* EJS是啥?

EJS (Embedded\_JavaScript)

* EJS stands for Embedded JavaScript.

嵌入式的JS

* EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.

是一個簡單的樣板，讓你可以透過JS的for loop去撰寫好多好多HTML tag(ex: li )

* It’s a module in npm.

是一個npm module(透過Node.js執行)

* Ejs環境設定

首先，環境就要先設定(因為它也是個Node.js module)

新創一個proj資料夾要先在: CMD > npm init;

創建好之後再把要安裝的裝一裝在: CMD > npm install express nodemon ejs

以上都安裝好之後可以用vscode確認是否安裝成功

到package.json看看"dependencies" 是否三個都有安裝到

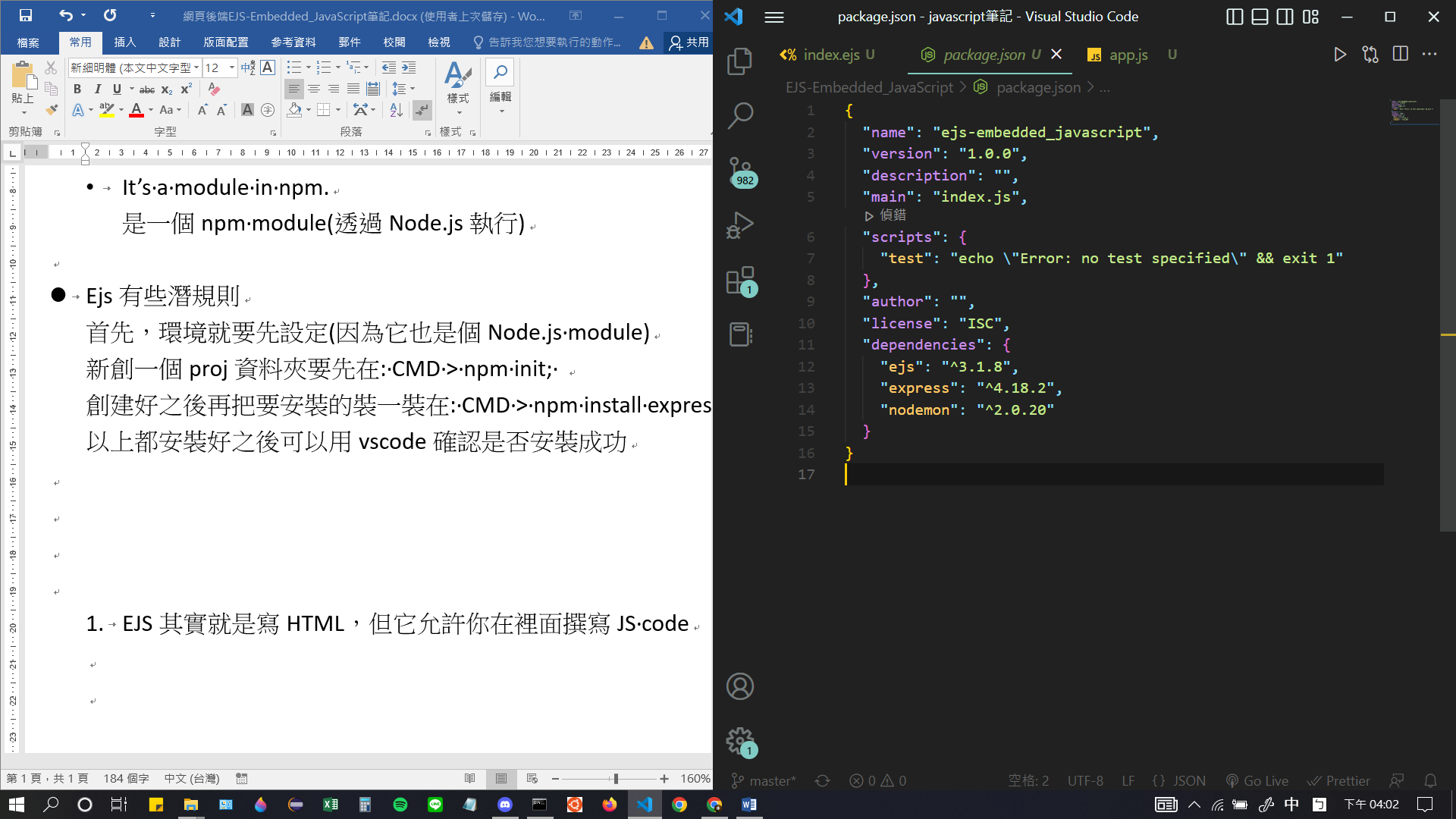
"dependencies": {

"ejs": "^3.1.8",

"express": "^4.18.2",

"nodemon": "^2.0.20"

}



* Ejs環境
  + app.js基本設定(express, ejs先require起來，放進變數)

const express = require("express");

const app = express();

const ejs = require("ejs");

//這邊的middleware是為了要讓<link href=””>正確的吃到.css檔

//資料夾另外創個public，並且把.css檔等要用的檔案放入

app.use(express.static("public"));

app.get("/",(req,res)=>{

//處理user要進到首頁的get請求，之前我們要回傳HTML是靠res.sendFile()如下

//res.sendFile(\_\_dirname+"/index.html"); 🡨🡨sendFile裡面裝路徑

//ejs不一樣，要透過res.render()裡面”直接放進.ejs檔名稱”

    res.render("index.ejs");

//透過res.render直接上ejs名稱有個大前提，*你必須再project資料夾中設一個views資料夾，*

*//並且將ejs資料放進裡面【這是ejs潛規則，這也是為什麼ejs叫做框架!】，*

//你沒創個views檔案會跳下面的錯誤

*// Error: Failed to lookup view "index.ejs" in views directory "C:\Users\林哲緯\Desktop\【前端資料\javascript筆記\EJS-Embedded\_JavaScript\views"*

});

//handle user get request

app.get("/:name",(req,res)=>{

    let {name}=req.params; *//(object destruction) 會取得user打在網址/後面的任意值*

    res.render("person.ejs",{ getNames:name });

*//設定name的屬性叫做name 當然可以用其他的值來去做替代*

*//res.render()*

*//第一個parameter要放進.ejs檔檔名*

*//第二個parameter可以放進一個object{ :前放要丟進ejs的代稱，:後放req.params取的值}*

*//，而這個object是從req.params來的，這個物件可以在person.ejs來做使用*

*//req.params 目前的設定log出來會是一個object然後是以{name : userinput}的樣子的object*

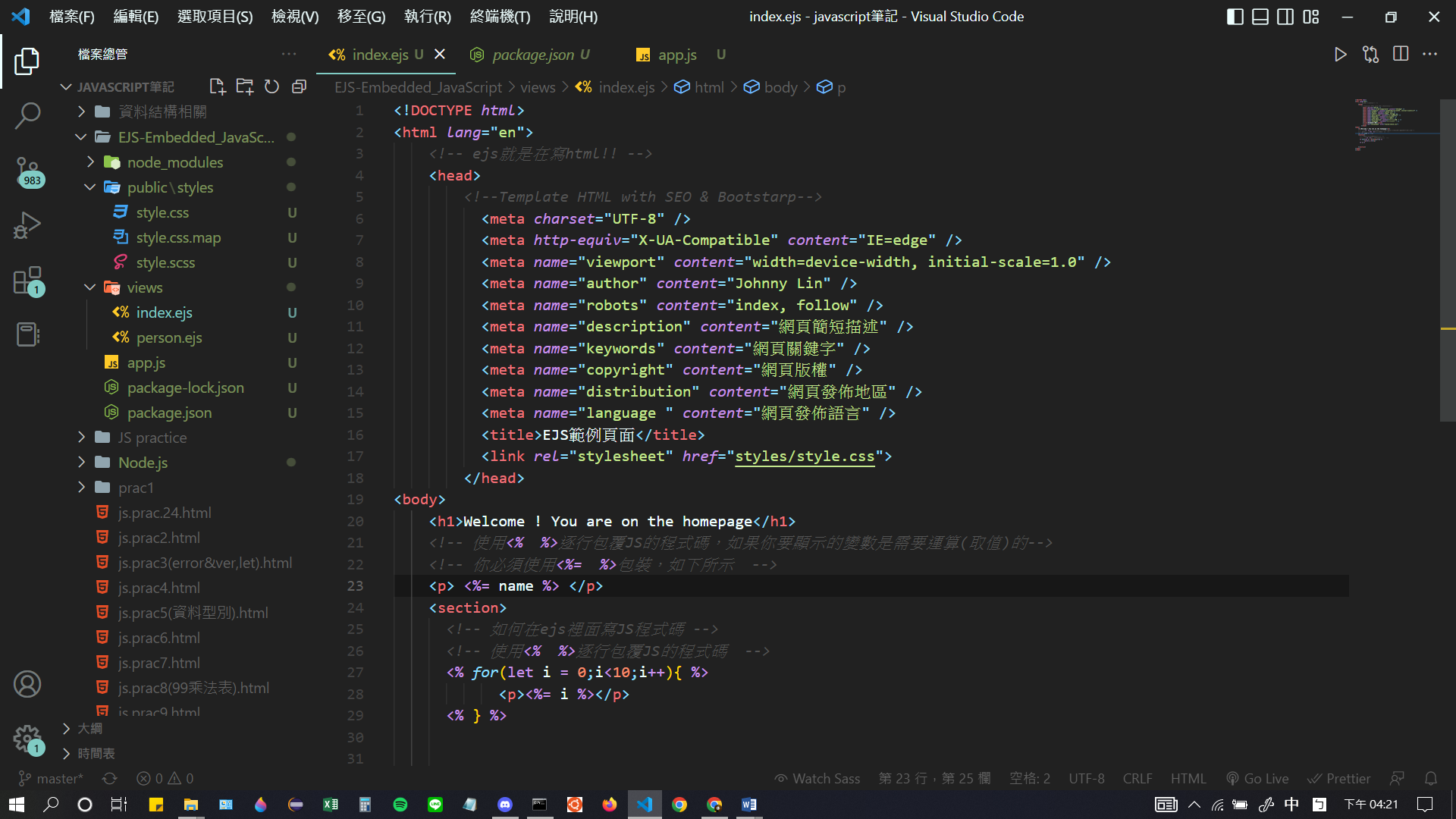
});

app.listen(3000,()=>{

    console.log('Server is running on 3000 port');

})

* Ejs環境
  + 按照上面提到的ejs的框架規則，我們必須在proj資料夾另外創個views資料夾
  + 把.ejs檔都放進去



* + EJS的寫法其實就是HTML tag，但它更進一步得允許你在裡面撰寫JS code

<!DOCTYPE *html*>

<html *lang*="en">

    <head>

          <meta *charset*="UTF-8" />

          <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

          <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

          <link *rel*="stylesheet" *href*="styles/style.css">

//注意，這邊的link tag它的href屬性attr要注意，*href*="styles/style.css"

//【原本應該是 ./public/styles/style.css】，但為配合middleware只寫styles/style.css

        </head>

* + 在ejs裡面撰寫js code的方法: <% %>包覆 JS code
  + 可以在vscode先載EJS Language Support

<body>

    <h1>Welcome ! You are on the homepage</h1>

*<!-- 使用<%  %>逐行包覆JS的程式碼，如果你要顯示的變數是需要運算(取值)的-->*

*<!-- 你必須使用<%=  %>包裝，如下所示  -->*

    <p> <%= name %> </p>

    <section>

*<!-- 如何在ejs裡面寫JS程式碼 -->*

*<!-- 使用<%  %>逐行包覆JS的程式碼  -->*

      <% *for*(let i = 0;i<10;i++){ %>

            <p><%= i %></p>

因為在ejs內html tag才是主體，有牽涉到tag時*<%=  %>必須放在tag裡面*

      <% } %> 連最後右大括號也算是JS code 記得也要*<%=  %>包覆*

    </section>

</body>

</html>

* Ejs環境處理表單(form tag):

基礎設定如上

App.js:

const express = require("express");

const app = express();

const ejs = require("ejs");

app.use(express.static("public"));

app.get("/",(req,res)=>{

    res.render("index.ejs");

});

*//handling a form tag in ejs【這邊ejs form來的是get request】*

app.get("/response",(req,res)=>{ //”/response” 跟form tag的action=”/response” 值要對應

    console.log(req.query); //所以我們要用req.query取得form來的資訊

    let {fullname,age}=req.query; //object destruct 取得user輸入的值

    res.render("respond.ejs",{fullname:fullname, age:age});

//取得的值放入res.render第二個parameter，讓fullname、age可以在respond.ejs使用

});

app.listen(3000,()=>{

    console.log('Server is running on 3000 port');

})

respond.ejs

  <body>

    <h1>Simple form</h1>

    <form *action*="/response" *method*="GET">

//注意這邊的action裡面要放/response對應app.js handle這個get request

      <label *for*="fullname">Your name:</label>

      <input *type*="text" *id*="fullname" *name*="fullname" />

      <label *for*="age">Your age:</label>

      <input *type*="text" *id*="age" *name*="age" />

      <button *type*="submit">Summit</button>

    </form>

  </body>

* Ejs環境處理串列(array like object)

很多資料庫出來的資料會是一個array ，尤其noSQL的資料庫

那要如何處理ARRAY資料，呈現在網頁上?

App.js

const express = require("express");

const app = express();

const ejs = require("ejs");

app.use(express.static("public"));

*//handling array*

app.get("/",(req,res)=>{

*//database => an array of objects*

    const lang =[ *//假定我們從資料庫收到的資訊是這種array like object的時候*

        {name:"Python",rating:9.5,popularity:9.7,trending:"super hot"},

        {name:"Java",rating:8.6,popularity:8.2,trending:"same"},

        {name:"C++",rating:6.6,popularity:7.7,trending:"same"},

        {name:"PHP",rating:2.5,popularity:4.7,trending:"decreasing"},

        {name:"JavaScript",rating:8.5,popularity:8.1,trending:"same"},

    ];

    res.render("index.ejs",{lang:lang});

});

app.listen(3000,()=>{

    console.log('Server is running on 3000 port');

})

Ejs部分

  <body>

    <h1>Common Programming Languages</h1>

    <table>

      <tr>

        <th>name</th>

        <th>rating</th>

        <th>popularity</th>

        <th>trending</th>

      </tr>

<br />

      <br />

*<!--內容就靠loop去一一填上-->*

      <%lang.forEach((e)=>{%>

<%console.log(e)%>

*<!--log結果如下-->*

*<!---{ name: 'Python', rating: 9.5, popularity: 9.7, trending: 'super hot' }*

*{ name: 'Java', rating: 8.6, popularity: 8.2, trending: 'same' }*

*{ name: 'C++', rating: 6.6, popularity: 7.7, trending: 'same' }*

*{ name: 'PHP', rating: 2.5, popularity: 4.7, trending: 'decreasing' }*

*{ name: 'JavaScript', rating: 8.5, popularity: 8.1, trending: 'same' }-->*

      <tr>

        <td><%=e.name%></td> //把要填入的資料放進td內

        <td><%=e.rating%></td>

        <td><%=e.popularity%></td>

        <td><%=e.trending%></td>

      </tr>

      <% });%>

    </table>

  </body>