
Table of Contents

Introduction	1.1
RWD prepare	1.2
RWD media query	1.3
RWD usage by Bootstrap	1.4
Bootstrap quick start	1.5

Responsive CSS Dojo

多重載具的時代，人們漸漸將手持裝置取代桌上型攜帶型電腦，甚至已經改變瀏覽的習慣。對於網頁開發者來說多重載具是一個現在進行式，舊有的固定寬度，table 排版，圖片的載入，甚至是表單的呈現都需要思考不同載具的呈現面，讓使用者如何在不同裝置不同螢幕尺寸上方便使用，Responsive 絕對是現代網頁開發者必要的基礎能力，這場 Responsive 除了基本尺寸觀念之外，再將幾個 Responsive 必要觀點帶給各位。

為何要使用 RWD

iPhone 第一代在 2007 年 1 月 9 日，當年 Jobs 在發表會上指出，他正在 reinvent 電話這件事情，距離現在不到 10 年的時間，手持式裝置，大螢幕載具已經是所有人的生活必需品其中之一。

在現代多重載具的狀況下，許多人回到家中幾乎已經不開電腦，使用的大多透過手機，可能是 iPhone, Android 等智慧型電子產品進行瀏覽頁面。

甚至是開始上網，搜尋，閱讀，看文章，玩遊戲等，所有的事情已經在這短短不到 10 年之間開始進行轉變。悄悄的進入到你我的生活之中。

RWD 前端開發必學技術

Google 也在 SEO 操作指南中指出，網站中有適當支援 RWD 將會提高 SEO 的排行，這才是符合人性化的實際設計規範。 (註1)

身為一位開發者就必須知道，目前已經有 80% 的使用者是採用手持式裝置在進行操作網頁，網站，許多人已經不太利用電腦。

開始面對不同載具，不同螢幕解析度，是現代開發者必須要面對的課題，也是網路應用首要討論的方向，對於無法解決人們需求的人，將會即將被淘汰。

指南內容

系列文章中，將會透過實際範例，帶領各位進入到 RWD 的世界當中，從第一個章節瞭解 media query 對於不同載具的使用方法，接著透過 bootstrap 框架，帶領各位開發者進入到 RWD 實戰領域中。

透過不斷的實做，以及瞭解 RWD 特色，越多得實際開發經驗，將會讓自己的網站作品，更容易讓人使用，這是所有開發者的課題，如果在流程上有任何關於 RWD 的問題，都希望你可以留言給我們，透過 github issue 提供我們書籍內容的方向及勘誤，並且協助我們讓內容更加完善。

- <https://github.com/exma-square/rwd-in-real>
- RWD in REAL 發問處

參考資料，

註1 Responsive Web Design

開始進入 RWD 世界

首先歡迎大家開始進入到 RWD 的世界中，首先需要請各位至少準備環境的建立，請先安裝以下軟體

- [Google Chrome browser](#)
- [Web Server - chrome extension](#)

關於 RWD

RWD 全名為『Responsive web design』，主要規範於 W3C 組織中，裡面有許多詳細的解說，用處在於解決手機端應用，視覺解析度上的問題。

同時除了手持式裝置之外，還有許多關於大尺寸螢幕，手錶，電子嵌入式裝置網頁顯示規範等，可以說是除了桌上型，筆記型電腦螢幕之外，目前設定規範都是涵蓋於 RWD 範圍中。

如果對於手機相關規範有興趣者，可以參考 W3C 詳細規範 [Standards for Web Applications on Mobile: current state and roadmap](#)

基礎技術

對於開始進入 RWD 之前，首先讀者必須要瞭解以下幾個技術項目，

- HTML & HTML5
- CSS

至少對於 Desktop browser 的 html / css 相關內容需要有一定程度瞭解，才有辦法進入以下課程內容，課程中將不在闡述關於基礎 HTML tag，attribute，Value，以及 CSS selector，block mode，inline mode 等差異，請各位讀者注意。

本書使用方法

主要內容可以參照[目錄](#)章節規劃進行，在[範例程式中](#)中可以看到每則範例程式都分別存放於不同儲存資料夾中。

可以透過前面所安裝的[Web Server - chrome extension](#)，啟動伺服器，

啟用伺服器流程

啟動 chrome 瀏覽器，透過瀏覽列中輸入

```
chrome://apps/
```

可以看到如下的畫面



點擊 Web Server，會跳出一個視窗，裡面進行選擇 [CHOOSE FOLDER](#)，並且選擇到對應的範例程式資料夾，啟動伺服器讓狀態顯示為 [Web Server: STARTED](#)



Web Server

The screenshot shows the configuration interface for the 'Web Server for Chrome' extension. At the top, a blue header bar displays the extension name and a yellow circular icon with '200 OK!'. Below the header, a message encourages users to leave a review. The main configuration area includes a 'CHOOSE FOLDER' button (set to 'Current: /mocha_test') with a blue toggle switch, and a status message 'Web Server: STARTED'. A list of 'Web Server URL(s)' shows a single entry: <http://127.0.0.1:8887>. The 'Options (needs restart)' section contains several checkboxes: 'Run in background' (unchecked), 'Start on login' (unchecked), 'Accessible on local network' (unchecked), 'Also on internet' (unchecked), 'Prevent computer from sleeping' (unchecked), and 'Automatically show index.html' (unchecked). An 'Enter Port' field is set to '8887', and a link to 'Show Advanced Settings' is present. At the bottom, a note links to reporting problems, stating 'Need to [Report a problem](#)? Open source, MIT license.'

Web Server for Chro… 

Please [leave a review](#) to help others find this software.

CHOOSE FOLDER Current: /mocha_test

Web Server: STARTED

Web Server URL(s)

- <http://127.0.0.1:8887>

Options (needs restart)

Run in background

Start on login

Accessible on local network

Also on internet

Prevent computer from sleeping

Automatically show index.html

Enter Port

8887

[Show Advanced Settings](#)

Need to [Report a problem](#)? Open source, MIT license.

接著就切換回到 Chrome 瀏覽器，網址列中輸入，就可以看到範例程式的結果

```
http://127.0.0.1:8887/index.html
```

開始吧

如果以上都已經瞭解，歡迎開始翻閱接下來的範例及程式碼，歡迎各位開始進入實做 RWD 世界中。

RWD 規範

<https://www.w3.org/TR/css3-mediaqueries/>

RWD 大部分使用內容是根據 W3C 所規範的 media queries 進行實做，而這些規範並不是全新的系統架構，只是針對 HTML4 及 CSS2 本身之前的架構進行的延伸。

HTML4 及 CSS2 中就可以設定 `screen`，`print` 不同用途上有不同的設定，例如使用不同字型，使用不同的字體大小解析度。

所以對於 RWD 的使用方式，大多採用與 CSS2 CSS3 相同的描述方式，使用相同屬性給予方式進行網頁外觀的描述。

RWD 起手式

RWD 主要是以 HTML 以及 CSS3 中 media queries 達成多重螢幕尺寸解析度的處理。

在 HTML5 的標準中，我們可以指定頁面的寬度可以依照裝置的檢視寬度來呈現，再利用 CSS 的 media queries 設定不同螢幕解析度下使用不同的 CSS 結果來呈現頁面。

所以在專案一開始時，就需要進行螢幕尺寸進行規劃，以及設定條件限制，提供不同寬度畫面進行不同 CSS 的設定。

頁面解析度設定

手機裝置中除了上述的 media queries 描述條件之外，需要特別注意對於 `pixel` 此單位的描述。

對於不同螢幕解析度來說，`pixel` 並不代表同樣的單位，例如 iPhone 在 retina 解析度中，一個單位為 4 `pixel`，在進行 media queries 通常都會不如預期。

因此需要透過 `viewports` 進行設定螢幕解析度，可以在 `html` 中的 `head` 之間加入 `viewports`，設定方式如下，

```
<html>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</html>
```

media queries 使用方法

首先在整個 `html` 架構裡面，我們可以透過以下方式進行引用，`media queries` 條件方式，

```
<link rel="stylesheet" type="text/css"
  media="screen and (max-device-width: 480px)"
  href="shetland.css">
```

又或者我們可以透過以下採用類似既有 `CSS` 的方式，但只是加上 `@media` 方式進行描述，

```
@media (query) {
  /* CSS Rules used when query matches */
}
```

以實際使用方式來看，如果需要設定尺寸與尺寸範圍之間，可以如下的設定方式，以下就是設定視窗寬度大於 `400px` 但小於 `700px` 時，套用某一段 `selector` 的屬性，此時屬性權重將會大於原本的設定，所以通常 `media queries` 如果都放置於同一個 `css` 檔案，通常都會置放於下方讓 `css` 描述權重提高。

```
@media screen and (min-width: 400px) and (max-width: 700px) {
  .selector {...}
}
```

如果視窗最小尺寸的時候，可以設定為500px，使用方式可以如下，

```
@media screen and (min-width:500px) {  
    .selector {...}  
}
```

如果視窗為直立，採用方式如下

```
@media screen and (orientation: portrait) {  
    .selector {...}  
}
```

其他使用方法

除了上面的直述式 css 描述方式之外，也可以透過引用的方式來進行，條件及過濾方法同樣，僅會採用某段 css 檔案，

```
<link rel="stylesheet" media="not|only screen and (color)" href="example.css" />
```

某些舊版瀏覽器只能讀media type(screen,projector等等)，不會讀media feature，也不會讀not或only，因此也不會套用此media query後的CSS檔。

RWD 直接套用

如果你跟我一樣不想知道原理，只想要快點使用，可以透過以下的範例直接套用在自己的 css 檔案中，開始進行瀏覽，透過 chrome 瀏覽器，可以透過 開發者模式工具 進行手持式裝置模擬，模擬視窗大小，以及點擊效果，看到實際的 css 編寫結果，

```
/* Smartphones (portrait and landscape) ----- */  
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) {
```

RWD media query

```
/* Styles */  
}  
  
/* Smartphones (landscape) ----- */  
@media only screen and (min-width : 321px) {  
/* Styles */  
}  
  
/* Smartphones (portrait) ----- */  
@media only screen and (max-width : 320px) {  
/* Styles */  
}  
  
/* iPads (portrait and landscape) ----- */  
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) {  
/* Styles */  
}  
  
/* iPads (landscape) ----- */  
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) {  
/* Styles */  
}  
  
/* iPads (portrait) ----- */  
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) {  
/* Styles */  
}  
*****  
iPad 3  
*****  
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : landscape) and (-webkit-min-device-pixel-ratio : 2) {  
/* Styles */  
}
```

```
@media only screen and (min-device-width : 768px) and (max-device-width : 1024px) and (orientation : portrait) and (-webkit-min-device-pixel-ratio : 2) {  
/* Styles */  
}  
/* Desktops and laptops ----- */  
@media only screen and (min-width : 1224px) {  
/* Styles */  
}  
  
/* Large screens ----- */  
@media only screen and (min-width : 1824px) {  
/* Styles */  
}  
  
/* iPhone 4 ----- */  
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) and (orientation : landscape) and (-webkit-min-device-pixel-ratio : 2) {  
/* Styles */  
}  
  
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) and (orientation : portrait) and (-webkit-min-device-pixel-ratio : 2) {  
/* Styles */  
}  
  
/* iPhone 5 ----- */  
@media only screen and (min-device-width: 320px) and (max-device-height: 568px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2){  
/* Styles */  
}  
  
@media only screen and (min-device-width: 320px) and (max-device-height: 568px) and (orientation : portrait) and (-webkit-
```

```
device-pixel-ratio: 2){
/* Styles */
}

/* iPhone 6 ----- */
@media only screen and (min-device-width: 375px) and (max-device-height: 667px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}

@media only screen and (min-device-width: 375px) and (max-device-height: 667px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}

/* iPhone 6+ ----- */
@media only screen and (min-device-width: 414px) and (max-device-height: 736px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}

@media only screen and (min-device-width: 414px) and (max-device-height: 736px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}

/* Samsung Galaxy S3 ----- */
@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : landscape) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}

@media only screen and (min-device-width: 320px) and (max-device-height: 640px) and (orientation : portrait) and (-webkit-device-pixel-ratio: 2){
/* Styles */
}
```

```
ice-height: 640px) and (orientation : portrait) and (-webkit-
device-pixel-ratio: 2){
/* Styles */
}

/* Samsung Galaxy S4 ----- */
@media only screen and (min-device-width: 320px) and (max-dev
ice-height: 640px) and (orientation : landscape) and (-webkit-
-device-pixel-ratio: 3){
/* Styles */
}

@media only screen and (min-device-width: 320px) and (max-dev
ice-height: 640px) and (orientation : portrait) and (-webkit-
device-pixel-ratio: 3){
/* Styles */
}

/* Samsung Galaxy S5 ----- */
@media only screen and (min-device-width: 360px) and (max-dev
ice-height: 640px) and (orientation : landscape) and (-webkit-
-device-pixel-ratio: 3){
/* Styles */
}

@media only screen and (min-device-width: 360px) and (max-dev
ice-height: 640px) and (orientation : portrait) and (-webkit-
device-pixel-ratio: 3){
/* Styles */
}
```

參考資料

- [Use CSS media queries for responsiveness](#)
- [Responsive design – harnessing the power of media queries](#)
- [MEDIA QUERIES FOR COMMON DEVICE BREAKPOINTS](#)

Bootstrap 快速入門 RWD

快速使用 Bootstrap

首先進行安裝 bower

```
npm install -g bower
```

透過 bower 安裝最新的 bootstrap

```
bower install bootstrap
```

安裝後資料儲存架構如下

```
└── bower_components
    ├── bootstrap
    │   ├── CHANGELOG.md
    │   ├── Gemfile
    │   ├── Gemfile.lock
    │   ├── Gruntfile.js
    │   ├── ISSUE_TEMPLATE.md
    │   ├── LICENSE
    │   ├── README.md
    │   ├── bower.json
    │   └── dist
    │       ├── css
    │       │   ├── bootstrap-theme.css
    │       │   ├── bootstrap-theme.css.map
    │       │   ├── bootstrap-theme.min.css
    │       │   ├── bootstrap-theme.min.css.map
    │       │   ├── bootstrap.css
    │       │   ├── bootstrap.css.map
    │       │   └── bootstrap.min.css
```

資料夾中如果有以上架構，表示安裝完成，接著可以新增 `index.html` 網頁，讓 bootstrap 可以生動活潑進行載入。

`index.html`

```
<!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">
    <!-- The above 3 meta tags *must* come first in the head;
    any other head content must come *after* these tags -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
        <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
        <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Step01

因為我們頁面上會需要 container 還有接下來 menu 的加入，因此需要設定自己的 css 檔案將 bootstrap 的預設樣式進行調整，所以會需要進行設定引入 css 檔案方式。

請在 `link bootstrap` 連結底下增加引入 `starter-template.css`

```
<link rel="stylesheet" type="text/css" href=".assert/css/starter-template.css">
```

Step02

在頁面中試著加入 `Bootstrap menu`，將底下這段 nav 的部分加入到 body 開頭之後，設定網站的 menu 內容。

```
<nav class="navbar navbar-inverse navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle collapse" data-toggle="collapse" data-target="#navbar" aria-expanded="false" aria-controls="navbar">
        <span class="sr-only">Toggle navigation</span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Project name</a>
    </div>
    <div id="navbar" class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
      </ul>
    </div>
  </div>
</nav>
```

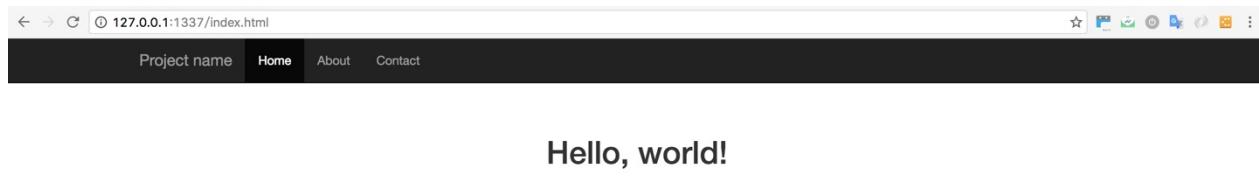
Step03

此時會發現 h1 的內容被 menu 蓋住了，因此需要 container 的協助，透過容器將 h1 包起來。

```
<div class="container">
  <div class="starter-template">
    <!-- Step03-1. 將 h1 貼上到此方下方處 -->
    <h1>Hello, world!</h1>
  </div>
</div>
```

Final

透過 web server 我們測試一下，內容結果應該符合如下的畫面，就表示我們使用 bootstrap 框架正式成功，這也是我們第一個完美的 starter bootstrap page.



同時我們也可以透過手機螢幕測試，或者透過 chrome 模擬手機瀏覽狀態，會看到 RWD menu 的樣式立即呈現在各位面前。

Bootstrap quick start

