Understanding ActionScript Syntax

فهم اسلوب لغة الاكشن اسكربت

Understanding Packages مفهوم الحزم البرمجية

ان معظم الخلايا وضعت منظمة فى تركيب يدعى حزم برمجية و لكى تفهم لغة الاكشن اسكربت يجب ان تفهم مفهوم الحزم .
ان الحزم البرمجية هى تجميع لمجموعات من الخلايا المتقاربة فى الغرض لكى يمكنك استخدامها داخل برامجك فى نطاق Scope مجال معين .
و تتسأل الان ما هو Scope ؟
انة مفهوم يوضح مدى رؤية الكائنات و المتغيرات بالنسبة لمراحل حياة البرنامج الرئيسى و الخلايا التى يتكون منها و لنفترض مثلا انك لديك متغير فانك من خلال معرفتك مدى او مجالة يمكنك ان تعرف اين هو بالضبط داخل الكود و كيف تستدعية و تسند لة قيمة اخرى .
و لكن ما فائدة ال Scope بالنسبة للحزم البرمجية ؟
الحزم البرمجية تسمح لك بعمل مجموعة من الخلايا بنفس الاسم لكن بشرط استخدامهم فى مجالات مختلفة اى Different Scope.
مثال Button Class هى جزء من الحزمة البرمجية mx.Controls اذن المسار الكامل للزر هو

```
mx.controls.Button
```

و علية فاذا كنت تريد عمل خلية زر اخرى فى حزمة برمجية اخرى دون ان تخاف ان يحدث تعارض بين الخلية الجديدة و الخلية القديمة mx.controls.Button
مثال على تعريف زر من الخلية Button فهناك اكثر من طريقة منها :

```
var button:mx.controls.Button;
```

او باستخدام Constructor

```
button = new mx.controls.Button( );
```

و يمكنك ايضا تعريف متغير من النوع زر من خلال استيراد الخلية زر

```
import mx.controls.Button;
```

و لكن فى هذة الحالة اذا قمت بتعريف زر اخر من خلية اخرى يجب ان تتعامل مع كلا منهما بالاسم الكامل اى بهذة الطريقة

```
var button:mx.controls.Button;
```

Declaring Classes الاعلان عن الخلايا

فى البداية يجب ان تقوم بعمل ملف يحتوى على كود الخلية و يكون امتداد هذا الملف as و يشترط ان يكون اسم الملف هو نفس اسم الخلية مثلا اذا كنت تريد عمل خلية اسمها Example فيجب ان يكون اسم الملف الخاص بها هو Example.as
مثال :

```
package com.example {
import flash.net.URLLoader;
import flash.net.URLRequest;
public class Example {
// الاكواد الخاصة بالخلية تكتب هنا
}
}
```