

Catherine Szaibrum

Aide-mémoire

**ActionScript
pour Flash MX
et Flash MX 2004**

© Editions OEM (Groupe Eyrolles),

ISBN 2-7464-0495-8

OEM

EYROLLES

Dynamic

Mot-clé

Flash Player 6, version 2

Classe dynamique

■ Syntaxe

```
dynamic class nomDeLaClasse [ extends superClass ]  
    [ implements nomDinterface [, nomDinterface... ] ]  
{  
    //définition de la classe ici  
}
```

Voir aussi : **Classe**, **Extends**

■ Explication

Autorise l'ajout de propriété dynamique aux objets de la classe précisée dans **nomDeLaClasse**.

■ Exemple

```
//la classe maClasse.as  
dynamic class maClasse extends class_A { //Elle est dynamique  
    function maClasse() { //Constructeur  
    }  
}
```

Extends**Mot-clé**

Flash Player 6, version 2

Sous-classe

■ Syntaxe**class** nomDeLaClasse **extends** autreNomDeClasse {}**interface** nomDinterface **extends** autreNomDinterface {}Voir aussi : **Classe**, **Implements**, **Interface****■ Explication**

Permet de définir la sous-classe d'une classe ou d'une interface qui hérite de toutes les méthodes et propriétés de la superclasse. La classe de début est la super classe.

nomDeLaClasse est le nom de la sous-classe.

autreNomDeClasse est le nom de la classe sur laquelle la sous-classe est basée.

nomDinterface est le nom de l'interface définie.

autreNomDinterface est le nom de l'interface sur laquelle l'interface est basée.

■ Exemple

```
class sousClasse extends class maClasse
{
    function sousClasse() { //Constructeur de sousClasse
    }
    function m():Number {renvoie 54;}
    function o(s:String):Void {trace(s);}
}
```

Implements**Mot-clé**

Flash Player 6, version 2

Classe d'implémentation

■ Syntaxe**maClasse implements** interface01 [, interface02, ...]Voir aussi : **Classe**, **Extends**, **Interface**

■ Explication

Permet de définir une classe qui contiendra les implémentations des méthodes d'une interface.

Import

Mot-clé

Flash Player 6, version 2

Importation de classe

■ Syntaxe

```
import nomDeLaClasse
```

```
import nomPaquet.*
```

■ Explication

Permet d'importer une classe dans Flash.

nomDeLaClasse est le nom de la classe ainsi que le chemin d'accès à celle-ci.

nomPaquet est le dossier stockant les fichiers de classe liés entre eux.

Interface

Mot-clé

Flash Player 6, version 2

Classe d'implémentation

■ Syntaxe

```
interface nomDinterface {}
```

```
interface nomDinterface [extends nomDinterface [,  
nomDinterface ...] {}
```

Voir aussi : **Classe**, **Extends**, **Implements**

■ Explication

Permet de définir une interface. Nous vous rappelons qu'une interface est une classe qui comprend des déclarations de méthodes et des membres publics.

MMExecute()**Fonction**

Flash Player 6, version 2

Commande API Flash JavaScript

■ Syntaxe

```
MMExecute("Commande API Flash JavaScript;")
```

■ Explication

Permet d'émettre des commandes de l'API Flash JavaScript à partir d'ActionScript. A partir de l'API Flash JavaScript (JSAPI), vous pouvez dupliquer ou émuler des commandes saisies par l'utilisateur.

Commande API Flash JavaScript est la commande à utiliser dans le fichier Flash JavaScript.

■ Exemple

```
var libe:Array =
MMExecute("fl.getDocumentDOM().epicerie.items;");//Dans un
tableau, amène le document epicerie.itemms
trace(libe.length + "articles de l'épicerie");//Affiche les articles de
l'épicerie
```

MMExecute()**Mot-clé**

Flash Player 6, version 2

Variable ou fonction privée

■ Syntaxe

```
class nomClasse{
  private var nom;
  private function nom() {
    // vos instructions
  }
}
```

Voir aussi : **Public, Static**

■ Explication

Permet de restreindre l'accès à une variable ou une fonction afin que celle-ci soit reconnue uniquement dans la classe ou la sous-classe qui la définit. Notez que seules les classes ou les sous-classes peuvent présenter cette restriction contrairement aux interfaces.

nom est le nom de la variable ou de la fonction privée.

PublicMMExecute()

Mot-clé

Flash Player 6, version 2

Variable ou fonction publique

■ Syntaxe

```
class NomClasse {
    public var nom;
    public function nom() {
        // vos instructions
    }
}
```

Voir aussi : **Private**, **Static**

■ Explication

Permet d'autoriser l'accès à une variable ou une fonction afin que celle-ci soit considérée comme publique ce qui implique que toutes les classes et sous-classes la reconnaissent.

StaticMMExecute()

Mot-clé

Flash Player 6, version 2

Variable ou fonction statique

■ Syntaxe

```
class NomClasse {
    static var nom;
    static function nom() {
```

```
// vos instructions
}
}
```

Voir aussi : **Private**, **Public**

■ Explication

Permet de définir que la variable sera appliquée à tous les objets d'une classe.

nom est le nom de la variable ou de la fonction statique.

Camera

Classe

Flash Player 6, version 2

Capture vidéo

Classe Constructeur Propriété Méthode Événement

■ Explication

Cette classe permet de capturer une vidéo à partir d'une caméra vidéo connectée à un ordinateur qui exécute Macromedia Flash Player.

■ Méthodes

get

setMode

setMotionLevel

setQuality

■ Propriétés

activityLabel

bandwidth

currentFps

fps

height

index

motionLevel

motionTimeOut

muted

name
 names
 Quality
 width

■ Événements

onActivity
 onStatus

Get

Camera

Flash Player 6

Référence une caméra

Objet Constructeur Propriété **Méthode** Événement

■ Syntaxe

Camera.get([index])

■ Explication

Renvoie une référence à la caméra de capture de vidéo. Notez que pour réaliser la capture, vous devez commencer par lier la caméra à un objet avec **Video.attachVideo**. Pour faire référence à la même caméra, vous devez utiliser **camera.get** et non **new camera**. Renvoie la référence de la caméra par défaut ou celle disponible. Si **null**, la caméra est occupée ou non installée.

De plus, le visiteur pourra choisir, au travers d'une boîte de dialogue, d'activer la caméra (voir **Camera.onStatus** et **Camera.muted**).

index est un entier de base zéro(facultatif) qui spécifie la caméra à récupérer, en référence au tableau de la propriété **Camera.names**.

SetMode

Camera

Flash Player 6

Dimension du mode de capture

Objet Constructeur Propriété **Méthode** Événement

■ Syntaxe

`active_cam.setMode(largeur, hauteur, fps [,Taillefavor])`

Voir aussi : `Camera.fps`, `Camera.currentFps`, `Camera.height`, `Camera.width`

■ Explication

Permet de définir la dimension du mode de capture de la caméra sur le mode natif qui convient le mieux. Notez que Flash omet les images de sorte que la taille de l'image soit conservée. Pour minimiser le nombre d'images omises, même s'il faut pour cela réduire la taille de l'image, définissez le paramètre **Taillefavor** sur **false**. Pour déterminer les valeurs affectées à ces propriétés, dès que Flash a sélectionné le mode qui correspond le mieux aux valeurs requises, utilisez `Camera.width`, `Camera.height` et `Camera.fps`.

largeur est la largeur en pixels, la valeur par défaut est 160.

hauteur est la hauteur en pixels, la valeur par défaut est 120.

fps est le nombre d'images par seconde que la caméra doit capturer., la valeur par défaut est 15.

Taillefavor (facultatif) est une valeur booléenne spécifiant le mode de redimensionnement si le mode natif correspondant aux conditions requises n'est pas proposé par la caméra. **True** permet de conserver la taille de la capture.

SetMotionLevel

Camera

Flash Player 6

Précise le mode d'invocation de `Camera.onActivity`

Objet Constructeur Propriété **Méthode** Événement

■ Syntaxe

`active_cam.setMotionLevel(sensibilité [, timeout])`

Voir aussi : `Camera.activityLevel`, `Camera.motionLevel`, `Camera.motionTimeOut`, `Camera.onActivity`

■ Explication

Mouvement nécessaire à l'invocation de la caméra si `Camera.onActivity` est **true**.

Précise le temps avant que Flash ne considère que les mouvements sont stoppés et invoque **Camera.onActivity** en **false**.

Notez que :

- ⇒ Pour bloquer la caméra, vous devez définir la sensibilité à 100.
- ⇒ Pour définir le nombre de mouvements, utilisez la propriété **Camera.activityLevel**.

sensibilité est la valeur numérique spécifiant la quantité de mouvements nécessaires pour invoquer **Camera.onActivity** sur **True**. Les valeurs sont comprises entre 0 et 100.

timeout est le nombre de millisecondes (facultatif) avant que **Camera.onActivity** soit **false**.

SetQuality

Camera

Flash Player 6

Bande passante maximale

Objet Constructeur Propriété **Méthode** Événement

■ Syntaxe

active_cam.setQuality(bandePassante, QualitéDimage)

Voir aussi : **Camera.bandwidth**, **Camera.quality**

■ Explication

Permet de définir la bande passante maximale par seconde ou la qualité d'image. Pour indiquer que l'utilisation de la bande passante est prioritaire, définissez une valeur pour **bandePassante** et paramétrez **qualitéDimage** sur 0. Pour indiquer que la qualité est prioritaire sur la bande passante, définissez **bandePassante** sur 0 et indiquez une valeur numérique pour **qualitéDimage**.

bandePassante est un entier précisant la bande passante maximale que la vidéo utilise, en octets par seconde.

qualitéDimage est un entier précisant le niveau nécessaire de qualité d'image (montant de compression d'image). Les valeurs sont comprises entre 1 (qualité minimale, compression maximale) et 100 (qualité maximale, pas de compression).

ActivityLevel**Camera**

Flash Player 6

Quantité de mouvements détectés par la caméra

Objet Constructeur **Propriété** Méthode Événement**■ Syntaxe**`active_cam.activityLevel`Voir aussi : `Camera.motionLevel`, `Camera.setMotionLevel`**■ Explication**

Entier indiquant la quantité de mouvements que la caméra détecte. Les valeurs sont comprises entre 0 (aucun mouvement détecté) et 100 (nombreux mouvements détectés).

Bandwidth**Camera**

Flash Player 6

Bande passante maximale en octets

Objet Constructeur **Propriété** Méthode Événement**■ Syntaxe**`active_cam.bandwidth`Voir aussi : `Camera.setQuality`**■ Explication**

Entier spécifiant la bande passante maximale que la vidéo peut utiliser, en octets par seconde. Si 0, implique que la vidéo peut utiliser autant de bande passante que nécessaire pour conserver la qualité d'image souhaitée. Pour définir cette propriété, utilisez `Camera.setQuality`.

■ Exemple

```
if(maCam.bandwidth >= 32768){//Si la bande passante est supérieure à 32 Ko
    loadMovie(«splat.swf»,_root.hiddenvar);//Charge la vidéo splat.swf
}
```

CurrentFps**Camera**

Flash Player 6

Taux de capture

Objet Constructeur **Propriété** Méthode Événement**■ Syntaxe**`active_cam.currentFps`Voir aussi : **Camera.fps**, **Camera.setMode****■ Explication**

Taux de capture (lecture seule). Pour définir la cadence maximale, utilisez **Camera.setMode**.

Fps**Camera**

Flash Player 6

Taux de capture maximum

Objet Constructeur **Propriété** Méthode Événement**■ Syntaxe**`active_cam.fps`Voir aussi : **Camera.curentFps**, **Camera.setMode****■ Explication**

Taux de capture maximum (lecture seule). Dépend des capacités de la caméra. Pour définir la valeur souhaitée de cette propriété, utilisez **Camera.setMode** et, pour le taux auquel la caméra capture les données, utilisez la propriété **Camera.currentFps**.

Height**Camera**

Flash Player 6

Hauteur de la capture

Objet Constructeur **Propriété** Méthode Événement**■ Syntaxe**`cam_active.height`Voir aussi : **Camera.setMode**, **Camera.width**

■ Explication

Hauteur de capture, en pixels. Pour définir la valeur de cette propriété, utilisez **Camera.setMode**.

Index

Camera

Flash Player 6

Index de la caméra

Objet Constructeur **Propriété** Méthode Événement

■ Syntaxe

active_cam.index

Voir aussi : **Camera.get**, **Camera.names**

■ Explication

Entier basé sur zéro qui précise l'index de la caméra, comme mentionné dans le tableau renvoyé par **Camera.names**.

■ Exemple

```
ma_cam = Camera.get(index);
```

MotionLevel

Camera

Flash Player 6.

La quantité de mouvements nécessaire pour invoquer **Camera.onActivity(true)**

Objet Constructeur **Propriété** Méthode Événement

Voir aussi : **Camera.activityLevel**, **Camera.onActivity**, **Camera.onStatus**, **Camera.setMotionLevel**

■ Syntaxe

active_cam.motionLevel

■ Explication

Entier qui précise la quantité de mouvements nécessaire pour invoquer **Camera.onActivity** sur **True**. Les valeurs sont comprises entre 0 et 100.