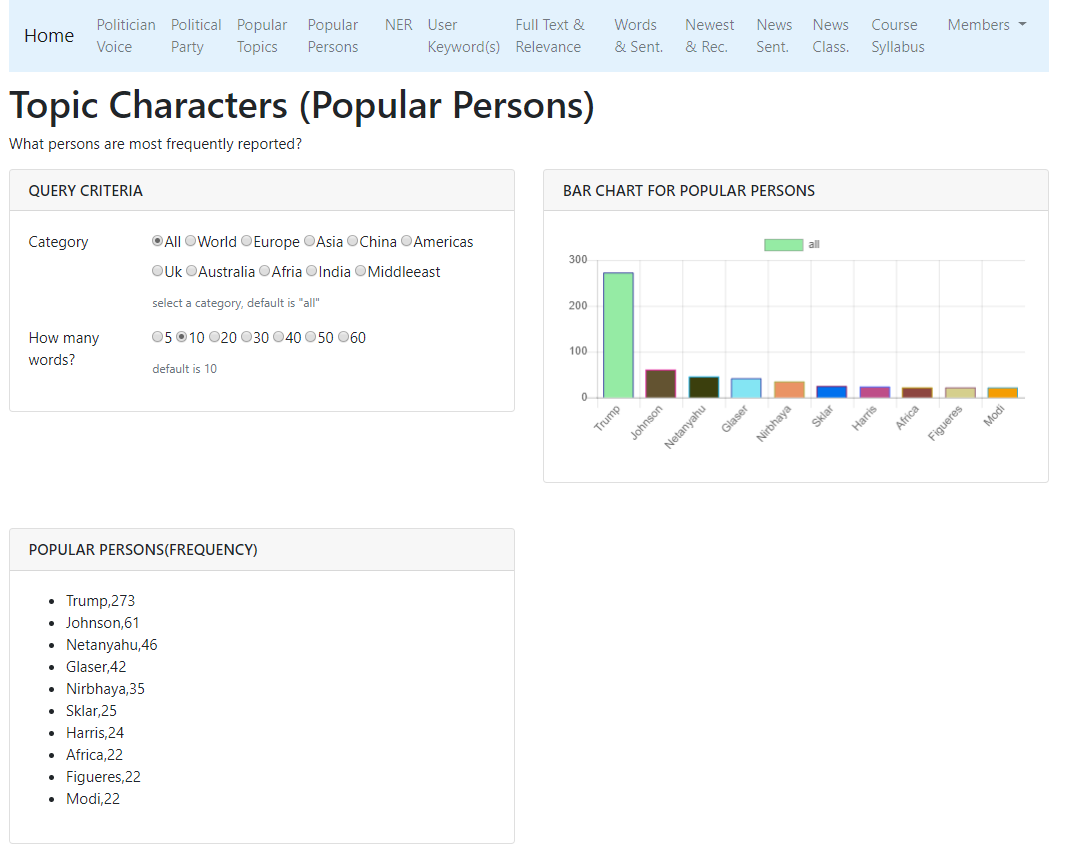
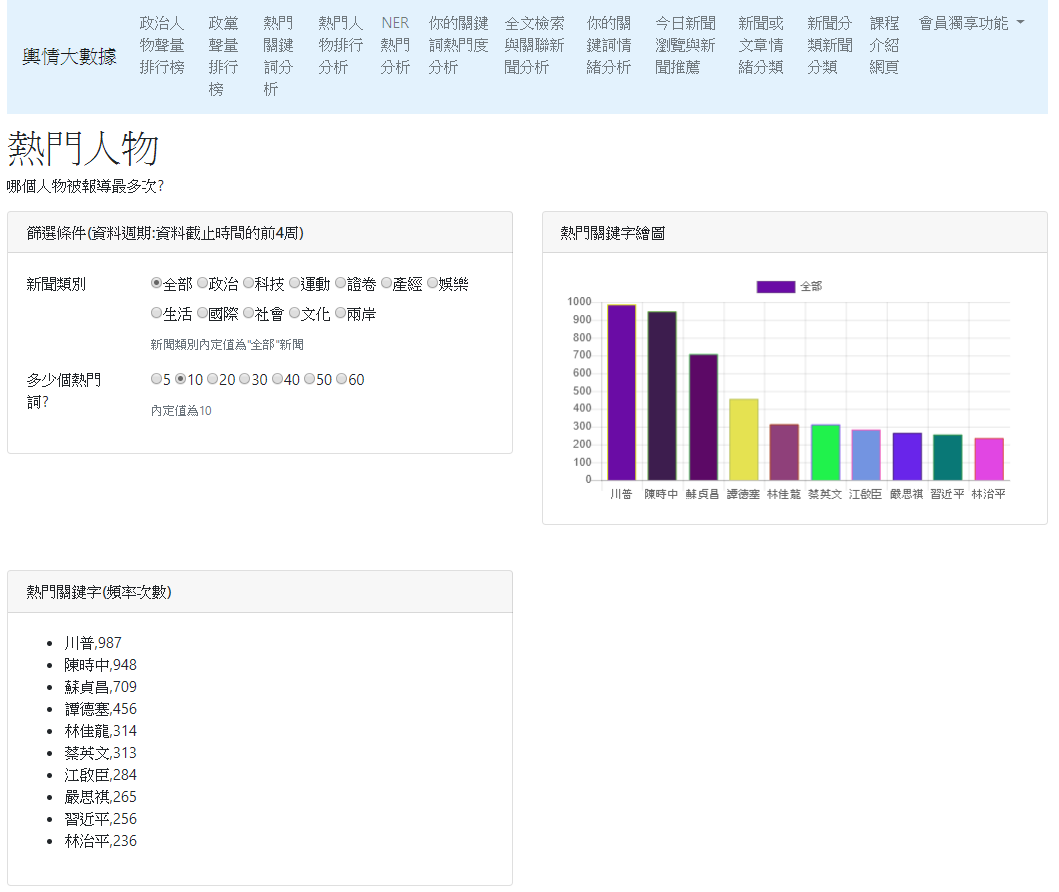
App Top Persons熱門人物分析

We want to display and chart the top persons which are the most frequently reported on our news website.

We use Ajax in this tutorial. However, you can use any other techniques to implement the website.

There are many chart libraries you can use, such as [D3.js](https://d3js.org/) and [Highcharts](http://www.highcharts.com/). We choose Char.js. We draw bar chart for top keywords using Char.js.





Development tools

In this class, we use “VS Code.” However, you can use PyCharm, Sublime, Atom, Notepad++, vi, vim, or any other favorite editor.

Continue from the previous project

We follow the previous project; that is, we just copy the previous project, and then add a new app in the copied project.

Step 1: Create a project “site\_news\_analysis\_v2”

We will continue from our previous project.

Create a folder “site\_news\_analysis\_v2” by duplicating the folder “site\_news\_analysis\_v1”

I suggest you rename your project folder with a meaningful name such as:

“site\_news\_analysis\_v2\_top\_persons”

django-admin startproject website\_configs .

Step2: Open the project with your favorite Editor

In this class, we use “VS Code.”

Please open the project and we write codes in VS Code.

Now let’s open project “site\_news\_analysis\_v2” using VS Code.

Step 3: Create a new APP “app\_top\_person”

Step 1: Please make sure if you are in the folder of “site\_news\_analysis\_v2.” If not, please go into the folder “site\_news\_analysis\_v2”

Step 2: Create an APP named “app\_top\_person”

django-admin startapp app\_top\_person

or

python manage.py startapp app\_top\_person

Step 4: settings.py

settings.py

|  |
| --- |
| ALLOWED\_HOSTS = ['127.0.0.1'] #or other public IP  INSTALLED\_APPS = [  …..  'app\_top\_keyword',  'app\_top\_person',  ]  TEMPLATES = [  {  'BACKEND': 'django.template.backends.django.DjangoTemplates',  'DIRS': [os.path.join(BASE\_DIR, 'templates')],  'APP\_DIRS': True,  'OPTIONS': …..  ……  ……  },  ] |

Notice 1: About the installed app:

Inside the INSTALLED\_APPS = [ … ], please make sure the app name should be included.

INSTALLED\_APPS = [

…..

' app\_top\_keyword',

' app\_top\_person',

]

Then, Django find your html files under the following folders

“app\_top\_keyword/templates/”

“app\_top\_person/templates/”

Usually our html file will be home.html or index.html. If we have several home.html, django cannot know which one is what you want to render. Therefore, we create another folder to avoid from confuse.

app\_top\_keyword/home.html

app\_top\_person/home.html

Therefore, the full path of the two home.html will be the following:

“app\_top\_keyword/templates/app\_top\_keyword/home.html”

“app\_top\_person/templates/app\_top\_person/home.html”

It is a very smart way to incrementally build up a large Django project.

Otherwise, we need to give each html file a different name.

Step 5: website\_configs/urls.py

website\_configs /urls.py

|  |
| --- |
| from django.urls import path  from django.urls import include  urlpatterns = [  # app top keywords  # path('', include('app\_top\_keyword.urls')), # root path  path('topword/', include('app\_top\_keyword.urls')),  # app top persons  path('topperson/', include('app\_top\_person.urls')),  ] |

Notice 2: About the templates path:

Inside the TEMPLATES =[ … ], please make sure if the following line exists:

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

Maybe you find it is as follows:

'DIRS': [],

If this happens, Django will not know the path of templates. You should modify it with the following:

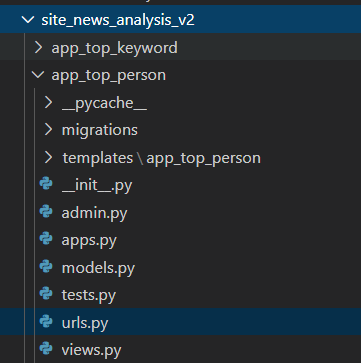
'DIRS': [os.path.join(BASE\_DIR, 'templates')],

Then, Django will find you html files in the websit\_news\_analysis\_v1/templates folder!

Step 6: app\_top\_person/urls.py

In folder app\_top\_person, create a Python file named “urls.py”.

app\_top\_person/urls.py



This is the urls.py you created.

app\_top\_keywrods/urls.py

|  |
| --- |
| from django.urls import path  from app\_top\_person import views  # declare a namespace for this APP  # the name of namespace is 'app\_top\_person'  # We will use the namespace in the future integrated website.  # 使用app\_name是讓各個APP的變數與方法名稱有區隔  # 若名稱不衝突，不使用app\_name也可以  # app\_name是一種namespace的概念  # 整合多個可獨立運作的APP成為一個大型專案必備知識  # 在template中如何使用?  # <a class="nav-link" href="{% url 'app\_top\_person:home' %}">熱門人物</a>  app\_name="app\_top\_person"  urlpatterns = [  # top (popular) persons  path('', views.home, name='home'),  # ajax path  #path('api\_get\_topPerson/', views.api\_get\_topPerson),  ] |

Notice 1:

path('api\_get\_topPerson/', views.api\_get\_topPerson),

path('api\_get\_topPerson', views.api\_get\_topPerson),

There is no slash at the end of path. It’s ok. But in the Ajax, its url should be also without slash.

Usually, there is a slash at the end of path.

Step 7: views.py

views.py

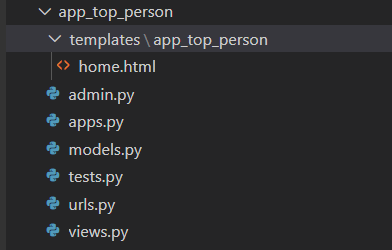
|  |
| --- |
| from django.shortcuts import render  import pandas as pd  from django.http import JsonResponse  from django.views.decorators.csrf import csrf\_exempt  def home(request):  return render(request, 'app\_top\_person/home.html')  print("app\_top\_person was loaded!") |

Step 8: HTML (Template)

home.html

In the app folder, create an HTML file named “home.html”

app\_top\_person/templates/app\_top\_person/home.html

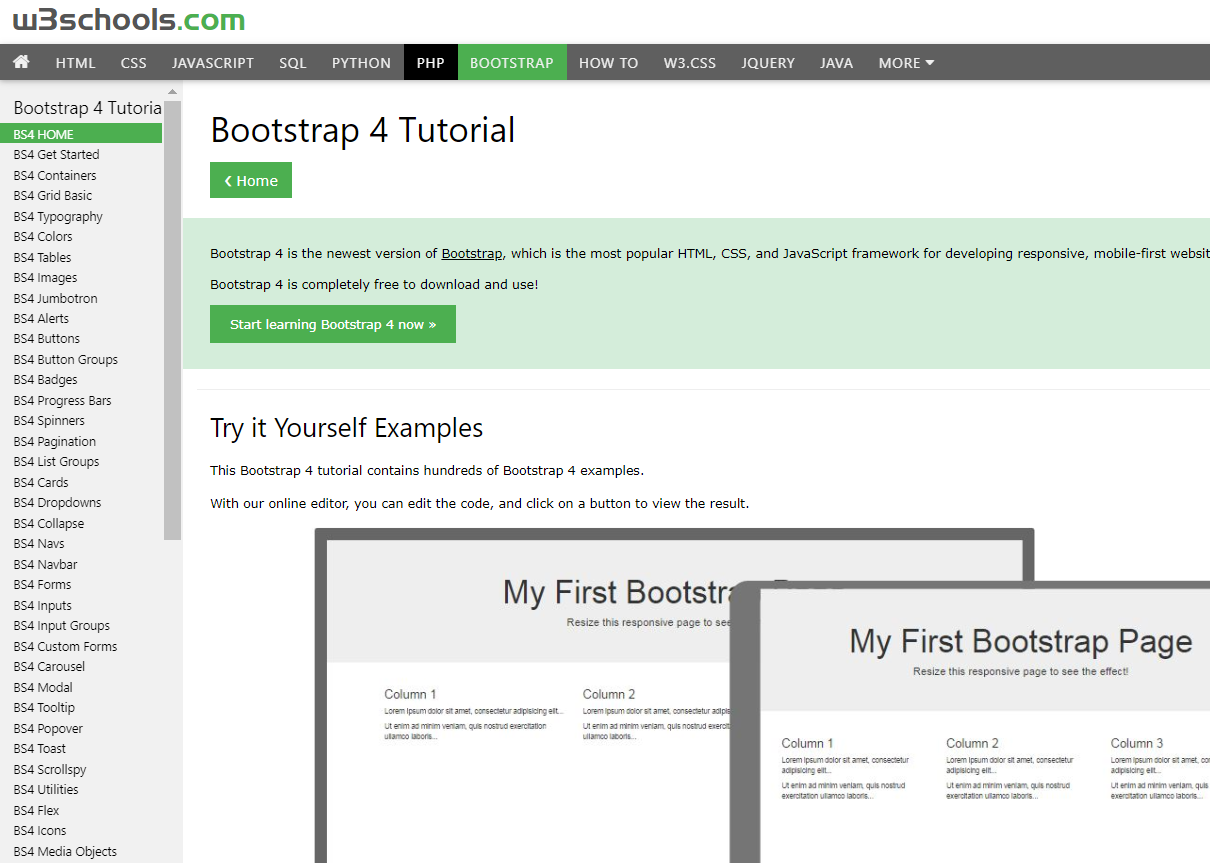


We can learn HTML and Bootstrap from:

<https://www.w3schools.com/bootstrap4/default.asp>

You can also learn many other frontend technologies from:

<https://www.w3schools.com/>



app\_top\_person/templates/app\_top\_person/home.html

home.html

|  |
| --- |
| {% extends 'base.html' %} {% block title %}  熱門人物  {% endblock %} {% block content %}  <div class="col-lg-12">  <h1>熱門人物</h1>  <p>哪個人物被報導最多次?</p>  </div>  <!-- column col-lg-6 card -->  <div class="col-lg-6 mb-5">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">篩選條件(資料週期:資料截止時間的前4周)</h3>  </div>  <div class="card-body">  <div class="mb-3 row">  <label class="col-sm-3 form-label">新聞類別</label>  <div class="col-md-9">  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="全部" name="cateradio" id="cateAll" checked />  <label class="form-check-label" for="cateAll">全部</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="政治" name="cateradio" id="catePolitics" />  <label class="form-check-label" for="catePolitics">政治</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="科技" name="cateradio" id="cateTech" />  <label class="form-check-label" for="cateTech">科技</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="運動" name="cateradio" id="cateSports" />  <label class="form-check-label" for="cateSports">運動</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="證卷" name="cateradio" id="cateFinance" />  <label class="form-check-label" for="cateFinance">證卷</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="產經" name="cateradio" id="cateEconomy" />  <label class="form-check-label" for="cateEconomy">產經</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="娛樂" name="cateradio" id="cateEntertain" />  <label class="form-check-label" for="cateEntertain">娛樂</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="生活" name="cateradio" id="cateLife" />  <label class="form-check-label" for="cateLife">生活</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="國際" name="cateradio" id="cateInternational" />  <label class="form-check-label" for="cateInternational">國際</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="社會" name="cateradio" id="cateSociety" />  <label class="form-check-label" for="cateSociety">社會</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="文化" name="cateradio" id="cateCulture" />  <label class="form-check-label" for="cateCulture">文化</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="兩岸" name="cateradio" id="cateCross" />  <label class="form-check-label" for="cateCross">兩岸</label>  </div>  <div class="form-text">新聞類別內定值為"全部"新聞</div>  </div>  </div>  <!-- form group -->  <!-- Number of persons form group -->  <div class="mb-3 row">  <label class="col-md-3 form-label">多少個熱門詞?</label>  <div class="col-md-9">  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="5" name="topkradio" id="topk5" />  <label class="form-check-label" for="topk5">5</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="10" name="topkradio" id="topk10" checked />  <label class="form-check-label" for="topk10">10</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="20" name="topkradio" id="topk20" />  <label class="form-check-label" for="topk20">20</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="30" name="topkradio" id="topk30" />  <label class="form-check-label" for="topk30">30</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="40" name="topkradio" id="topk40" />  <label class="form-check-label" for="topk40">40</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="50" name="topkradio" id="topk50" />  <label class="form-check-label" for="topk50">50</label>  </div>  <div class="form-check form-check-inline">  <input class="form-check-input" type="radio" value="60" name="topkradio" id="topk60" />  <label class="form-check-label" for="topk60">60</label>  </div>  <div class="form-text">內定值為10</div>  </div>  </div>  <!-- form group -->  </div>  <!-- card body -->  </div>  <!-- card -->  </div>  <!-- column -->  <!-- 繪圖區塊 -->  <div class="col-lg-6 mb-5">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">熱門關鍵字繪圖</h3>  </div>  <div class="card-body">  <canvas id="mychart"></canvas>  </div>  </div>  </div>  <!-- column -->  <!-- 熱門關鍵字區塊 -->  <div class="col-lg-6 mb-5">  <div class="card">  <div class="card-header">  <h3 class="h6 text-uppercase mb-0">熱門關鍵字(頻率次數)</h3>  </div>  <div class="card-body">  <ul id="topkeys"></ul>  </div>  </div>  </div>  <!-- column -->  {% endblock %} {% block extra\_js %}  <!-- jQuery js -->  <script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>  <!-- chartjs js -->  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.min.js"></script>  <!-- Your JavaScript Codes -->  <script>  //Write your JS code here!    // Call ajax function when page is loaded  call\_ajax()    // event handling for category radio button  $("input[name='cateradio']").on('change', function () {  call\_ajax()  }) //event function    // event handling for top k radio button  $("input[name='topkradio']").on('change', function () {  call\_ajax()  }) //event function    // Here is the Ajax function to draw chart  function call\_ajax() {  const cate = $("input[name='cateradio']:checked").val()  const topk = $("input[name='topkradio']:checked").val()  $.ajax({  type: 'POST',  url: '/topperson/api\_get\_topPerson/',  //url: "http://163.18.23.21:8000/topperson/api\_get\_topPerson/",  //url: "http://127.0.0.1:8000/topperson/api\_get\_topPerson/",  //url: "http://163.18.80.112:8000/topperson/api\_get\_topPerson/",  //url: "http://163.18.80.112:8000/topperson/api\_get\_topPerson/",    data: { news\_category: cate, topk: topk },  success: function (received) {  chart\_data = received.chart\_data  wf\_pairs = received.wf\_pairs  // Show chart data on console  console.log(wf\_pairs)    // Draw chart  showChart(chart\_data)  showTopKeys(wf\_pairs)  } //ajax function  }) //ajax  } //call\_ajax    //\* 顯示關鍵詞資料函數  function showTopKeys(items) {  //先清除前一次的資料  $('#topkeys').empty()    //將內容加上li標籤附加起來，顯示在顯示區"topkeys"  for (let i = 0; i < items.length; i++) {  let item\_li = '<li>' + items[i] + '</li>'  $('#topkeys').append(item\_li)  }  } //function    //\*\*繪圖函數showChart()  function showChart(chart\_data) {  // 畫圖需要的數據資料  let values = chart\_data.values  let labels = chart\_data.labels  let category = chart\_data.category    //第1個變數: 餵給chart的資料  let data = {  labels: labels,  datasets: [  {  label: category,  data: values,  backgroundColor: randomColors(values.length),  borderColor: randomColors(values.length),  borderWidth: 1  }  ]  }    //第2個變數: chart的選項 指定y坐標軸從零開始顯示  let options = {  scales: {  yAxes: [  {  ticks: {  beginAtZero: true  }  }  ]  }  }    //取得在前面html區域欲顯示的圖代號  let canvas\_mychrat = document.getElementById('mychart')    //\*\*先清除前一個圖 再繪新圖  // 可以印出barchart物件是否存在  // console.log(window.barchart);  //先清除前一個圖 再繪新圖 if 有以下兩種寫法皆可  // if (window.barchart) //若存在則為true  // if (typeof (barchart) != "undefined"){  if (window.barchart) {  barchart.destroy()  }    //\*\*繪圖(產生一個圖物件變數名稱為barchart)  // 必須全域變數--注意:前面不要有let, var, const等修飾詞  // 理由: 我們要讓它存在於網頁全域變數，  // 這樣我們才方便判斷是否有前一次的圖，如果存在有，要刪除之，否則，很多張圖會疊在一起  barchart = new Chart(canvas\_mychrat, {  type: 'bar',  data: data,  options: options  })    //\*\* 產生隨機顏色  function randomColors(num\_colors) {  let colors = []    for (let i = 0; i < num\_colors; i++) {  let r = Math.floor(Math.random() \* 255)  let g = Math.floor(Math.random() \* 255)  let b = Math.floor(Math.random() \* 255)  let rgb = `rgba(${r},${g},${b},0.5)` // (red, green, blue, alfa) alfa透明度    colors.push(rgb)  }  return colors  }  } //show chart function  </script>  {% endblock %} |

Step 9: base.html

base.html

|  |
| --- |
| <!DOCTYPE html>  {% load static %}  <html lang="zh-TW">  <head>  <title>  {% block title %}  輿情分析平台  {% endblock %}  </title>  <meta charset="utf-8" />  <meta name="viewport" content="width=device-width, initial-scale=1" />  <!-- Bootstrap 5 CSS -->  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" />  {% block extra\_css %}  {% endblock %}  </head>  <body>  <div class="container">  <div class="row">  <!-- Navigation Bar -->  <div class="col-lg-12 mb-2 mt-2">  <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #e3f2fd">  <div class="container-fluid">  <a class="navbar-brand" href="/">輿情大數據</a>  <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation"><span class="navbar-toggler-icon"></span></button>  <div class="collapse navbar-collapse" id="navbarSupportedContent">  <ul class="navbar-nav me-auto mb-2 mb-lg-0">  <!-- 基礎熱門分析 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">熱門分析</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="{% url 'app\_top\_keyword:home' %}">熱門關鍵詞分析</a>  <a class="dropdown-item" href="#">熱門人物排行分析</a>  <a class="dropdown-item" href="#">命名實體熱門分析</a>  <a class="dropdown-item" href="#" style="color: green">昨日誰最大</a>  </div>  </div>  <!-- 進階自訂分析 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">進階查詢</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">自訂關鍵詞熱門度分析</a>  <a class="dropdown-item" href="#">自訂全文檢索與關聯分析</a>  <a class="dropdown-item" href="#">自訂關鍵詞之情緒分析</a>  </div>  </div>  <!-- 特色分析 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">特色分析</button>  <div class="dropdown-menu">  <a class="nav-link" href="#" style="color: green">時中聲量我最大</a>  <div class="dropdown-divider"></div>  <a class="nav-link" href="#" style="color: green">台北市長選舉聲量觀測站</a>  <div class="dropdown-divider"></div>  <a class="nav-link" href="#">政黨聲量排行榜</a>  <a class="nav-link" href="#">政治人物聲量排行榜</a>  <div class="dropdown-divider"></div>  <a class="nav-link" href="#" style="color: green">事件人物聲量相關分析</a>  </div>  </div>  <!-- 新聞推薦系統 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">新聞推薦系統</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">Bert/Qwen新聞推薦-新聞查找相似新聞</a>  </div>  </div>  <!-- NLP應用 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">自然語言理解</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">語言模型Bert/Qwen情緒分類</a>  <a class="dropdown-item" href="#">語言模型Bert/Qwen新聞分類</a>  </div>  </div>  <!-- 使用資料庫 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">使用DB</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">熱門人物資料庫查詢</a>  <a class="dropdown-item" href="#">昨日新聞資料庫全文檢索</a>  </div>  </div>  <!-- 其他特色應用 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">特色應用</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">高雄市市長選舉人氣分析</a>  <a class="dropdown-item" href="#">珊珊市長人氣</a>  <a class="dropdown-item" href="#">阿邁市長人氣</a>  <div class="dropdown-divider"></div>  <a class="dropdown-item" href="#">疫情發燒監視</a>  <a class="dropdown-item" href="#">蘋果發燒站</a>  <div class="dropdown-divider"></div>  <a class="dropdown-item" href="#">人氣美食排行榜</a>  <a class="dropdown-item" href="#">財經議題排行榜</a>  <a class="dropdown-item" href="#">股市新聞監視站</a>  <a class="dropdown-item" href="#">科技新聞熱門議題分析</a>  <a class="dropdown-item" href="#">人力銀行職缺大熱門</a>  <div class="dropdown-divider"></div>  <a class="dropdown-item" href="#">批踢踢酸民大吐槽</a>  <a class="dropdown-item" href="#">批踢踢鄉民來八卦</a>  <a class="dropdown-item" href="#">滴卡大學生關心議題</a>  <div class="dropdown-divider"></div>  <a class="dropdown-item" href="#">新聞媒體政黨傾向調查分析</a>  <a class="dropdown-item" href="#">自訂競爭大PK</a>  </div>  </div>  <!-- 會員訂閱 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">付費訂閱</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">長時間(超過3個月)監測分析(收費表)</a>  <a class="dropdown-item" href="#">更多資料庫分析(收費表)</a>  <a class="dropdown-item" href="#">為你訂製輿情分析任務(報價)</a>  </div>  </div>  <!-- 關於 -->  <div class="btn-group">  <button type="button" class="btn dropdown-toggle" data-bs-toggle="dropdown" aria-expanded="false">關於</button>  <div class="dropdown-menu">  <a class="dropdown-item" href="#">課程介紹網頁</a>  <a class="dropdown-item" href="#">如何使用輿情API</a>  </div>  </div>  </ul>  </div>  </div>  </nav>  </div>  <!-- End of Navigation Bar -->  <!-- Main Content -->  {% block content %}  {% endblock %}  </div>  </div>  <!-- Bootstrap 5 JS Bundle with Popper -->  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>  <!-- Additional Scripts -->  {% block extra\_js %}  {% endblock %}  </body>  </html> |

Modify href link

Only one line you need to modify:

 <a class="nav-link" href="{% url 'app\_top\_person:home' %}">熱門人物分析</a>

</li>

{% url 'app\_top\_person:home' %}

app\_name\_space:url\_name

app\_top\_person:home

app\_name\_space

url\_name

app\_top\_person and home are defined in urls.py as follows.

app\_top\_keywrods/urls.py

path('', views.home, name='home'), # app\_top\_person:home

In the “base.html”, modify the following line:

|  |
| --- |
| <li class="nav-item">  <a class="nav-link" href="{% url 'app\_top\_keyword:home' %}">熱門關鍵詞分析</a>  </li>  <li class="nav-item">  <a class="nav-link" href="{% url 'app\_top\_person:home' %}">熱門人物排行分析</a>  </li> |

Step 9: Run server

python manage.py runserver 8000

<http://127.0.0.1:8000/topperson/>

What does the web page look like?

Step 11: Try radio button event using jQuery

Now you can write your JavaScript codes.

|  |
| --- |
| <!-- jQuery js-->  <script src="https://code.jquery.com/jquery-3.1.0.min.js"></script>  <!-- chartjs js-->  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.3/Chart.min.js"></script>  <!-- Your JavaScript Codes -->  <script>  //Write your JS code here!  </script> |

Test radio button on change event

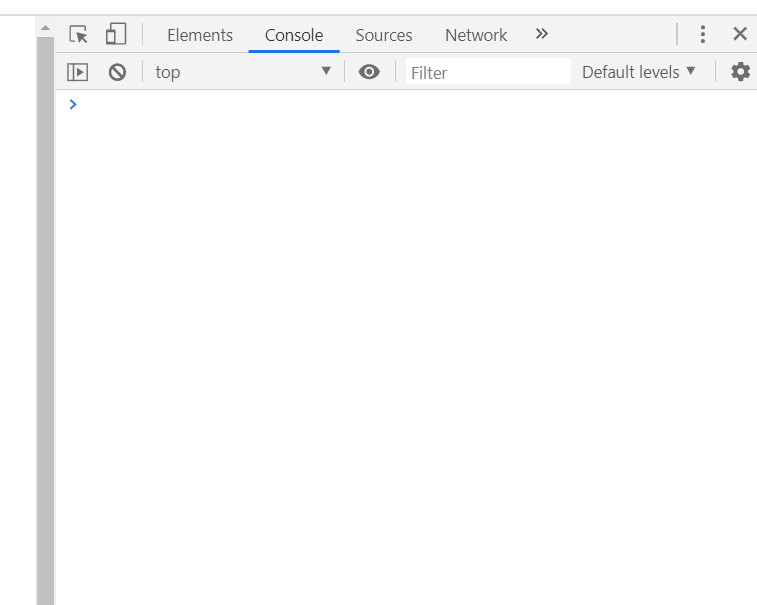
|  |
| --- |
| // event handling for category radio button  $("input[name='cateradio']").on('change', function () {  console.log("category radio button");  const cate = $("input[name='cateradio']:checked").val();  console.log(cate);  //call\_ajax();  }); //event function |

 Test event handling for top k radio button

|  |
| --- |
| // event handling for top k radio button  $("input[name='topkradio']").on('change', function () {  const topk = $("input[name='topkradio']:checked").val();  alert('top k:' + topk);  //call\_ajax();  }); //event function |

Press F12

See what happens in the following screen when you click on the radio button.



Step 12: Write your Ajax function to get top persons from other’s public Restful API

There is a public API where you can get top persons information.

url: "http://163.18.23.21:8000/topperson/api\_get\_topPerson/",

Now let’s write an Ajax function to get top persons of a certain category.

Ajax function

|  |
| --- |
| // Call ajax function when page is loaded  call\_ajax();  // Here is the Ajax function to draw chart  function call\_ajax() {  const cate = $("input[name='cateradio']:checked").val();  const topk = $("input[name='topkradio']:checked").val();  $.ajax({  type: "POST",  //url: "/topperson/api\_get\_topPerson/",  url: "http://163.18.23.21:8000/topperson/api\_get\_topPerson/",  //url: "http://127.0.0.1:8000/topperson/api\_get\_topPerson/",  data: { "news\_category": cate, "topk": topk },  success: function(received) {  chart\_data = received.chart\_data;  wf\_pairs = received.wf\_pairs;  // Show chart data on console  console.log(wf\_pairs);  // Show top keywords  //showTopKeys(wf\_pairs);  // Draw chart  //showChart(chart\_data);  } //ajax function  }); //ajax  }; //call\_ajax |

A concept you need to know:

There are two ways to define a function:

* Function declaration with hoisting提升:

function call\_ajax() {}

call\_ajax() can be used before or after function definition. That is, the function can be called anywhere before or after its declaration.

* Function expression:

var call\_ajax = function() {}

call\_ajax() can be used only after function declaration.

That is, to use the function, we should declare it before using it. Just like a variable—we declare it, then we can use it.

Radio button on change event to call Ajax function

|  |
| --- |
| $("input[name='cateradio']").on('change', function() {  call\_ajax();  }); //event function  $("input[name='topkradio']").on('change', function() {  call\_ajax();  }); //event function |

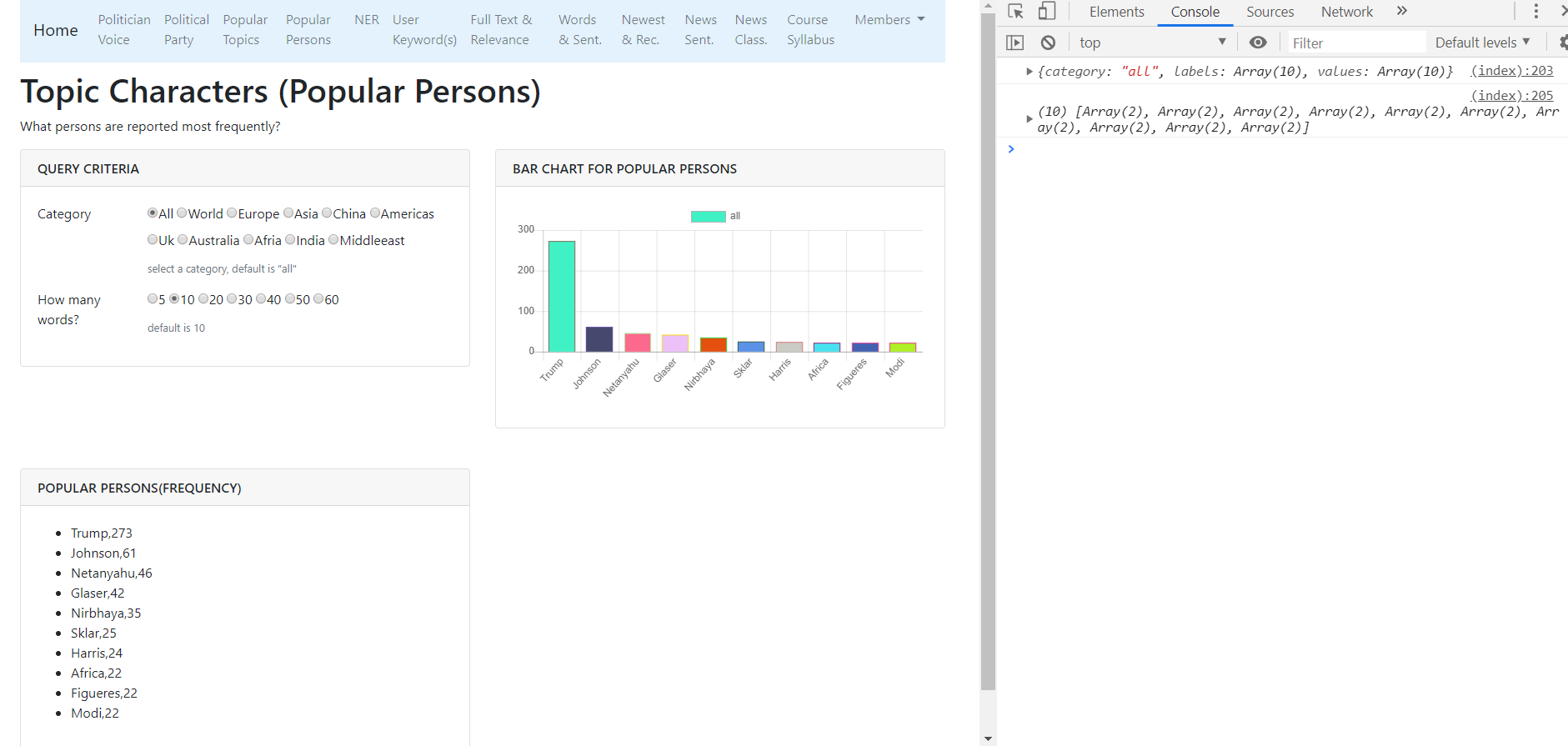
Here is the web page we got.



Look at the data you get from a public API

Step 13: Display and draw bar chart

What is the web page look like?



Display keyword and frequency

|  |
| --- |
| //\* 顯示關鍵詞資料函數  function showTopKeys(items) {  //先清除前一次的資料  $('#topkeys').empty();  //將內容加上li標籤附加起來，顯示在顯示區"topkeys"  for (let i = 0; i < items.length; i++) {  let item\_li = "<li>" + items[i] + "</li>";  $('#topkeys').append(item\_li);  }  } //function |

Show chart function

|  |
| --- |
| //\*\*繪圖函數showChart()  function showChart(chart\_data) {  // 畫圖需要的數據資料  let values = chart\_data.values;  let labels = chart\_data.labels;  let category = chart\_data.category;  //第1個變數: 餵給chart的資料  let data = {  labels: labels,  datasets: [{  label: category,  data: values,  backgroundColor: randomColors(values.length),  borderColor: randomColors(values.length),  borderWidth: 1,  }],  };  //第2個變數: chart的選項 指定y坐標軸從零開始顯示  let options = {  scales: {  yAxes: [{  ticks: {  beginAtZero: true  }  }]  },  };  //取得在前面html區域欲顯示的圖代號  let canvas\_mychrat = document.getElementById("mychart");  //\*\*先清除前一個圖 再繪新圖  // 可以印出barchart物件是否存在  // console.log(window.barchart);  //先清除前一個圖 再繪新圖 if 有以下兩種寫法皆可  // if (window.barchart) //若存在則為true  // if (typeof (barchart) != "undefined"){  if (window.barchart) {  barchart.destroy();  }  //\*\*繪圖(產生一個圖物件變數名稱為barchart)  // 必須全域變數--注意:前面不要有let, var, const等修飾詞  // 理由: 我們要讓它存在於網頁全域變數，  // 這樣我們才方便判斷是否有前一次的圖，如果存在有，要刪除之，否則，很多張圖會疊在一起  barchart = new Chart(canvas\_mychrat, {  type: 'bar',  data: data,  options: options,  });  //\*\* 產生隨機顏色  function randomColors(num\_colors) {  let colors = [];  for (i = 0; i < num\_colors; i++) {  let r = Math.floor(Math.random() \* 255);  let g = Math.floor(Math.random() \* 255);  let b = Math.floor(Math.random() \* 255);  let rgb = `rgba(${r},${g},${b},0.5)` // (red, green, blue, alfa) alfa透明度  colors.push(rgb);  }  return colors;  }  } //show chart function |

Appendix 1 is the complete version of the home.html for your reference.

Step 14: Make your backend API work

Now we can write codes in views.py to make our backend API work.

Step 14.1: Put your dataset into the project

Copy your top keywords csv file into folder “dataset” under your app\_top\_person as follows:

app\_top\_person/dataset/news\_top\_person\_by\_category\_via\_ner.csv

~~site\_news\_analysis\_v2/dataset/news\_top\_person\_by\_category\_via\_ner.csv~~

Step 14.2: views.py

views.py

|  |
| --- |
| from django.shortcuts import render  import pandas as pd  from django.http import JsonResponse  from django.views.decorators.csrf import csrf\_exempt  '''  the format of data:  {'政治': [('韓國瑜', 6344),  ('蔡英文', 2114),  ('賴清德', 1480),  ...  }  '''  # load data  def load\_data\_topPerson():  # read df  df\_topPerson = pd.read\_csv(  'app\_top\_person/dataset/news\_top\_person\_by\_category\_via\_ner.csv')  # refresh data  global data # make data global. It can be used everywhere.  data = {}  for idx, row in df\_topPerson.iterrows():  data[row['category']] = eval(row['top\_keys'])  # Load data first when starting server.  load\_data\_topPerson()  def home(request):  return render(request, 'app\_top\_person/home.html')  # csrf\_exempt is used for POST  # 單獨指定這一支程式忽略csrf驗證  @csrf\_exempt  def api\_get\_topPerson(request):  # chart\_data, wf\_pairs = get\_category\_topkey("科技", 10) #先做簡單的測試  cate = request.POST.get('news\_category')  topk = request.POST.get('topk')  topk = int(topk)  #print(cate, topk)  chart\_data, wf\_pairs = get\_category\_topPerson(cate, topk)  # print(chart\_data)  response = {'chart\_data': chart\_data,  'wf\_pairs': wf\_pairs,  }  return JsonResponse(response)  def get\_category\_topPerson(cate, topk):  wf\_pairs = data[cate][0:topk]  words = [w for w, f in wf\_pairs]  freqs = [f for w, f in wf\_pairs]  chart\_data = {  "category": cate,  "labels": words,  "values": freqs}  return chart\_data, wf\_pairs # chart\_data is for charting  print("app\_news\_analysis--類別熱門人物載入成功!") |

Step 14.3: Unmark your Api path in app\_top\_person/urls.py

Your ajax path:

path('api\_get\_topPerson/', views.api\_get\_topPerson),

app\_top\_person/urls.py

|  |
| --- |
| from django.urls import path  from app\_top\_person import views  # 使用app\_name是讓各個APP的變數與方法名稱有區隔  # 若名稱不衝突，不使用app\_name也可以  # app\_name是一種namespace的概念  # 整合多個可獨立運作的APP成為一個大型專案必備知識  # 在template中如何使用?<a class="nav-link" href="{% url 'app\_top\_person:home' %}">熱門人物</a>  app\_name="app\_top\_person"  urlpatterns = [  # top (popular) persons  path('', views.home, name='home'),  # ajax path  path('api\_get\_topPerson/', views.api\_get\_topPerson, name='api\_get\_topPerson'),  ] |

Step 14.4: Ajax url path in HTML (Template)

Modify ajax url to call your backend Api function.

|  |
| --- |
| $.ajax({  type: "POST", //or GET  //url: "/topperson/api\_get\_topPerson/", //絕對路徑  //url: "api\_get\_topPerson/", //相對路徑  //url: "http://163.18.23.120:8000/topperson/api\_get\_topPerson/",  url: "http://127.0.0.1:8000/topperson/api\_get\_topPerson/", |