

jsp php asp.net ROR 簡單比較

PHP

PHP 主要是用途在於處理動態網頁，主要應用在網頁伺服器，處理使用者的輸入來產生網頁，命令列腳本或是視窗程式介面 (GUI) 的開發也是 PHP 的主要應用範圍。

特色：

1. PHP 是一種嵌入在網頁 HTML 原始檔中的程式語言。利用 PHP 直譯器執行網頁中的 PHP 程式而產生動態的網頁內容。

2. 應用範圍：

#1. 伺服器端腳本語言：PHP 最初就是設計成伺服器端腳本語言，因此這也是 PHP 應用最廣的部份。

#2. 命令列腳本：PHP 也可以用命令列上執行來使用，例如分析紀錄檔、系統管理工作等。

#3. 圖形使用者介面 (GUI)：透過和 GUI 函式庫 GTK+ 的結合，PHP 也可以應用在圖形使用者介面 (GUI) 的開發上，目前的應用還不普遍。

3. 為 Open Source 的軟體。

4. 跨平台開發的相容性。

5. 語法上簡單易懂，參考了 Perl、C 語言，也可整合於 HTML 之中。(以下介紹的其實都可以...)

6. 提供完善的資料庫處理函式，透過這些函式可以在網頁程式中存取大部分的資料庫系統，如：MySQL、ORACLE 等。

PHP 環境建置：

可在任何系統下開發：MAC、LINUX、UNIX、WINDOWS 等。

ASP.NET

ASP.NET 是微軟所提供的網站開發技術。程式設計師可以利用這些技術建置一個網站/頁應用程式。很多人都把 ASP.NET 當做是一種程式語言，但它實際上只是一個由 .NET.Framework 提供的一種開發平台 (development platform)，並非程式語言。

特色：

1. ASP.NET 在運行上會將原始碼編譯成.dll (dynamic link library，動態連結函式庫)，然後再透過 Common Language Runtime 執行，這種方法比傳統的直譯較有效率。
2. 擁有和視窗環境非常相似的 Web 控制項：像是 Button、Label 等等。這些控制項都有各自的事件，除此之外，也可以利用程式碼來設定這些控制項的屬性，可用在網頁應用程式的開發工作，讓兩者之間的轉換更直接。另外也有許多的控制項及類別庫讓應用程式的開發工作更快速。
3. Data Binding：將資料與表單元件連結，.NET 平台會自行處理控制項的所有細節；伺服器控制項的資料來源與資料欄位設定好，其他的畫面呈現細節都由.NET 架構幫你完成。.NET 平台會先處理控制項在畫面 (Web Form) 上產生的 HTML 標籤，然後再把處理的結果送到使用者的瀏覽器中。
4. CodeBehind：也就是代碼分離，在 ASP.NET 的設計中，主要是利用兩個檔來創建一個 ASP.NET 的網頁，其中一個是設計檔，一般以.aspx 為副檔名，.aspx 檔案只是很簡單的 Web Form，定義事件和處理方式等，其 Web Form 中物件的相關處理方法皆儲存於一個獨立的原始碼檔案中。使得程式設計較模組化，增加管理上的方便。
5. 有別於 ASP，導入物件導向的概念，許多存取和運算都可以模組化。
6. 使用者可自訂控制項可以讓多個網頁當作樣板一般套用：例如像網站上的選單。
7. 主要開發語言有 Visual Basic.NET 和 C#兩種，以物件導向為基礎。
8. .NET 的偵錯功能十分的強，自動完成的功能(類似 IE 的自動完成)，方便了程式作業。另外有很多如資料庫連結的物件，都可用拖曳的方式完成，增加整體開發效率。

ASP.NET 環境建置：

平台 WINDOWS

JSP

JSP 是 Java 技術的一部份，建立在 Java Servlet 技術之上，利用 HTML 與 Java 語言來開發動態網頁應用程式。

特色

1. JSP 是由 Java 寫成的物件；一個簡單的描述程式語法，用來處理物件與 JSP 集裝軟體之間的溝通過程。
2. JSP 網頁執行前，需要先經由 Web Server 上的 JSP Container 轉換與編譯成 Java Servlet，才能執行；此轉譯為自動處理，且只在第一次收到網頁要求時執行。
3. 由簡單的描述語言語法以及構成 JSP 標準的 Java 物件。
4. 為離散式以網頁為基礎的應用程式 (Distributed web-based application)，這表示它的邏輯處理過程是分散在不同的主機上面。大部分的處理過程是發生在 JSP 應用程式所在的主機上，額外的邏輯處理時間則是花在觀看網頁的客戶端處理 HTML 檔案的上面。
5. 設計上必須了解結合客戶端以及伺服器端邏輯的網站應用程式的雙面性質。

JSP 環境建置：

須安裝支援 JSP 解譯引擎的伺服器軟體(如 Tomcat、Resin..)外，還需要 JDK 來做最後程式編譯的動作

Ruby on Rails (ROR)

Ruby，一種為簡單快捷，為了物件導向而創的腳本語言，由日本人松本行弘開發，Ruby on Rails 是用 Ruby 寫的網頁應用程式開發框架(web application framework)，一般簡稱

Rails 或是 RoR，在此架構下，目標是要加快以資料庫驅動的網站的開發速度與簡化過程，以兩個重大原則為目標：

1. 絕不做重覆的事，(DRY, Don' t Repeat Yourself)，程式設計師要寫的程式碼愈少愈好，因為程式碼少的話，出錯的機率也會降低，最後維護起來也容易。
2. 用慣例(通則)取代設定(Convention over Configuration)。

MVC 的架構：

RoR 的基本架構是 MVC(Model-View-Controller)，Model 處理和資料庫之間的溝通，View 則是產生使用者介面，使用者透過 View 的頁面發出要求，Controller 則是處理使用者的要求，若要使用到資料庫，則和 Model 溝通，由 Model 去進行資料庫的存取，再由 View 根據結果產生使用者看到的頁面。依照這樣的處理模式，資料的處理和使用使用者介面不會互相干擾，程式較容易撰寫也較容易維護。

ruby 語言特色：

1. 完全物件導向：在 Ruby 語言中，任何東西都是對象，包括其他語言中的基本數據類型，比如整數變數沒有類型：Ruby 的變數可以保有任何類型的數據。不需要宣告即可使用。
2. 任何東西都有值：不管是數學或者邏輯表達式還是一個語句，皆都有值。
3. Ruby 的變數有一定簡單易懂的規則。
4. 已經定義的類別可以在運行時修改：Ruby 是動態語言，你可以在程序中修改先前定義過的類別。也可以在某個類別的實例中定義該實例特有的方法，這叫做單立方法。
5. 自動化的記憶體管理：擁有自動化的 garbage collection，使人們不需要花心思在記憶體管理上。