目錄

[箭頭函數 3](#_Toc492476032)

[模組化物件 3](#_Toc492476033)

[this指標 3](#_Toc492476034)

[JS使用原則 3](#_Toc492476035)

[JSON 4](#_Toc492476036)

[資料格式 4](#_Toc492476037)

[宣告方式 4](#_Toc492476038)

[JSON與JS物件轉換 4](#_Toc492476039)

[REQUIRE 4](#_Toc492476040)

[回乎函數 (CallBack Function) 5](#_Toc492476041)

[檔案系統-FS 5](#_Toc492476042)

[NPM 7](#_Toc492476043)

[檢查是否安裝 7](#_Toc492476044)

[建立方式 7](#_Toc492476045)

[安裝檔案到package專案目錄 7](#_Toc492476046)

[檔案版本更新 7](#_Toc492476047)

[刪除檔案後從新安裝 7](#_Toc492476048)

[檔案版本更新 7](#_Toc492476049)

[伺服器修改後，自動重新啟動 7](#_Toc492476050)

[EXPRESS 8](#_Toc492476051)

[安裝EXPRESS 8](#_Toc492476052)

[開發階段測試 8](#_Toc492476053)

[EXPRESS網路監聽 8](#_Toc492476054)

[HTTP請求方式 8](#_Toc492476055)

[快速建立專案 8](#_Toc492476056)

[建立了一個在express-generator目錄之內的專案 9](#_Toc492476057)

[產生ejs的template 9](#_Toc492476058)

[擷取網頁URL 10](#_Toc492476059)

[範例一 10](#_Toc492476060)

[範例二 11](#_Toc492476061)

[靜態網頁 12](#_Toc492476062)

[JS尋找CSS 12](#_Toc492476063)

[EJS傳POST資料 13](#_Toc492476064)

[SASS改變CSS 14](#_Toc492476065)

[動態網頁 15](#_Toc492476066)

[取得POST DATA 15](#_Toc492476067)

# 箭頭函數

( ) => {…} no argument

x => {…} one argument

(x,x) => {…} several argument

greet = () =>console.log(‘L’);

greet();

# 模組化物件

module.exports 建立模組化物件

require 使用模組化物件

每個模組都會自動創建一個module對象，module對象會創建一個exports屬性，初始值是{}，例:

module.exports = {};而exports是引用module.exports的值

module.exports被改變時，exports不會被改變，而模組倒出時，執行的是module.exports，而不是exports。

exports = module.exports = somethings，意義在於:

module.exports = somethings是對module.exports進行覆蓋，module.exports 和exports關係斷裂，module.exports指向新的存塊，exports還是指向原來存塊，所以必須寫成exports = module.exports

Node.js提供許多原生模組

# this指標

物件方法中的this---物件的方法是被物件呼叫

函數之中的this---函數是被系統呼叫

# JS使用原則

物件的方法使用正規函數宣告，this就會指向物件

物件的方式裡面的函數使用箭頭函數宣告，this就會指向物件

(wek3 p.24 picture)

一個類別繼承另一個類別，子類別的建構子內就必須用super關鍵字呼叫父類別的建構子

# JSON

## 資料格式

JavaScrip Object Notation(JSON)

## 宣告方式

設aaa=bbb

在JSON寫成

“aaa”:”bbb”

所有name與字串都要用雙引號” ”包起來

JSON資料型態支援:String, Number, Objects, Arrays, Booleans, null

## JSON與JS物件轉換

JSON→JS

JSON.parse(…)

把JSON轉成JS

---------------------------

JS→JSON

JSON.stringify(…)

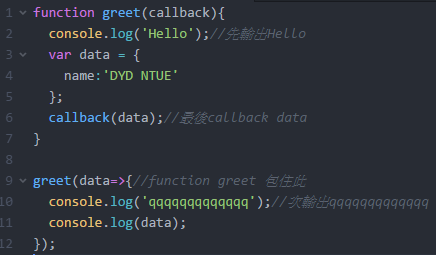
把JS轉成JSON

# REQUIRE

Require會自動把JSON資料轉成JS物件

(wk3 p.51)

# 回乎函數 (CallBack Function)



# 檔案系統-FS

Synchronous:同一時間只能執行一個動作

Asynchronous:同一時間只能執行多個動作



.readFileSync:同步檔案處理

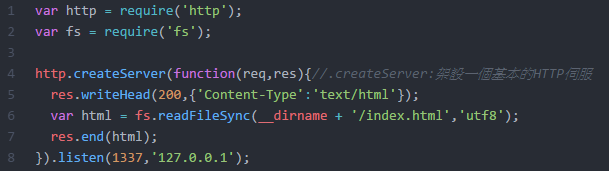
.readFile:非同步檔案處理

.readFile在背景執行，當程式結束後會回傳data，有如callback功能

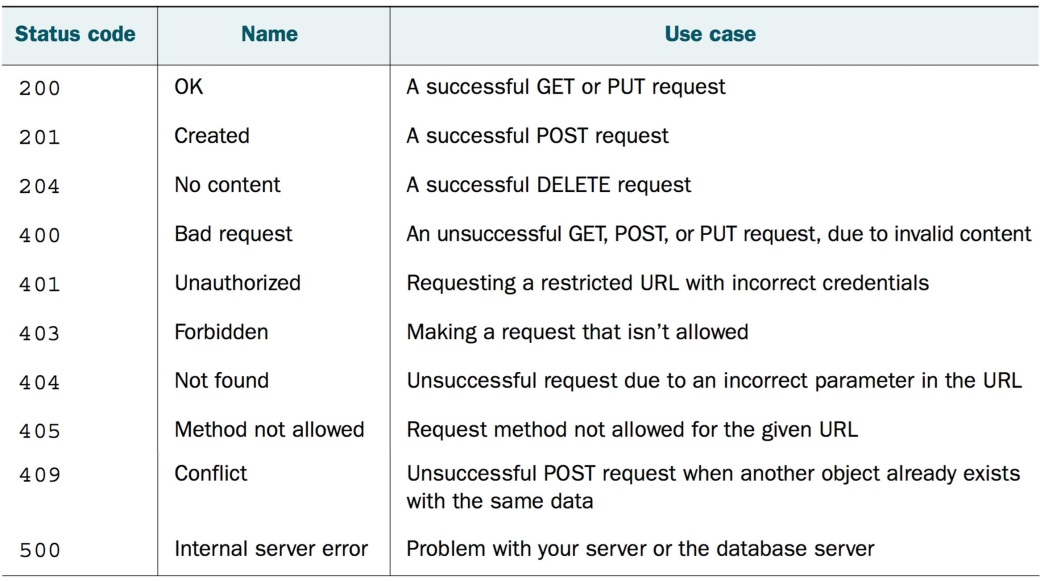
指令置換

replace

wek4 p.12



Content-type又稱為MINE-TIPE(多用途網際網路郵件擴展)



# NPM

管理Node套件的依賴管理系統

## 檢查是否安裝

npm -v

## 建立方式

建立專案，並且切換到目錄底下

* 輸入npm init

更改name, description, entry point(系統會建立package.json，完成npm專案)

* 輸入npm init –y(系統會自動建立一個package.json)

## 安裝檔案到package專案目錄

(moment為範例)

npm install moment --save

## 檔案版本更新

npm只會對minor及patch更新

Patch number2:錯誤修正

Minor number7:新功能

Major number1:重大更新

## 刪除檔案後從新安裝

npm install

## 檔案版本更新

npm update

## 伺服器修改後，自動重新啟動

先安裝npm install -g nodemon

輸入nodemon app.js

# EXPRESS

請求處理(Request Heandling)

中間件(Middleware)

渲染網頁(Rendering HTML)

路由器(Router)

## 安裝EXPRESS

輸入npm I express --save

用i取代install

## 開發階段測試

(以jasmine-node)

輸入npm install jasmine-node --save-dev

## EXPRESS網路監聽



## HTTP請求方式

POST=新增資廖

GET=讀取資料

PUT=更新資料

DELETE=刪除資料

## 快速建立專案

npm I express-generator –g

全域都可以使用的套件

## 建立了一個在express-generator目錄之內的專案

express express-generator

在views目錄裡面預設的template是jade

express -e express-generator

在views目錄裡面的template是ejs

所有的靜態檔案都放在public目錄

## 產生ejs的template

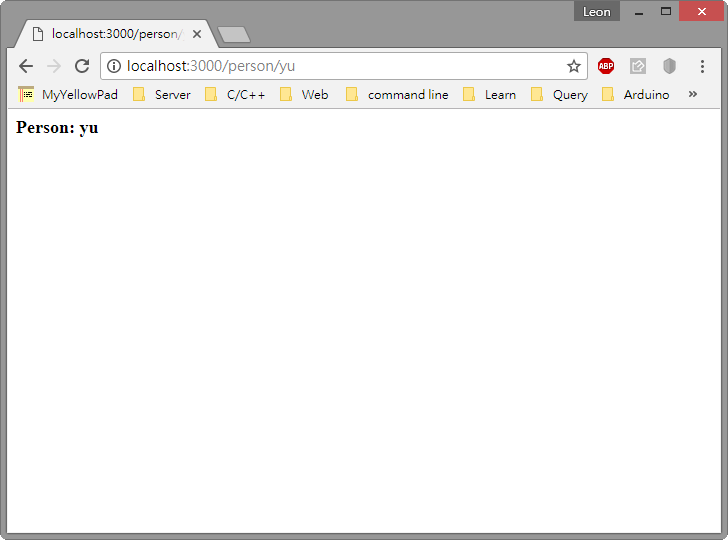
express –h

## 擷取網頁URL

## 範例一

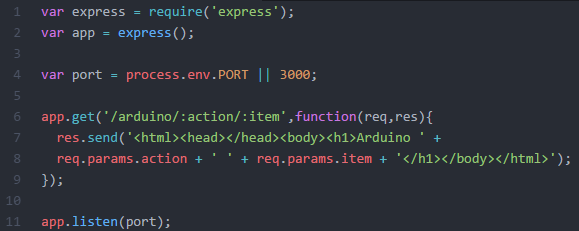


Express的parser把URL視為”/person/:參數”，把參數命名稱id，person之後的yu可以利用req.params裡面的req.params.id取的yu

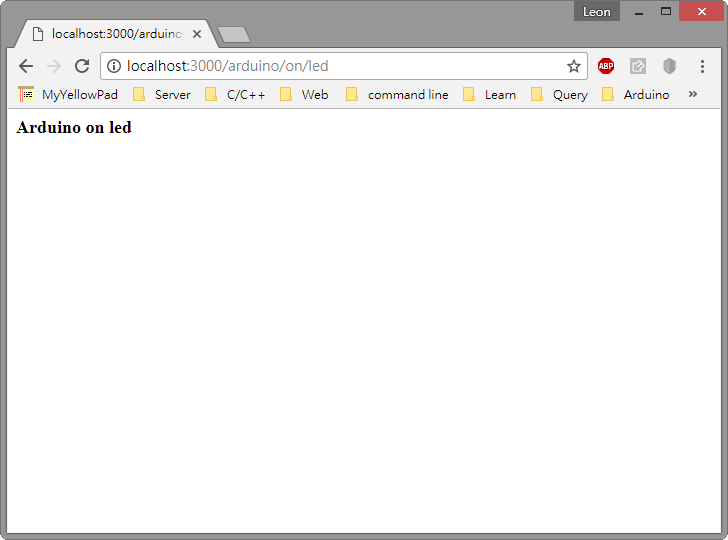


瀏覽器輸入locohost:3000/person/yu

## 範例二



如下將參數1, 2分別命名為action與item，可以利用req.params裡面的req.params.action與req.params.item取得on與led

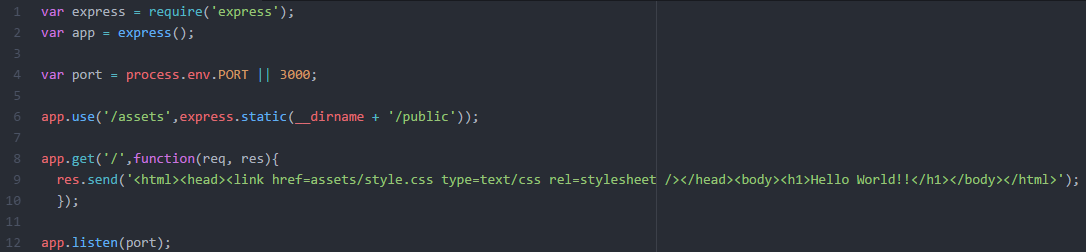


瀏覽器輸入locohost:3000/arduino/on/led

# 靜態網頁

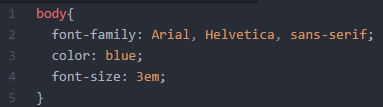
## JS尋找CSS

app.js



app.use(…)是Express的Middleware指令，第一個參數是url，第二個參數是對應動作，當瀏覽器url解析出/assets，就會到express.static(path)指令裡面的path目錄之內，尋找靜態檔案回應瀏覽器。

style.css

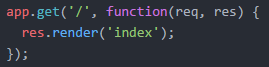


## EJS傳POST資料

npm install ejs --save



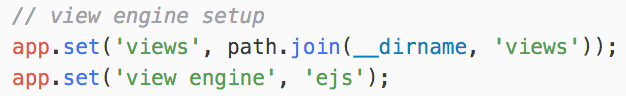
在middleware加入



將res.send改成res.render

Express看到res.render(‘index’)，會到views目錄，找尋index.ejs的檔案翻譯成index.html，再將index.html送回到瀏覽器

EJS預設會去view尋找ejs檔案，可以使用set指定目錄

(wk5 p.35)

* EJS指令以<%...%>包起來，內容被EJS render執行
* EJS參數，以<%...%>包起來，內容會出現在html檔案內

Template可以提供同一個重複使用

範本+資料=html檔案

## SASS改變CSS

SASS是CSS前置處理器，可以讓CSS的開發模組化

使用npm的node-sass-middleware套件，可以在後台江sass→css轉換。

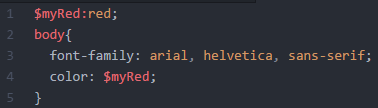
當middleware設定sass的處理，當前端發處對/sassets目錄請求，就進行sass→css轉換動作。

app.js

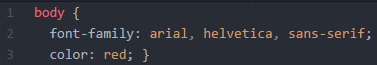


Middleware指令是安照順序執行，所以sass指令要放在靜態當按目錄的指令前

style.scss



style.css



# 動態網頁

前端往後端送資料，常見方式:

* Web Request
* Form Submit
* Ajax

## 取得POST DATA

npm i body-parser --save

bodyParser.urlencoded→可以處理附加POST後的資料

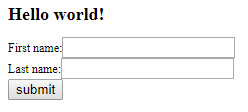
bodyParser.json→可以處理附加Ajax裡面資料

app.js



index.ejs





瀏覽器

(wek5 P.64)