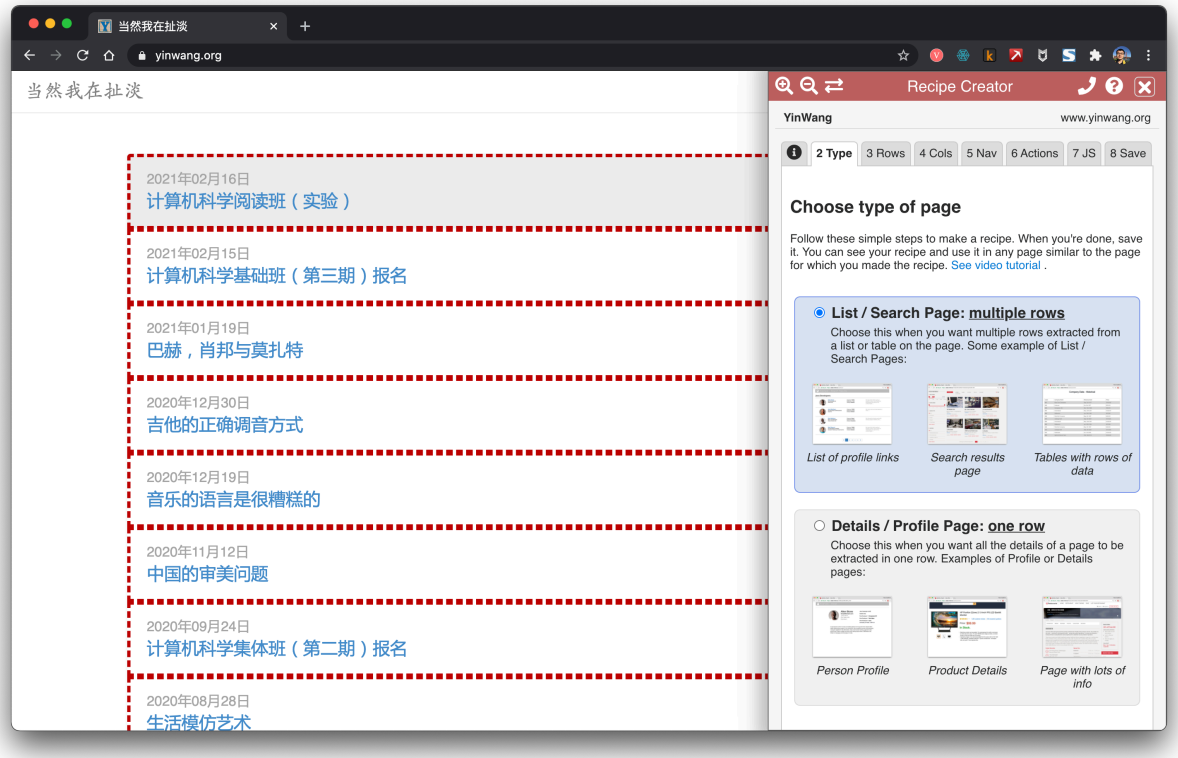
بلج ىوتحم عقوملا

إذا ذلك، وعم الإلكترونية. المواقع محتوى استخراج اهنكمي التي الجاهزة الأدوات من العديد بالفعل هناك ل قد خاصة، أو معقدة مواقع مع تتمكن لن اهاندمختسا إذا أفضل. بشكل الكامنة العملية فده من مكنتم بأنفسنا، الأدوات بناء إلى تحتاج هذه بالاستخدام المدرجة التنائج على الحصول أفضل بشكل لفهمها بأنفسنا، الأدوات بناء إلى تحتاج هذه بالاستخدام المدرجة التنائج على الحصول أفضل بشكل لفهمها واستخدامها بشكل أكثر فعالية.

النوع يقلن ظرة على عرض الأدوات الجاهزة المتاحة.

البيانات مُنقِّب



□□□□□□ 1: □□□□□

Data Miner وه ةفاضإ ةمئالم اذج ىلع حفصتم □□□□□□. نكمي همادختسا جراخت ساال طبارلا ىوتحملا بسهولة.

□□□□□□□

getbook وه ةاداً ةديفم اذج ءاشنإل بتكلا الإلكترونية.

```
pip install getbook
```

```
{
  "title": "   ",
  "author":   " "    ,
  "description":   " "    ,
  "language": "ar",
  "plugins": [
    "highlight",
    "search",
    "lunr"
  ]
}
```

```
{
  "uid": "book",
  "title":    " "    ,
  "author": "Armin",
  "chapters": [
    "http://lucumr.pocoo.org/2018/7/13/python/",
    "http://lucumr.pocoo.org/2017/6/5/diversity-in-technology",
  ]
}
```

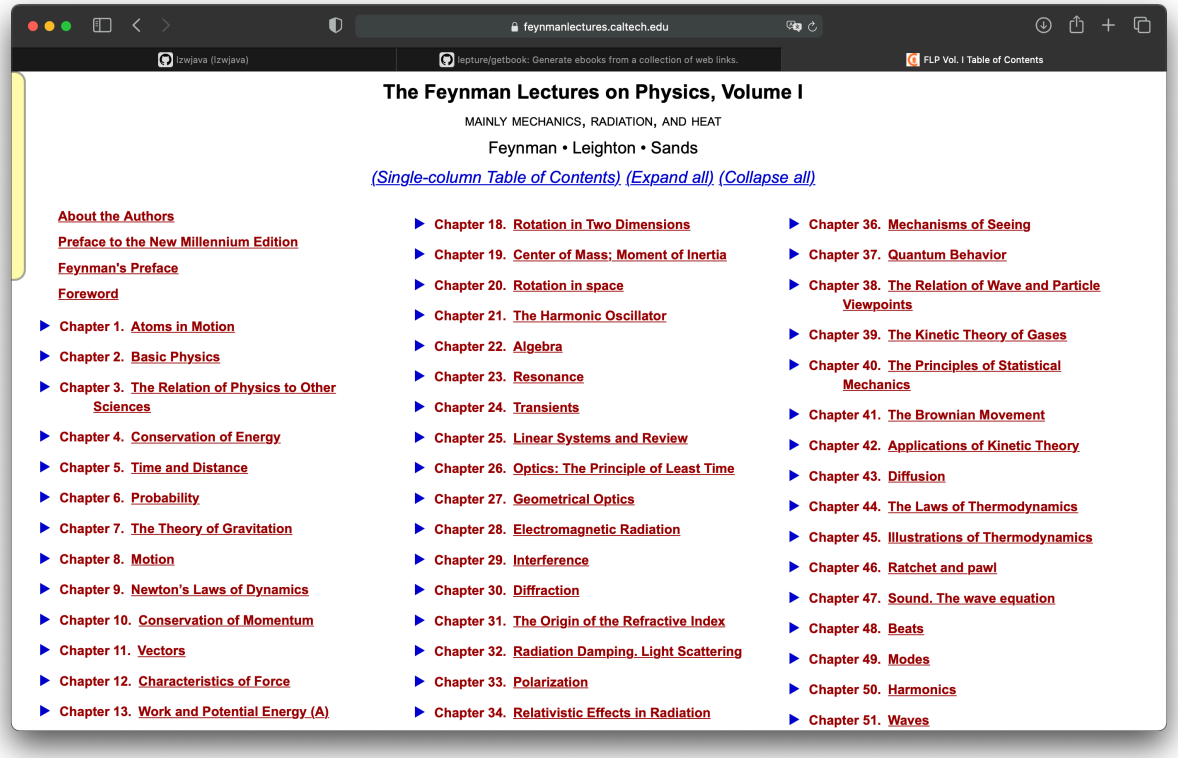
```
getbook -f ./book.json --mobi
```

حيث، getbook و Data Miner باستخدام إلكتروني. كتاب إلى بسهولة الروابط بعض تحويل تم الطريقة، بهذه بسهولة. إلكترونية كتب إنشاء يمكن إلكتروني، كتاب إلى بتحويلها والثاني الروابط بجمع الأول يقوم

فايرمان فيزياء محاضرات

تحويل يفيد المتعلمين إلكتروني، كتاب إلى فايرمان فيزياء محاضرات تحويل عملية: مشروع فصل في يفيد لن رى المشروع هذا في نستمر نحن إلكتروني. كتاب إلى mathjax باستخدام عرضها تم html وب صفحة تتكون من ثلاثة مجلدات. صفحات جميع على الحصول أعلاه الصورة المجلد الفهرس الأول. للمجلد

[illegible] بروتوكول عميل [illegible].[illegible]


feynmanlectures.caltech.edu
lzwjava (lzwjava)
lepture/getbook: Generate ebooks from a collection of web links.
FLP Vol. I Table of Contents
The Feynman Lectures on Physics, Volume I
MAINLY MECHANICS, RADIATION, AND HEAT
Feynman • Leighton • Sands
(Single-column Table of Contents) (Expand all) (Collapse all)
About the Authors
Preface to the New Millennium Edition
Feynman's Preface
Foreword
Chapter 1. Atoms in Motion
Chapter 2. Basic Physics
Chapter 3. The Relation of Physics to Other Sciences
Chapter 4. Conservation of Energy
Chapter 5. Time and Distance
Chapter 6. Probability
Chapter 7. The Theory of Gravitation
Chapter 8. Motion
Chapter 9. Newton's Laws of Dynamics
Chapter 10. Conservation of Momentum
Chapter 11. Vectors
Chapter 12. Characteristics of Force
Chapter 13. Work and Potential Energy (A)
Chapter 18. Rotation in Two Dimensions
Chapter 19. Center of Mass; Moment of Inertia
Chapter 20. Rotation in space
Chapter 21. The Harmonic Oscillator
Chapter 22. Algebra
Chapter 23. Resonance
Chapter 24. Transients
Chapter 25. Linear Systems and Review
Chapter 26. Optics: The Principle of Least Time
Chapter 27. Geometrical Optics
Chapter 28. Electromagnetic Radiation
Chapter 29. Interference
Chapter 30. Diffraction
Chapter 31. The Origin of the Refractive Index
Chapter 32. Radiation Damping. Light Scattering
Chapter 33. Polarization
Chapter 34. Relativistic Effects in Radiation
Chapter 36. Mechanisms of Seeing
Chapter 37. Quantum Behavior
Chapter 38. The Relation of Wave and Particle Viewpoints
Chapter 39. The Kinetic Theory of Gases
Chapter 40. The Principles of Statistical Mechanics
Chapter 41. The Brownian Movement
Chapter 42. Applications of Kinetic Theory
Chapter 43. Diffusion
Chapter 44. The Laws of Thermodynamics
Chapter 45. Illustrations of Thermodynamics
Chapter 46. Ratchet and pawl
Chapter 47. Sound. The wave equation
Chapter 48. Beats
Chapter 49. Modes
Chapter 50. Harmonics
Chapter 51. Waves

□□□□□□ 2: □□

الكود المصدري: [illegible].

استخدامه يتم لا عادةً و[illegible]. [illegible] بروتوكولي من العميل جانب تُنفذ فئات الوحدة هذا يُعرّف [illegible] تستخدم التي الروابط مع للتعامل [illegible] الوحدة يستخدمه حيث [illegible] مباشرةً و[illegible].

انظر أيضًا: يوصى باستخدام حزمة [illegible] لواجهة [illegible] ذات مستوى الأعلى.

أعلى. مستوى ذات برمجية واجهة هي requests أن الواضح من

```
import requests

def main():
    r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
    print(r.status_code)
```

```
401
```

```
import requests

def main():
    r = requests.get('https://github.com')
    print(r.status_code)
    print(r.text)
```

الرئيسية الوظيفة main()

```
```html
200
<html>
  ...
</html>
```

صحيح. بشكل تعمل requests واجهة أن وتبين جربت،

```
<div class="toc-chapter" id="C03">
   <span class="triangle">
```
```

```
 â ¶
</span>
<a class="chapterlink" href="javascript:Goto(1,3)">
 <span class="tag">
      .3
 </span>

</a>
<div class="sections">
 <a href="javascript:Goto(1,3,1)">
  <span class="tag">
   3-1
  </span>

 </a>
 <a href="javascript:Goto(1,3,2)">
  <span class="tag">
   3-2
  </span>

 </a>
 <a href="javascript:Goto(1,3,3)">
  <span class="tag">
   3-3
  </span>

 </a>
 <a href="javascript:Goto(1,3,4)">
  <span class="tag">
   3-4
  </span>

 </a>
 <a href="javascript:Goto(1,3,5)">
  <span class="tag">
   3-5
```

```
      </span>

     </a>
     <a href="javascript:Goto(1,3,6)">
      <span class="tag">
       3-6
      </span>

     </a>
     <a href="javascript:Goto(1,3,7)">
      <span class="tag">
       3-7
      </span>

     </a>
    </div>
</div>
```

هذا هو كود html للفصل الثالث من الفهرس. صفحة في الثالث أريد استخراج روابط كل فصل من هنا. <a href="javascript:Goto(1,3,7)'
يمكن ملاحظة أنه رابط javascript.

https://www.feynmanlectures.caltech.edu/I_03.html

الثالث الفصل إلى يشير I_03.html المثال، سبيل على للغاية. منتظمة كانت فصل لكل مسارات أن لاحظت ثم
الأول. المجلد من

```
import requests
from bs4 import BeautifulSoup
from multiprocessing import Process

def scrape(chapter):
    if chapter < 1 or chapter > 52:
        raise Exception(f'chapter {chapter}')
    chapter_str = '{:02d}'.format(chapter)
    url = f'https://www.feynmanlectures.caltech.edu/I_{chapter_str}.html'
    print(f'       }url}')
    r = requests.get(url)
```

```
    if r.status_code != 200:
        raise Exception(r.status_code)
    soup = BeautifulSoup(r.text, features='lxml')
    f = open(f'./chapters/I_{chapter_str}.html', 'w')
    f.write(soup.prettify())
    f.close()
```

ملاحظة: تمت ترجمة نص النزيل الإنجليزي إلى العربية مع الحفاظ على أسماء الدوال والمتغيرات كما هي.

```
def main():
    for i in range(52):
        p = Process(target=scrape, args=(i+1))
        p.start()
        p.join()

main()
```

لن تستخدم هذا الزخرف. كود كتابة في تستمر Process.

```
    raise RuntimeError('''
RuntimeError:

                        .

                           `fork`
                               :

if __name__ == '__main__':
    freeze_support()
    ...

              "freeze_support()"
                       .
```

```python
def main():
    for i in range(52):
        p = Process(target=scrape, args=(i+1,))
```

```
        p.start()
    p.join()
```

العربية: إلى الكود ترجمة

```
def main():
    for i in range(52):
        p = Process(target=scrape, args=(i+1,))
        p.start()
    p.join()
```

ملاحظة: الكود المكتوب بلغة □□□□□□، ولكن تمت ترجمة أسماء الدوال أو المتغيرات أو الأكواد إلى ما هو أقرب للبرمجية.

```
if __name__ == "__main__":
    main()
```

```
def main():
    start = timeit.default_timer()
    ps = [Process(target=scrape, args=(i+1,)) for i in range(52)]
    for p in ps:
        p.start()
    for p in ps:
        p.join()
    stop = timeit.default_timer()
    print(':    ', stop - start)

if __name__ == "__main__":
    main()
```

```
                https://www.feynmanlectures.caltech.edu/I_01.html
                https://www.feynmanlectures.caltech.edu/I_04.html
...
                https://www.feynmanlectures.caltech.edu/I_51.html
                https://www.feynmanlectures.caltech.edu/I_52.html
:     9.144841699
```

the most information in the fewest words? I believe it is the *atomic hypothesis* (or the atomic *fact* , or whatever you wish to call it) that *all things are made of atomsâlittle particles that move around in perpetual motion, attracting each other when they are a little distance apart, but repelling upon being squeezed into one another* . In that one sentence, you will see, there is an *enormous* amount of information about the world, if just a little imagination and thinking are applied.

Figure 1â1

To illustrate the power of the atomic idea, suppose that we have a drop of water a quarter of an inch on the side. If we look at it very closely we see nothing but waterâsmooth, continuous water. Even if we magnify it with the best optical microscope availableâroughly two thousand timesâthen the water drop will be roughly forty feet across, about as big as a large room, and if we looked rather closely, we would *still* see relatively smooth waterâbut here and there small football-shaped things swimming back and forth. Very interesting. These are paramecia. You may stop at this

□□□□□□ 3: ﺷﻜﻞ

```
<div class="figure" id="Ch1-F1">
        <img src="img/FLP_I/f01-01/f01-01_tc_big.svgz">
        <div class="caption empty">
         <span class="tag">
              1â 1
         </span>
        </div>
</div>
```

```
import requests
from bs4 import BeautifulSoup
from multiprocessing import Process
import timeit

def scrape(chapter):
    if chapter < 1 or chapter > 52:
        raise Exception(f'chapter {chapter}')
    chapter_str = '{:02d}'.format(chapter)
    url = f'https://www.feynmanlectures.caltech.edu/I_{chapter_str}.html'
    print(f'               }url}')
```

```
    r = requests.get(url)
    if r.status_code != 200:
        raise Exception(r.status_code)
    soup = BeautifulSoup(r.text, features='lxml')
    f = open(f'./chapters/I_{chapter_str}.html', 'w')
    f.write(soup.prettify())
    f.close()
```

ملاحظة: تمت ترجمة النص الإنجليزي إلى العربية مع الحفاظ على أسماء الدوال والمتغيرات كما هي.

```
def main():
    start = timeit.default_timer()
    ps = [Process(target=scrape, args=(i+1,)) for i in range(52)]
    for p in ps:
        p.start()
    for p in ps:
        p.join()
    stop = timeit.default_timer()
    print(':    ', stop - start)

if __name__ == "__main__":
    main()
```

انظر إلى الرابط.

```
    imgs = soup.find_all('img')
    for img in imgs:
        print(img)
```

ملاحظة: الكود أعلاه مكتوب بلغة [illegible] مدخت [illegible] لاستخراج جميع الصور من جزء الإنجليزية المكتوبة والمتغيرات الأوامر أن حيث ترجمة إلى الكود يحتاج إلى ويب. صفحة من <img> البرمجة. لغة

```
                https://www.feynmanlectures.caltech.edu/I_01.html
<img id="TwitLink" src=""/>
<img id="FBLink" src=""/>
<img id="MailLink" src=""/>
```

```
<img id="MobileLink" src=""/>
<img id="DarkModeLink" src=""/>
<img id="DesktopLink" src=""/>
<img src="img/camera.svg"/>
<img src="img/FLP_I/f01-00/f01-00.jpg"/>
<img data-src="img/FLP_I/f01-01/f01-01_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-02/f01-02_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-03/f01-03_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-04/f01-04_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-05/f01-05_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-06/f01-06_tc_big.svgz"/>
<img class="first" data-src="img/FLP_I/f01-07/f01-07_tc_iPad_big_a.svgz"/>
<img class="last" data-src="img/FLP_I/f01-07/f01-07_tc_iPad_big_b.svgz"/>
<img data-src="img/FLP_I/f01-08/f01-08_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-09/f01-09_tc_big.svgz"/>
<img data-src="img/FLP_I/f01-10/f01-10_tc_big.svgz"/>
```

[illegible]

سيل كيدل نذإ لوصول للى إذه .المورد

[illegible] 2.4.38 [illegible] مدخلا على [illegible] ذفنملا 443

```shell
% pip install selenium
      selenium
          selenium-3.141.0-py2.py3-none-any.whl         904) (
           urllib3       /usr/local/lib/python3.9/site-packages   ) selenium) (1.24.2)
        :     selenium
      selenium-3.141.0

export CHROME_DRIVER_HOME=$HOME/dev-env/chromedriver
export PATH="${PATH}:${CHROME_DRIVER_HOME}"
```

ترجمة:

```
export CHROME_DRIVER_HOME=$HOME/dev-env/chromedriver
export PATH="${PATH}:${CHROME_DRIVER_HOME}"
```

ملاحظة: يجب والمتغيرات الأوامر أن حيث عادةً، ترجمته يتم ولا البرمجي نص هو أعلاه النص صحيح. بشكل للعمل هي كما تبقى أن

```
% chromedriver -h
:      chromedriver [   ]


  --port=PORT
  --adb-port=PORT                       adb
  --log-path=FILE                                stderr               INFO
  --log-level=LEVEL                      :    ALL, DEBUG, INFO, WARNING, SEVERE, OFF
  --verbose                                    )   --log-level=ALL)
  --silent                                      )   --log-level=OFF)
  --append-log
  --replayable                     (   )                                          .
  --version
  --url-base                              URL                     wd/url
  --readable-timestamp
  --enable-chrome-logs                                )          (
  --allowed-ips                            IP                     ChromeDriver

#
```

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support.expected_conditions import presence_of_element_located
```

تشمل: المكتبات هذه الويب. متصفح مع للتعامل selenium من اللازمة المكتبات استيراد تم

- webdriver: المتصفح. في للتحكم
- By: ديد للعناصر في الصفحة طرق باستخدام مثل الـ أو الاسم أو الـ.
- Keys: الإرسال لضغطات على لوحة المفاتيح مثل أو.

- WebDriverWait: انتظار للحتى يتم تحميل عنصر معين في الصفحة.
- presence_of_element_located: تأكد من وجود عنصر معين في الصفحة قبل المتابعة.

```
with webdriver.Chrome() as driver:
    wait = WebDriverWait(driver, 10)
    driver.get("https://google.com/ncr")
    driver.find_element(By.NAME, "q").send_keys("cheese" + Keys.RETURN)
    first_result = wait.until(presence_of_element_located((By.CSS_SELECTOR, "h3>div")))
    print(first_result.get_attribute("textContent"))
```

```
 )                              (
```

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support.expected_conditions import presence_of_element_located
import urllib

def main():
    driver = webdriver.Chrome()
    wait = WebDriverWait(driver, 10)
    driver.get("https://www.feynmanlectures.caltech.edu/I_01.html")
    elements = driver.find_elements(By.TAG_NAME, "img")
    # print(dir(elements[0]))
    print(driver.page_source)
    i = 0
    for element in elements:
        # src = element.get_attribute('src')
        element.screenshot(f'images/{i}.png')
        i +=1
    driver.close()
main()

from bs4 import BeautifulSoup
from multiprocessing import Process
import timeit
```

```
from pathlib import Path
from selenium import webdriver
from selenium.webdriver.common.by import By

def img_path(chapter):
    return f'./chapters/{chapter}/img'

def img_name(url):
    splits = url.split('/')
    last = splits[len(splits) - 1]
    parts = last.split('.')
    name = parts[0]
    return name
```

العربية: إلى الكود ترجمة

```
def img_name(url):
    #                               '/'
    splits = url.split('/')

    #
    last = splits[len(splits) - 1]

    #                         '.'
    parts = last.split('.')

    #                          )        (
    name = parts[0]

    #
    return name
```

هذا الكود يستخرج اسم الملف من الرابط https://example.com/images/ph□□□ دون امتداد. على سبيل المثال، إذا كان الرابط هو الرابط الدالة فإن سترجع photo.

```
def download_images(driver: webdriver.Chrome, chapter):
    path = img_path(chapter)
```

```
    Path(path).mkdir(parents=True, exist_ok=True)

    elements = driver.find_elements(By.TAG_NAME, "img")
    for element in elements:
        src = element.get_attribute('src')
        name = img_name(src)
        element.screenshot(f'{path}/{name}.png')
```

إلى: أعلاه الكود ترجمة تمت

```
def download_images(driver: webdriver.Chrome, chapter):
    path = img_path(chapter)
    Path(path).mkdir(parents=True, exist_ok=True)

    elements = driver.find_elements(By.TAG_NAME, "img")
    for element in elements:
        src = element.get_attribute('src')
        name = img_name(src)
        element.screenshot(f'{path}/{name}.png')
```

:ةظحالم تم ظافحلا ىلع ءامسأ الدوال تارييغتملا واهب ةيجمرب ألا ءامسأ ل هنأ هي كما ولي متي .هتمجرت

[illegible] 5.0 [illegible] 10_15_6 [illegible] 605.1.15 [illegible], [illegible] 14.0.3 [illegible] 605.1.15

```
def scrape(chapter):
    if chapter < 1 or chapter > 52:
        raise Exception(f'chapter {chapter}')
    chapter_str = '{:02d}'.format(chapter)
    url = f'https://www.feynmanlectures.caltech.edu/I_{chapter_str}.html'
    driver = webdriver.Chrome()
    driver.get(url)
    page_source = driver.page_source
    Path(f'./chapters/{chapter_str}').mkdir(parents=True, exist_ok=True)
    print(f'scraping {url}')

    download_images(driver, chapter_str)
```

```
    soup = BeautifulSoup(page_source, features='lxml')
    imgs = soup.find_all('img')
    for img in imgs:
        if 'src' in img.attrs or 'data-src' in img.attrs:
            src = ''
            if 'src' in img.attrs:
                src = img.attrs['src']
            elif 'data-src' in img.attrs:
                src = img.attrs['data-src']
                del img.attrs['data-src']
            name = img_name(src)
            img.attrs['src'] = f'img/{name}.png'

    f = open(f'./chapters/{chapter_str}/I_{chapter_str}.html', 'w')
    f.write(soup.prettify())
    f.close()

    driver.close()
```

العربية: إلى الكود ترجمة

```
def scrape(chapter):
    if chapter < 1 or chapter > 52:
        raise Exception(f'chapter {chapter}')
    chapter_str = '{:02d}'.format(chapter)
    url = f'https://www.feynmanlectures.caltech.edu/I_{chapter_str}.html'
    driver = webdriver.Chrome()
    driver.get(url)
    page_source = driver.page_source
    Path(f'./chapters/{chapter_str}').mkdir(parents=True, exist_ok=True)
    print(f'scraping {url}')

    download_images(driver, chapter_str)

    soup = BeautifulSoup(page_source, features='lxml')
    imgs = soup.find_all('img')
    for img in imgs:
```

```
        if 'src' in img.attrs or 'data-src' in img.attrs:
            src = ''
            if 'src' in img.attrs:
                src = img.attrs['src']
            elif 'data-src' in img.attrs:
                src = img.attrs['data-src']
                del img.attrs['data-src']
            name = img_name(src)
            img.attrs['src'] = f'img/{name}.png'

    f = open(f'./chapters/{chapter_str}/I_{chapter_str}.html', 'w')
    f.write(soup.prettify())
    f.close()

    driver.close()
```

الكود: شرح

- تم تعريف دالة scrape التي تأخذ رقم الفصل المراد كشطه.
- تم التحقق إذا ما كان رقم الفصل بين 1 و52، وإذا لم يكن، فذلك خطأ. إرجاع تم تلقائيًا.
- تم تحويل رقم الفصل إلى سلسلة نصية من رقمين، مثل 01 للفصل الأول.
- تم إنشاء رابط بناءً على رقم الفصل.
- تم فتح متصفح باستخدام webdriver.Chrome().
- تم التوجه إلى الرابط المحدد باستخدام driver.get(url).
- تم حفظ مصدر الصفحة في المتغير page_source.
- تم إنشاء مجلد لحفظ الفصل إذا لم يكن موجودًا بالفعل.
- تم تعيين الرسالة الطباعية لتفيد بأنه تم الانتهاء من البيانات الرابط.
- تم استدعاء دالة download_images لتنزيل الصور.
- تم تحليل مصدر الصفحة باستخدام BeautifulSoup.
- تم البحث عن جميع الوسوم <img> في الصفحة.
- تم تعديل مسار الصور بحيث تشير إلى الصور المحفوظة محليًا.
- تم حفظ الصفحة المعدلة في ملف.
- تم إغلاق المتصفح.

```
def main():
    start = timeit.default_timer()
    ps = [Process(target=scrape, args=(i+1,)) for i in range(2)]
```

```
    for p in ps:
        p.start()
    for p in ps:
        p.join()
    stop = timeit.default_timer()
    print(':    ', stop - start)

if __name__ == "__main__":
    main()

                https://www.feynmanlectures.caltech.edu/I_01.html
                https://www.feynmanlectures.caltech.edu/I_02.html
:    21.478510914999998

 errpipe_read, errpipe_write = os.pipe()
OSError: [Errno 24]

% ulimit a
ulimit:      :   a
lzw@lzwjava feynman-lectures-mobi % ulimit -a
-t:         (     )
-f:        (    )
-d:             (      )
-s:         (      )            8192
-c:          core (    )           0
-v:           (      )
-l:              (      )
-u:                                  2784
-n:                                 256

12
   -
12
mathjax2svg
latexs 128
svg_    0
svg_    0
```

```
svg_    1
svg_    1
svg_    2
svg_    2
svg_    3
svg_    3


12

  -
12
mathjax2svg
latexs 0
latexs 0

:     11.369145162

% grep --include=\*.html -r '\$' *
43/I_43.html:       $T$                 $N$    .
43/I_43.html:               .$T$
43/I_43.html:               $tau\/1$
43/I_43.html:$\tau$           .    $tau\$
43/I_43.html:        $60$    $\tau$       .
43/I_43.html:  $\tau$    ) (

  E21018:          Mobi                  .    :     <In earlier chapters> :    /private/var/folders/_3/
     W28001:    Kindle         CSS        .              CSS: 'max-width' :    /private/var/folders/_3/n
     W29004:                    :   <span amzn-src-id="985"> :    /private/var/folders/_3/n3b7dq8x6652d
     W29004:                    :   <p amzn-src-id="975"> :    /private/var/folders/_3/n3b7dq8x6652drmx

     W14001:                         :    /private/var/folders/_3/n3b7dq8x6652drmx6_d3t3bh0000gr/T/97c9
     W14001:                         :    /private/var/folders/_3/n3b7dq8x6652drmx6_d3t3bh0000gr/T/97c9
     W14001:                         :    /private/var/folders/_3/n3b7dq8x6652drmx6_d3t3bh0000gr/T/97c9

<span class="disabled" href="#Ch1-F1">
          1 1
</span>
```

```
    'OEBPS/84b8b4179175f097be1180a10089107be75d7d85.svg'              1264x1011
    'OEBPS/23a4df37f269c8ed43f54753eb838b29cff538a1.svg'              1264x259
Traceback (most recent call last):
  File "runpy.py", line 194, in _run_module_as_main
  File "runpy.py", line 87, in _run_code
  File "site.py", line 39, in <module>
  File "site.py", line 35, in main
  File "calibre/utils/ipc/worker.py", line 216, in main
  File "calibre/gui2/convert/gui_conversion.py", line 41, in gui_convert_override
  File "calibre/gui2/convert/gui_conversion.py", line 28, in gui_convert
  File "calibre/ebooks/conversion/plumber.py", line 1274, in run
  File "calibre/ebooks/conversion/plugins/mobi_output.py", line 214, in convert
  File "calibre/ebooks/conversion/plugins/mobi_output.py", line 237, in write_mobi
  File "calibre/ebooks/oeb/transforms/rasterize.py", line 55, in __call__
  File "calibre/ebooks/oeb/transforms/rasterize.py", line 142, in rasterize_spine
  File "calibre/ebooks/oeb/transforms/rasterize.py", line 152, in rasterize_item
  File "calibre/ebooks/oeb/transforms/rasterize.py", line 185, in rasterize_external
  File "calibre/ebooks/oeb/base.py", line 1092, in bytes_representation
  File "calibre/ebooks/oeb/base.py", line 432, in serialize
TypeError:                  'NoneType'

% kindlepreviewer feynman-lectures-on-physics-volumn-1.epub -convert
              .
        .
       1/1 .(  )
               !     : /Users/lzw/projects/feynman-lectures-mobi/feynman-lectures-on-physics-volumn-1.ep
        .
             /       /Users/lzw/projects/feynman-lectures-mobi/output

        ...
                        ...
       AZW3...
    ...
                              ...
       KF8
           KF8...
                aid           .              .
```

aid . .

aid . .

...

...