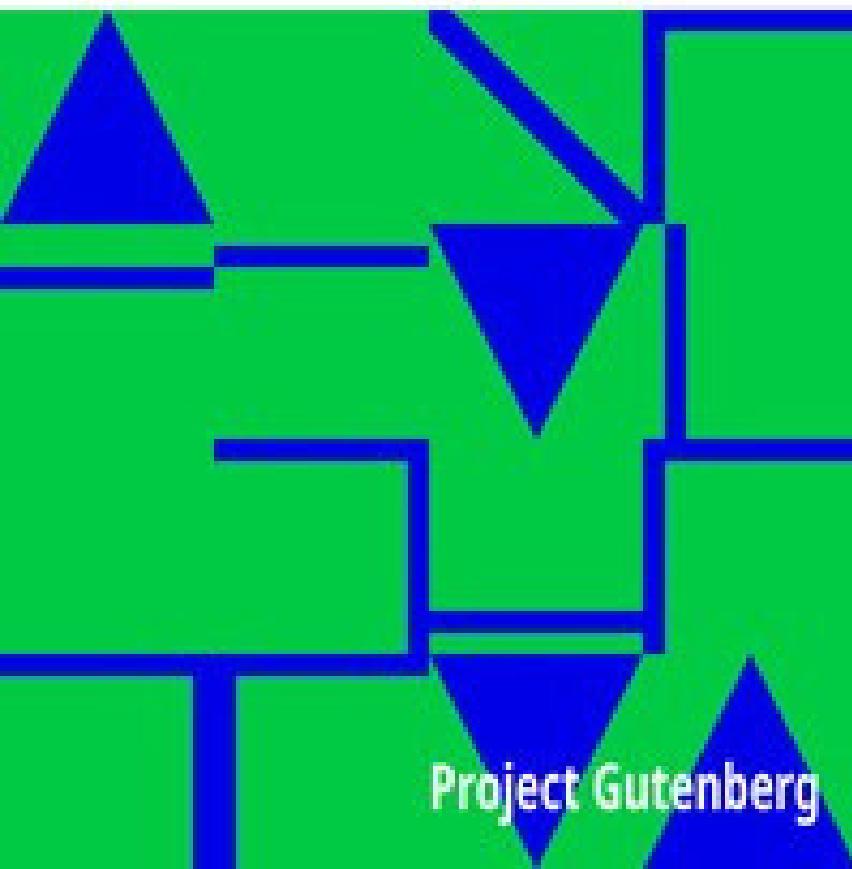
# função de pegar links das imagens

```
# livro não tem imagem nenhuma
if len(imgs_tags) == 0:
    return {
        'cover_link':None,
        'image_links':None
    }

# livro tem só a capa
elif len(imgs_tags) == 1:
    cover_tag = imgs_tags.pop(0)
    cover_link = cover_tag['src']
    return {
        'cover_link':cover_link,
        'image_links':None
    }
```

A Modest Proposal

Jonathan Swift



Project Gutenberg

ALICE IN WONDERLAND



LEWIS CARROLL

# Caso de mais de uma imagem

```
# livro tem capa e outras imagens
else:
    # retira a primeira imagem(capa do livro) da lista
    # e passa para outra variavel
    cover_tag = imgs_tags.pop(0)
    cover_link = cover_tag['src']
    # filtra os links das tags em outra lista
    for img in imgs_tags:
        image_links.append(img['src'])

return {
    'cover_link':cover_link,
    'image_links':image_links
}
```



Great Expectations



Charles Dickens

# Passando parâmetros pro download

```
image_obj = core.get_images_url(source_html)
if image_obj['cover_link'] is not None and image_obj['image_links'] is not None:
    core.download_images(domain, title, image_obj['cover_link'], image_obj['image_links'])
elif image_obj['cover_link'] is not None and image_obj['image_links'] is None:
    core.download_images(domain, title, image_obj['cover_link'])
```

`cover_link` e `image_links` são opcionais

# Função de download de imagens

```
def download_images(domain, title, cover=None, image_links=None):
    # baixa a imagem de capa na raiz da pasta do livro
    def download_cover(cover, title, domain):
        write_image(f'books/{title}/cover - {title}.jpg', domain+cover)

    # baixa as imagens na pasta images
    def download_images_in_folder(image_links, title, domain):
        # retirando as imagens de maximização de imagens da página
        image_links = list(filter(lambda x: 'enlarge' not in x, image_links))

        # baixando as imagens e nomeando por sequência em ordem
        for index, image_link in enumerate(image_links):
            write_image(f'books/{title}/images/image {index+1} - {title}.jpg', domain+image_link)
```

essa função usa a função **write\_image** anterior

# cabeçalho do site

## The Project Gutenberg eBook of Great Expectations

This ebook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this ebook or online at [www.gutenberg.org](http://www.gutenberg.org). If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Great Expectations

Author: Charles Dickens

Release date: July 1, 1998 [eBook #1400]

Most recently updated: October 8, 2023

Language: English

Credits: An Anonymous Volunteer and David Widger

\*\*\* START OF THE PROJECT GUTENBERG EBOOK GREAT EXPECTATIONS \*\*\*

## Great Expectations



# footer do site

*And will continue friends apart,* said Estella.

I took her hand in mine, and we went out of the ruined place; and, as the morning mists had risen long ago when I first left the forge, so the evening mists were rising now, and in all the broad expanse of tranquil light they showed to me, I saw no shadow of another parting from her.

\*\*\* END OF THE PROJECT GUTENBERG EBOOK GREAT EXPECTATIONS \*\*\*

Updated editions will replace the previous one—the old editions will be renamed.

Creating the works from print editions not protected by U.S. copyright law means that no one owns a United States copyright in these works, so the Foundation (and you!) can copy and distribute it in the United States without permission and without paying copyright royalties. Special rules, set forth in the General Terms of Use part of this license, apply to copying and distributing Project Gutenberg™ electronic works to protect the PROJECT GUTENBERG™ concept and trademark. Project Gutenberg is a registered trademark, and may not be used if you charge for an eBook, except by following the terms of the trademark license, including paying royalties for use of the Project Gutenberg trademark. If you do not charge anything for copies of this eBook, complying with the trademark license is very easy. You may use this eBook for nearly any purpose such as creation of derivative works, reports, performances and research. Project Gutenberg eBooks may be modified and printed and given away—you may do practically ANYTHING in the United States with eBooks not protected by U.S. copyright law. Redistribution is subject to the trademark license, especially commercial redistribution.

#### START: FULL LICENSE

#### THE FULL PROJECT GUTENBERG LICENSE

PLEASE READ THIS BEFORE YOU DISTRIBUTE OR USE THIS WORK

To protect the Project Gutenberg™ mission of promoting the free distribution of electronic works, by using or distributing this work (or any other work associated in any way with the phrase "Project Gutenberg"), you agree to comply with all the terms of the Full Project Gutenberg™ License available with this file or online at [www.gutenberg.org/license](http://www.gutenberg.org/license).

#### Section 1. General Terms of Use and Redistributing Project Gutenberg™ electronic works

1.A. By reading or using any part of this Project Gutenberg™ electronic work, you indicate that you have read, understand, agree to and accept all the terms of this license and intellectual property (trademark/copyright) agreement. If you do not agree to abide by all the terms of this agreement, you must cease using and return or destroy all copies of Project Gutenberg™ electronic works in your possession. If you paid a fee for obtaining a copy of or access to a Project Gutenberg™ electronic work and you do not agree to be bound by the terms of this agreement, you may obtain a refund from the person or entity to whom you paid the fee as set forth in paragraph 1.E.8.

1.B. "Project Gutenberg" is a registered trademark. It may only be used on or associated in any way with an electronic work by people who agree to be bound by the terms of this agreement. There are a few things that you can do with most Project Gutenberg™ electronic works even without complying with the full terms of this agreement. See paragraph 1.C below. There are a lot of things you can do with Project Gutenberg™ electronic works if you follow the terms of this agreement and help preserve free future access to Project Gutenberg™ electronic works. See paragraph 1.E below.

1.C. The Project Gutenberg Literary Archive Foundation ("the Foundation" or PGLAF), owns a compilation copyright in the collection of Project Gutenberg™ electronic works. Nearly all the individual works in the collection are in the public domain in the United States. If an individual work is unprotected by copyright law in the United States and you are located in the United States, we do not claim a right to prevent you from copying, distributing, performing, displaying or creating derivative works based on the work as long as all references to Project Gutenberg are removed. Of course, we hope that you will support the Project Gutenberg™

# escrever cabeçalho do site

```
def write_header(source_html, title):
    header_tag = source_html.find('section', attrs={'id': 'pg-header'})

    span_tags = header_tag.find_all('span')
    # <span>
    # *** START OF THE PROJECT GUTENBERG EBOOK ALICE'S ADVENTURES IN WONDERLAND ***
    # </span>
    span_tag = span_tags[1]

    with open(f'books/{title}/README.txt', 'w') as file_header:
        header_text = header_tag.text
        # substituindo o separador de página por uma string vazia
        header_text= header_text.replace(str(span_tag.text), '').strip()

        file_header.write(header_text)
```

Módulo header

# Com separador de página

The Project Gutenberg eBook of Alice's Adventures in Wonderland

This ebook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this ebook or online at [www.gutenberg.org](http://www.gutenberg.org). If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Alice's Adventures in Wonderland

Author: Lewis Carroll

Release date: June 27, 2008 [eBook #11]

Most recently updated: March 30, 2021

Language: English

Credits: Arthur DiBianca and David Widger

\*\*\* START OF THE PROJECT GUTENBERG EBOOK ALICE'S ADVENTURES IN WONDERLAND \*\*\*

# Após retirar separador de página

The Project Gutenberg eBook of Alice's Adventures in Wonderland

This ebook is for the use of anyone anywhere in the United States and most other parts of the world at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this ebook or online at [www.gutenberg.org](http://www.gutenberg.org). If you are not located in the United States, you will have to check the laws of the country where you are located before using this eBook.

Title: Alice's Adventures in Wonderland

Author: Lewis Carroll

Release date: June 27, 2008 [eBook #11]

Most recently updated: March 30, 2021

Language: English

Credits: Arthur DiBianca and David Widger

# Módulo pdf

```
def create_pdf(source_html, domain, title):

    def remove_minimize_image(source_html:BeautifulSoup):
        pattern = re.compile(r'<img[^>]*?src="images/enlarge\.jpg"[^>]*?>')

        # Substituir as correspondências pela string vazia (remover as tags)
        source_html_string = re.sub(pattern, '', str(source_html))

    return source_html_string

    def remove_site_header_and_footer(source_html:BeautifulSoup):
        header_tag = source_html.find('section', attrs={'id':'pg-header'})
        footer_tag = source_html.find('section', attrs={'id':'pg-footer'})
        # retirando cabeçalho e footer da página
        source_html = str(source_html).replace(str(header_tag), '').replace(str(footer_tag), '')

    return source_html
```

# Módulo pdf

```
def write_domain_in_image_link(source_html:BeautifulSoup, domain:str):
    padrao_href_src = re.compile(r'(src)="\([^\"]+\)"')
    # Substituir texto entre aspas de atributos src pelo domínio + texto original
    web_page_string = padrao_href_src.sub(fr'\1="{domain}\2"', str(source_html))
    return web_page_string
```



# buscar caminho do wk

```
def get_wkhtmltopdf_config():
    if platform.system() == 'Windows':
        try:
            path_wkhtmltopdf = os.popen('where wkhtmltopdf').read()
            config = pdfkit.configuration(wkhtmltopdf=path_wkhtmltopdf)
        except:
            path_wkhtmltopdf = r'C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe'
            config = pdfkit.configuration(wkhtmltopdf=path_wkhtmltopdf)
    else:
        try:
            path_wkhtmltopdf = os.popen('which wkhtmltopdf').read()
            config = pdfkit.configuration(wkhtmltopdf=path_wkhtmltopdf)
        except:      You, anteontem • implementado hack de jvcgomes14@gmail.com na
                    path_wkhtmltopdf = '/usr/local/bin/wkhtmltopdf'
                    config = pdfkit.configuration(wkhtmltopdf=path_wkhtmltopdf)
                    break
```

# conversão para pdf

```
pdfkit.from_string(web_page_string, f'./books/{title}/{title}.pdf',
configuration=get_wkhtmltopdf_config(), options={
    'enable-local-file-access': None,
    'encoding': 'UTF-8',
    '--image-quality': 100
})
```

# PDF criado

5 de 275

- + Zoom automático

Great Expectations [1867 Edition] by Charles Dickens

Contents Chapter I. Chapter II. Chapter III. Chapter IV. Chapter V. Chapter VI. Chapter VII. Chapter VIII. Chapter IX. Chapter X. Chapter XI. Chapter XII. Chapter XIII. Chapter XIV. Chapter XV. Chapter XVI. Chapter XVII. Chapter XVIII. Chapter XIX. This is the end of the first stage of Pip's expectations. Chapter XX. Chapter XXI. Chapter XXII. Chapter XXIII.



"You young dog," said the man, licking his lips, "what fat cheeks you ha' got."

I believe they were fat, though I was at that time undersized for my years, and not strong.

"Darn me if I couldn't eat 'em," said the man, with a threatening shake of his head, "and if I han't half a mind to!"

I earnestly expressed my hope that he wouldn't, and held tighter to the tombstone on which he had put me; partly, to keep myself upon it; partly, to keep myself from crying.

"Now lookee here!" said the man. "Where's your mother?"

"There, sir!" said I.

He started, made a short run, and stopped and looked over his shoulder.

"There, sir!" I timidly explained. "Also Georgiana. That's my mother."

"Oh!" said he, coming back. "And is that your father alonger your mother?"

"Yes, sir," said I; "him too; late of this parish."

"Hal!" he muttered then, considering. "Who d'y'e live with,—supposin' you're kindly let to live, which I han't made up my mind about?"

Obrigado pela atenção!