中學生Python解難 培訓教材

著作: 周艾迪 版權所有

## 一.系統環境

|  |  |  |
| --- | --- | --- |
| 系統環境 | Windows 10 |  |
| Python 版本 | Python 3.9.6 |  |
| 提供套件： | 提供的套件：Python 內置 Modules，及以下：  Requests  Untangle  BeautifulSoup4  Selenium | Numpy  Pandas  MatPlotLib  python-docx,  openpyxl |

## 二. 解難範圍(資料來源查看ref)

|  |  |  |
| --- | --- | --- |
| 1. 工作管理  a. 保持工作環境安全  b. 工作排程及重要性安排  c. 檔案有效管理 | 2. 溝通及客戶要求理解  a. 理解項目需求  b. 對代碼撰寫文檔  c. 以符合客戶要求 | 3. 編程開發邏輯  a. 開發邏輯  b. 應用內置套件  c. 應用適當的數據結構 |
| 4. 辦公自動化  a. 文件及資料夾自動化  b. 辦公文檔自動化  c. 辦公試算表自動化 | 5. 數據處理分析  a. 試算表及 CSV 數據整合  b. 數據篩選  c. 應用 Numpy  d. 應用 Pandas | 6. 網絡數據下載及取得  a. 應用 Requests  b. 應用 BeautifulSoup  c. 應用 Untangle  d. 應用 Selenium 及 Web Driver  e. 處理 XML/JSON 數據 |
| 7. 數據圖表繪製  a. 應用 MatplotLib  b. 對處理後數據進行圖表繪製  c. 能於一張圖繪製多張圖表  d. 能輸出圖 |  |  |

## 三. 知識點詳述 (源碼 code/ )

## 3.1 系統時間

|  |
| --- |
| import datetime  today\_date = datetime.date.today()  ''' today in YYYY-MM-DD format.'''  print(today\_date.isoformat()) |

## 3.2 文字檔案

|  |
| --- |
| #code/e3\_2.py 寫文字檔:  def out\_txt(msg):      print(msg)      with open(f"./out.txt", 'a',encoding='utf-8') as the\_file:          the\_file.write(str(msg))          the\_file.write('\n')  out\_txt("a")  out\_txt("b")  out\_txt("c") |

## 3.3 文件及資料夾

|  |
| --- |
| #code/e3\_3.py 查詢資料夾中所有文件,並只查詢JPEG檔:  from os import listdir  mypath = "./in/"  files = listdir(mypath)  for f in files:      if ".jpg" in f and '-' in f:         print(f) |

## 3.4 分類,使用資料結構 Dict {} 及 List []

|  |  |
| --- | --- |
|  |  |
| #code/e3\_4.py 資料分類:  files ="""1001-a.png  1001-b.jpeg  1001-c.jpg  21444-a.jpg  21444-b.jpg""".split("\n")  category={}  for f in files:      if ".jpg" in f  and '-' in f:          key , val =f.split('-')          if key in category:              category[key].append( f )          else:              category[key]=[f]  print(category) | |

## 3.5 python-docx辦公文檔自動化

|  |
| --- |
| from docx import Document  from docx.shared import Inches  document = Document()  document.add\_heading('Hello World!', 0)  document.save('demo.docx') |

## 3.6 request

|  |
| --- |
| import requests  \_url=f" https://xml.smg.gov.mo/c\_actual\_brief.xml "  \_res = requests.get(\_url)  \_res.encoding = "utf-8"  print(\_res.text) |

## 3.7 beautifulSoup

|  |
| --- |
| from bs4 import BeautifulSoup  import requests  \_url=f"http://www.macaodaily.com/html/2020-12/20/node\_1.htm"  \_res = requests.get(\_url)  \_res.encoding = "utf-8" #\_res.text 完整HTML內容, 包含標籤指今文字, 不是直觀宜懂  soup = BeautifulSoup(\_res.text, "html.parser")  links = soup.select("#all\_article\_list a") #需學習CSS設定篩選選擇條件,篩選出新聞標題  for link in links:      print(link.getText() ) |

## 3.8 pandas

|  |
| --- |
| import numpy as np  import pandas as pd  df = pd.DataFrame(  {  "Name": ["Braund, Mr. Owen Harris","Allen, Mr. William Henry","Bonnell, Miss. Elizabeth",],  "Age": [22, 35, 58],  "Sex": ["male", "male", "female"],  }  )  print(df.head())  -------------------------------------------  **Out[3]:**  Name Age Sex  0 Braund, Mr. Owen Harris 22 male  1 Allen, Mr. William Henry 35 male  2 Bonnell, Miss. Elizabeth 58 female |

# 3.9 MatPlotLib (Grouped bar chart with labels)

|  |
| --- |
| **import** **matplotlib.pyplot** **as** **plt**  **import** **numpy** **as** **np**  [labels](https://docs.python.org/3/library/stdtypes.html#list) = ['G1', 'G2', 'G3', 'G4', 'G5']  [men\_means](https://docs.python.org/3/library/stdtypes.html#list) = [20, 34, 30, 35, 27]  [women\_means](https://docs.python.org/3/library/stdtypes.html#list) = [25, 32, 34, 20, 25]  [x](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html#numpy.ndarray) = [np.arange](https://docs.scipy.org/doc/numpy/reference/generated/numpy.arange.html#numpy.arange)(len([labels](https://docs.python.org/3/library/stdtypes.html#list))) *# the label locations*  [width](https://docs.python.org/3/library/functions.html#float) = 0.35 *# the width of the bars*  [fig](https://matplotlib.org/stable/api/figure_api.html#matplotlib.figure.Figure), ax = [plt.subplots](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.subplots.html#matplotlib.pyplot.subplots)()  [rects1](https://matplotlib.org/stable/api/container_api.html#matplotlib.container.BarContainer) = [ax.bar](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.bar.html#matplotlib.axes.Axes.bar)([x](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html#numpy.ndarray) - [width](https://docs.python.org/3/library/functions.html#float)/2, [men\_means](https://docs.python.org/3/library/stdtypes.html#list), [width](https://docs.python.org/3/library/functions.html#float), label='Men')  [rects2](https://matplotlib.org/stable/api/container_api.html#matplotlib.container.BarContainer) = [ax.bar](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.bar.html#matplotlib.axes.Axes.bar)([x](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html#numpy.ndarray) + [width](https://docs.python.org/3/library/functions.html#float)/2, [women\_means](https://docs.python.org/3/library/stdtypes.html#list), [width](https://docs.python.org/3/library/functions.html#float), label='Women')  *# Add some text for labels, title and custom x-axis tick labels, etc.*  [ax.set\_ylabel](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.set_ylabel.html#matplotlib.axes.Axes.set_ylabel)('Scores')  [ax.set\_title](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.set_title.html#matplotlib.axes.Axes.set_title)('Scores by group and gender')  [ax.set\_xticks](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.set_xticks.html#matplotlib.axes.Axes.set_xticks)([x](https://docs.scipy.org/doc/numpy/reference/generated/numpy.ndarray.html#numpy.ndarray))  [ax.set\_xticklabels](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.set_xticklabels.html#matplotlib.axes.Axes.set_xticklabels)([labels](https://docs.python.org/3/library/stdtypes.html#list))  [ax.legend](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.legend.html#matplotlib.axes.Axes.legend)()  [fig.tight\_layout](https://matplotlib.org/stable/api/figure_api.html#matplotlib.figure.Figure.tight_layout)()  [plt.show](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html#matplotlib.pyplot.show)() Scores by group and gender |

## 四. 例題(資料來源查看ref)

### 例一: 圖像檔案分類

輸入: 多個圖像檔案

輸出: 多個含有圖片的Word.docx 檔案

描述: 現有以下圖片檔案, 分類並存放不同的Word.docx檔,

* . 1001-.a.png
* . 1001-b.jpeg
* . 1001-c.jpg
* . 21444-a.jpg
* . 21444-b.jpg

輸出檔:

* 1001.docx
* 21444.docx

### 例二: 網站截取:

輸入: 新聞網站

輸出: 一個TXT檔及多個Word.docx

描述: 現有<http://localhost/example-1/news.php> 希望能截取最新的內容, 一個txt記錄所有標題。另外每則新聞標題及文字內容則放到一個獨立Word.docx檔案中, 一則新聞一個檔. 新聞標題為檔名.

### **例三：訪客統計**

輸入：多個 csv 檔案

輸出：一個 Excel .xlsx 檔案及一張圖表圖像

1. 描述：現在有多個 csv 檔案，分別以 YYYY-MM-DD.csv 命名，每個 csv 檔案包括當 天的顧客每小時訪客量。現在生成一個 Excel 報表，第一欄為日期，第二欄 為每日訪客總量，及第三欄為訪客總量三日移動平均值。 另外，希望輸出一個圖表，同時顯示每日訪客總量棒型圖及三日移動平均 線，並設有圖例、標題、軸描述

### 四之一. 第一題ex1/ex1\_4: 圖像檔案分類

|  |
| --- |
| from os import listdir  def out\_docx(mypath,filename,jpeglist):      from docx import Document      document = Document()      document.add\_heading('Document Title', 0)      for f in jpeglist:          pic\_path=f"{mypath}{f}"          document.add\_picture(pic\_path)      document.save('./out/'+filename)  mypath = "./in/"  files = listdir(mypath)  category={}  for f in files:      if ".jpg" in f  and '-' in f:          key , x =f.split('-')          if key in category:              category[key].append( f )          else:              category[key]=[f]  for k in category:      filename =f"{k}.docx"      print(filename)      print(category[k])      out\_docx(mypath,filename,category[k]) |

### 四之二: 第二題:ex2/ex2\_2新聞網站1

|  |
| --- |
| # BeautifulSoup & requests  # soup.select("#all\_article\_list a")  #<div class="list" id="all\_article\_list">  #<ul>  #<li><span class="default"><a href="content\_1483.htm">苗無現嚴重不良反應</a></span></li>  #<li><span class="default"><a href="content\_1485.htm">衛健委：仍要戴口罩</a></span></li>  #</ul>  #</div>  def out\_txt(msg):      with open(f"./out/news\_title.txt", 'a',encoding='utf-8') as the\_file:          the\_file.write(str(msg))          the\_file.write('\n')  from bs4 import BeautifulSoup  import requests  \_url=f"http://www.macaodaily.com/html/2020-12/20/node\_1.htm"  title\_res = requests.get(\_url)  title\_res.encoding = "utf-8"  soup = BeautifulSoup(title\_res.text, "html.parser")  links = soup.select("#all\_article\_list a")  for link in links:      news\_title = link.getText()      out\_txt(news\_title)  print("finished .") |

### 四之三: 第二題ex2/ex2\_3新聞網站2

|  |
| --- |
| #<div class="small-8 cell infoContentContainer">  #  <h5 class="captionSize">  #     <a href="https://www.gcs.gov.mo/detail/zh-hant/">  #       <span class="infoSubject">教青局向教育界送上節日祝福</span>  #     </a>  #  </h5>  #  <div class="dateNSourceContainer hide-for-small-only grid-x">  #    <div class="sourceLabel shrink cell" title="教育及青年發展局">教育及青年發展局</div>  #    <div class="render\_timeago\_css" datetime="2021-09-10" data-tid="5">昨天</div>  #  </div>  #  <div class="baseContent baseSize hide-for-small-only fade"  #      style="max-height:4.5rem;">  #     教育暨青年局與高等教育局於今年初合併為教育及青年發展局後，秉持特區政府“優化教育及人  #     才培養工作”、“深化師資隊伍建設”的施政方針，持續革新和優化教師隊伍建設。9  #  </div>  #</div>  def out\_txt(msg):      with open(f"./out/news\_title.txt", 'a',encoding='utf-8') as the\_file:          the\_file.write(str(msg))          the\_file.write('\n')  def out\_docx(filename,context):      from docx import Document      document = Document()      document.add\_heading(filename, 0)      document.add\_paragraph( context, style='List Bullet' )      document.save("./out/"+filename+".docx")  from bs4 import BeautifulSoup  import requests  \_url=f"https://www.gcs.gov.mo/list/zh-hant/news/%E6%95%99%E8%82%B2%E9%AB%94%E8%82%B2?7"  title\_res = requests.get(\_url)  title\_res.encoding = "utf-8"  soup = BeautifulSoup(title\_res.text, "html.parser")  links = soup.select('[class="small-8 cell infoContentContainer"]')  for link in links:      news\_title=link.select("h5.captionSize")[0].getText().replace("'","").replace('"',"").replace("\n","")      news\_context = link.getText()      out\_txt(news\_title)      out\_docx(news\_title,news\_context)  print("finished .") |

### 四之四: 訪客統計ex3/ex3\_1.py

|  |
| --- |
| from os import listdir  import pandas as pd  mypath = "./in/"  files = listdir(mypath)  data\_date=[]  data\_customer=[]  data\_average=[]  for f in files:      if ".csv" in f  :          print(f)          df = pd.read\_csv(f'{mypath}{f}', delimiter=',')          data\_date.append(f.replace(".csv",""))          data\_customer.append(df["customer"].sum())          data\_average.append(0)  for idx,val in enumerate(data\_customer):      if idx==0:          pass      elif idx==len(data\_customer)-1:          pass      else:          data\_average[idx]=int((data\_customer[idx-1]+val+data\_customer[idx+1])/3)  print(data\_date)  print(data\_customer)  print(data\_average)  d = {'date': data\_date, 'total\_customer': data\_customer,'Ave':data\_average }  total\_df = pd.DataFrame(data=d)  total\_df.to\_excel("out/output.xlsx")  import matplotlib.pyplot as plt  import numpy as np  x = np.arange(len(data\_date))  # the label locations  width = 0.35  # the width of the bars  fig, ax = plt.subplots()  rects1 = ax.bar(x - width/2, data\_customer, width, label='customer')  rects2 = ax.bar(x + width/2, data\_average, width, label='3 days average')  # Add some text for labels, title and custom x-axis tick labels, etc.  ax.set\_ylabel('customer')  ax.set\_title('customer / pre day')  ax.set\_xticks(x)  ax.set\_xticklabels(data\_date)  ax.legend()  plt.xticks(rotation=30)  fig.tight\_layout()  plt.savefig("out/out.png") |

## 五: 網頁HTML 及 CSS(超文本,多媒體,超連結)

## diner/run.bat

|  |
| --- |
| explorer <http://localhost:8000/index.html>  explorer <http://localhost:8000/news.html>  explorer http://localhost:8000/ex1.html  explorer <http://localhost:8000/ex2.html>  explorer <http://localhost:8000/ex3.html>  explorer http://localhost:8000/ex4.html  python -m http.server --directory www/ |

##### HTML (diner/www/ex3.html)

<html>

<head>

</head>

<body>

Coding here

</body>

</html>

標籤<tag> :

<htm>,<body>

文本編輯標籤

<h1>..<h5>

<p>文字段落</p>

<div>文字段落</div>

<br>

<img src=url>

<a href=url>link</a>

##### CSS (diner/www/ex4.html)

<style>

div.news\_item{}

h5.title{color:white}

div.contxt{ color:white }

</style>

<div id=id01 class=”news\_item”>

<h5 class=”tilte”>dinner is ready!</h5>

<div class=”context”>

Dinner is <br> ready!

</div>

</div>

## 六: CSS 選擇練習, diner/run.bat

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Learn CSS Selector https://flukeout.github.io     |  |  | | --- | --- | | 01A | plate | | 02A | bente | | 03#id | #fancy | | 04A B | plate apple | | 05#id A | #fancy pickle | | 06.classname | .small | | 07A.className | orange.small | | 08Put your back into it! | bento orange.small | | 09A, B | plate ,bento | | 10\* | \* | | 11A \* | plate \* | | 12A + B | plate+ apple | | 13A ~ B | bento~pickle | | 14A > B | plate>appple | | 15:first-child | orange:first-child | | 16:only-child | plat \*:only-child | | 17:last-child | apple.small, pickle | | 18:nth-child(A) | plate:nth-child(3) | | 19:nth-last-child(A) | bento:nth-last-child(2) | | 20:first-of-type | apple:first-of-type | | 21:nth-of-type(A) | plate:nth-of-type(even) | | 22:nth-of-type(An+B) | plate:nth-of-type(2n+3) | | 23:only-of-type | plate apple:only-of-type | | 24:last-of-type | :last-of-type | | 25:empty | bento:empty | | 26:not(X) | apple:not(.small) | | 27[attribute] | \*[for] | | 28A[attribute] | plate[for] | | 29[attribute="value"] | [for="Vitaly"] | | 30[attribute^="value"] | [for^="Sa"] | | 31[attribute$="value"] | [for$="to"] | | 32[attribute\*="value"] | [for\*="bb"] | |