# Découvrir l'ActionScript

1. A quoi sert l'ActionScript ?
2. Comment scripter une animation ?
3. Qu'est-ce qu'un gestionnaire ?
4. Nomenclature
5. Qu'est-ce que la syntaxe pointée
6. Apprendre à utiliser les raccourcis clavier
7. Les onglets de la fenêtre Actions

## 1 A quoi sert l'ActionScript ?

Vous savez réaliser des animations en utilisant le scénario et vous souhaitez à présent ajouter de l'interactivité. Contrôler la tête de lecture ou des occurrences sur la scène, lire des sons ou des vidéos, ce sont autant de possibilités offertes par l'ActionScript.

Ce langage permet, d'un clic sur l'occurrence d'un bouton, d'un clip, ou avec le clavier, de contrôler le déroulement d'une animation. Vous allez devoir rédiger des scripts, c'est à dire écrire des lignes d'instructions avec une syntaxe très précise, dans la fenêtre Actions, réservée à cet usage.

## 2 Comment scripter une animation ?

Il existe de nombreuses méthodes pour scripter une animation. Sans entrer dans les détails, voici les "endroits" où vous pouvez placer des scripts.

- Dans un fichier texte externe
- Sur une image clé
- Sur l'occurrence d'un clip d'un bouton, mais plus dans la version 3.0 de l'ActionScript
- Sur l'occurrence d'un bouton, mais déjà déconseillé dans la version 2.0 de l'ActionScript

Nous allons découvrir les méthodes, avantages et inconvénients des 3 dernières possibilités. La première, fichier externe, ne sera pas abordée car cette technique est principalement utilisée dans une programmation plus lourde et/ou plus performante, ce qui n'est pas le but premier de ce module. Même si …

Avant d'aborder les différentes méthodes, il faut insister sur le fait qu'un script a une structure précise, une syntaxe à respecter. Un script peut contenir un gestionnaire qui va permettre de définir une relation entre l'utilisateur et l'ordinateur A nous d'écrire correctement cette relation.

### 2.1 Sur une image clé

C'est la technique à retenir ! En effet, placer un script sur une image clé permet de rendre interactive l'animation, avec ou sans intervention de l'utilisateur.

De toutes façons, il est impossible de poser un script sur une autre image que clé !

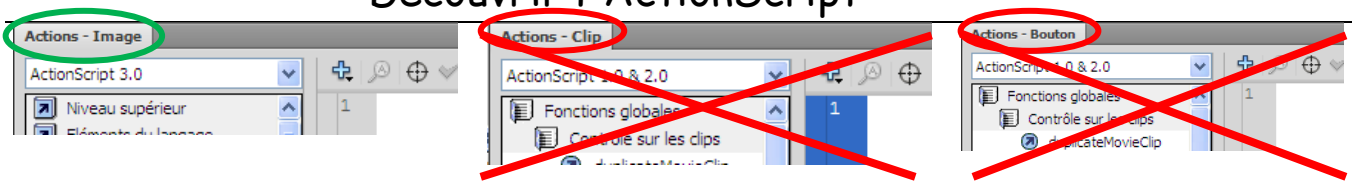
> **Méthodologie : Placer un script sur une image clé**

- Sélectionnez une image-clé sur un scénario
- Cliquez dans la fenêtre "Actions"
- Saisir les lignes d'instructions du script

Remarque : Il est bien écrit "sur **un** scénario" et non "sur **le** scénario" car il existe autant de scénarios que de symboles dans la bibliothèque en plus du scénario de l'animation

ATTENTION : Si vous avez choisi comme format d'animation AS 2.0, au démarrage, lorsque vous placez un script sur une image clé, assurez-vous que le haut de la fenêtre "Actions" affiche bien "Actions - Image" et non "Actions - Clip" ou Actions - Bouton".

# Découvrir l'ActionScript




Avantages : Les scripts sont faciles à modifier car faciles à trouver dans une animation. Un petit "a" indique dans le scénario qu'un script se trouve sur une image clé précise (Sur une image clé ou une sur une image clé vide ).

Inconvénient : Pour les novices, cette technique est un peu déroutante car le code est "détaché" de l'occurrence à laquelle il fait référence.

Exemple :

- Ouvrez le fichier Exercice_1.fla, après avoir décompressé le fichier Exos_les_flas.zip présent dans \\KAPOUER\SHS\PUB\Flash, ou téléchargé à partir du blog.
- Testez l'animation, avec le raccourci CTRL+ENTREE : un texte s'affiche au bout de 15 secondes
- Fermez la fenêtre d'aperçu de l'animation.
- Sélectionnez l'image clé 25 du calque « Actions », déployez la fenêtre Actions (F9).

```
1   gotoAndPlay(1) ;
```

- Testez votre animation : le texte n'apparaît plus
- Fermez la fenêtre d'aperçu
- Le a est apparu dans la case de l'image clé 25 : du code est associé à cette image clé
- Fermez ce fichier après l'avoir enregistré.

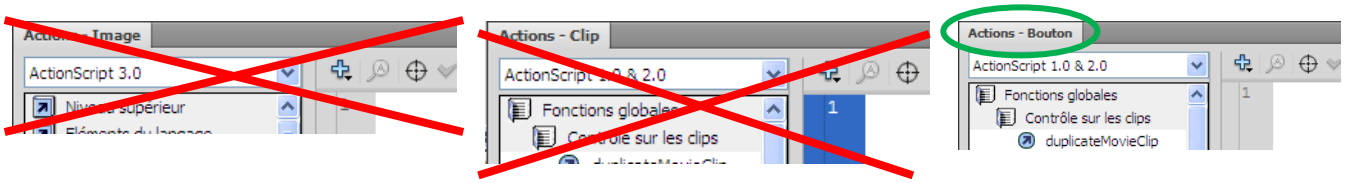
## 2.2 Sur l'occurrence d'un bouton

Cette méthode reste encore valable aujourd'hui, A CONDITION de dégrader le script en version AS 1.0 et AS 2.0, liste déroulante disponible en haut, à gauche, de la fenêtre Actions, mais il est fortement conseillé de placer les scripts que vous souhaitez associer à un événement lié à un bouton (survol, clic, etc.) sur une image clé. Mais il faudra y revenir …

### > Méthodologie : Placer un script sur l'occurrence d'un bouton

- Sélectionnez une occurrence de bouton sur la scène
- Cliquez dans la fenêtre "Actions"
- Saisir les lignes d'instructions du script

Remarque : Assurez-vous que le haut de la fenêtre Actions contient bien le terme "Bouton" comme dans la troisième copie d'écran ci-dessus, avant de saisir le script.




Pour un bouton, vous devez utiliser un gestionnaire (voir 3. Qu'est-ce qu'un gestionnaire ?), commande **on**. Voici à quoi doit ressembler un script placé sur l'occurrence d'un bouton :

# Découvrir l'ActionScript

```
on(press) {
    Ligne d'instructions ;
    Ligne d'instructions ;
}
```

Avantage : Pour les novices, le script est placé sur l'occurrence du bouton, la compréhension de la relation script/occurrence se fait plus facilement.

Inconvénients : La mise à jour/correction d'une animation se fait plus difficilement car il faut d'abord trouver les occurrences sur lesquelles se trouvent les scripts à modifier. L'impression des scripts doit se faire en plusieurs fois (autant de fois qu'il y a d'occurrences contenant des scripts).

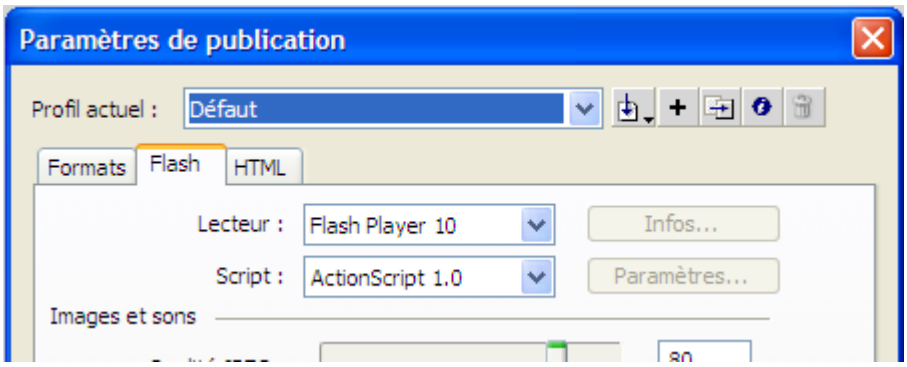
Exemple :

- Ouvrez le fichier Exercice_2.fla
- Testez l'animation : un bouton est visible dans l'animation, mais ne fonctionne pas
- Fermez la fenêtre d'aperçu
- Sélectionnez le bouton placé sur le calque « LeBouton » (occurrence nommée « monBouton » de « B_monBouton »)
- déployez la fenêtre Actions

```
on(press) {
    gotoAndPlay(30) ;
}
```

- Testez votre animation
- Fermez la fenêtre d'aperçu
- Fermez ce fichier après l'avoir enregistré.

Remarque : pour pouvoir fonctionner, cette animation a été rendue compatible AS 1.0, par l'intermédiaire des paramètres de publication : Fichier\Paramètres de publication, onglet Flash et dans la liste pour Script : ActionScript 1.0 .




Bien évidemment, un tel choix ne permet plus de bénéficier des instructions AS3 !, ou si vous passez en AS3, le bouton ne fonctionnera plus et générera un message d'erreur. Mais la simplicité du langage peut séduire !

Voici ce que cela donne, en version AS3, fichier Exercice_2_AS3.fla, le script est posé dans l'image-clé1 du calque « Actions ».

```
1   Object(root).monBouton.addEventListener(MouseEvent.MOUSE_UP,
    passerImage30) ;
2   function passerImage30(evt:MouseEvent) :void {
3       gotoAndPlay(30) ;
4   }
```

Cela est bien sûr moins parlant ! Mais pas d'inquiétude, on va y arriver !

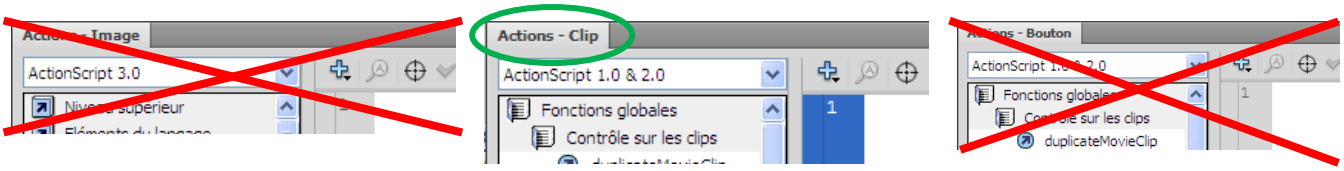
## 2.3 Sur l'occurrence d'un clip

Cette méthode reste encore valable aujourd'hui, mais encore moins recommandée que celle d'un script placé sur l'occurrence d'un bouton. Cette méthode est historique, Flash 4 et 5, déjà avec la version Mx et les suivantes, d'autres possibilités sont offertes. Voici la technique, pour le cas où …

### > Méthodologie : Placer un script sur l'occurrence d'un clip

- Sélectionnez une occurrence de clip sur la scène
- Cliquez dans la fenêtre "Actions"
- Saisir les lignes d'instructions qui composent votre script

Remarque : Assurez-vous que le haut de la fenêtre Actions contient bien le terme "Clip" comme dans la deuxième copie d'écran ci-dessus, avant de saisir votre script.




Pour un clip, vous devez utiliser un gestionnaire (voir 3. Qu'est-ce qu'un gestionnaire ?), commande **on**. Voici à quoi doit ressembler un script placé sur l'occurrence d'un clip :

```
1   onClipEvent(enterFrame) {
2       Ligne d'instructions ;
3       Ligne d'instructions ;
4   }
```

Avantage : Il n'en existe plus vraiment aujourd'hui, si ce n'est que le script est rattaché à l'occurrence

Inconvénients : Identiques à ceux d'un script sur l'occurrence d'un bouton : La mise à jour/correction d'une animation se fait plus difficilement car il faut d'abord trouver les occurrences sur lesquelles se trouvent les scripts à modifier. L'impression des scripts doit se faire en plusieurs fois (autant de fois qu'il y a d'occurrences qui contiennent des scripts).

Exemple :

- Ouvrez maintenant le fichier Exercice_3.fla .
- Testez l'animation : un disque bleu clignote dans l'animation, mais c'est tout
- Fermez la fenêtre d'aperçu
- Sélectionnez le clip placé sur le calque « LeMoteur » (occurrence nommée « monMoteur » de « MC_moteur »)
- déployez la fenêtre Actions

```
1  onClipEvent(enterFrame) {
2      _root.info_txt.text = _root._currentframe ;
3  }
```

- Testez votre animation : le texte affiche le numéro de l'image en cours
- Fermez la fenêtre d'aperçu
- Fermez ce fichier après l'avoir enregistré.

Bien évidemment, pour « fonctionner », ce fichier a été écrit en AS1 et publié en AS1. L'évolution du langage en AS3 nous donne la version du fichier Exercice_3_CS3.fla, disponible lui aussi, pour tester, et pour lequel, le code AS3 est le suivant :

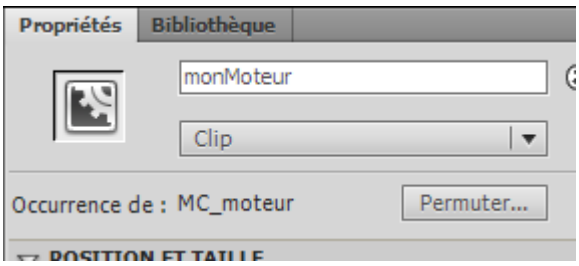
```
1  Object(root).monMoteur.addEventListener(Event.ENTER_FRAME,
   affiche) ;
2  function affiche(evt:Event):void {
3      Object(root).info_txt.text = Object(root).currentFrame ;
4  }
```

### 2.4 Nommer une occurrence.

Cette phase est des plus importantes dans Flash car la programmation objet de Flash requière un nom pour chaque occurrence de l'animation. Pour contrôler une occurrence ou l'utiliser comme zone cliquable sur la scène, il est indispensable de la nommer afin de la différencier des autres.

> **Méthodologie : Nommer une occurrence**

- Sélectionnez une occurrence sur la scène
- Cliquez dans la zone de saisie du nom de l'occurrence dans la palette des Propriétés
- Donnez un nom




Pas de nom avec des trous, utilisez le tiret bas, ce nom sera utilisé pour pointer l'objet, il doit être facilement identifiable.

## 3 Qu'est-ce qu'un gestionnaire ?

Vous pouvez bouger la souris dans tous les sens, cliquer sur l'un de ses boutons, faire un simple, clic, un double clic, un clic maintenu, etc.. La souris peut être au-dessus d'un objet, sortir de cet objet, … Ces opérations sont autant d'événements souris qui peuvent être interceptées par logiciel.

Pour illustrer ceci, lancez le fichier Exercice_4.SWF placé où vous savez.

Le survol évoqué ci-dessus se décompose plus précisément en 2 instants : lorsque le curseur de la souris entre sur une occurrence et lorsqu'il en ressort.
Pour chaque « événement » une instruction est exécutée.
Le code ActionScript associé à l'image 1 de l'animation est le suivant :

```
1  zone1.onRollOver = function() {
2      _root.texteAction.text = "Survol de la zone" ;
3  }
4
5  Zone1.onRollOut = function() {
```

```
6       _root.texteAction.text = "Sortie de la zone" ;
7   }
8
```

Un gestionnaire est donc une partie d'un script qui a pour fonction de surveiller un objet et une interaction de la souris, et d'exécuter la, ou les instructions, prévues dans ce cas. On parle de programmation événementielle.

Un gestionnaire se compose d'une syntaxe très précise qui varie tout de même selon le résultat espéré.

Il serait très simple de faire « réagir » notre zone de test à un bouton gauche de la souris, pour faire afficher un autre message dans la zone de texte dynamique. Par exemple :

```
9   Zone1.onRelease = function() {
10      _root.texteAction.text = "Gauche relâché dans la zone" ;
11  }
12
```

Tout ceci étant valable pour de l'AS 1.0.

Autre exemple :

- Lancez le fichier Exercice_5.SWF contenu dans l'archive Exo_les_swfs.zip placé dans \\KAPOUER\SHS\PUB\Flash ou téléchargeable sur le blog.
- Lorsque l'occurrence intitulée zone1 est survolée, un gestionnaire utilisant l'événement onRollOver change son échelle à 125%. Lorsque le curseur n'est plus sur l'occurrence, un deuxième gestionnaire utilisant un autre événement, onRollOut, change à nouveau l'échelle à 100%.

```
1   zone1.onRollOver = function() {
2       this._xscale =125;
3       this._yscale =125;
4   }
5
6   Zone1.onRollOut = function() {
7       this._xscale =100;
8       this._yscale =100;
9   }
```

Remarque : Notez dans cet exemple que le mot clé **this** fait référence au nom de l'occurrence à laquelle il est fait référence dans le gestionnaire, zone1 dans le cas présent. Vous pourriez dire, "cette" occurrence sur laquelle le rollover (survol) se fait.

Maintenant que vous commencez à entrevoir la façon de fonctionner d'un gestionnaire, il va falloir passer à la structure des gestionnaires sous AS 3.0.
Un gestionnaire est donc une structure qui se compose ainsi :

Objet-à-surveiller.addEventListener(type-événement.EVENEMENT, fonction-exécutée)

La première partie du gestionnaire est généralement un nom d'occurrence mais pour être plus précis, il s'agit d'un nom d'instance. Vous pouvez aussi avoir le mot clé **this** ainsi que **root** mais qui font maintenant référence à l'animation principale. Le fichier Exercice_5_AS3.fla montre la version AS3 de l'exemple précédent. Le code y est le suivant :

```
1   this.zone1.addEventListener(MouseEvent.MOUSE_OVER, survol);
```

```
this.zone1.addEventListener(MouseEvent.MOUSE_OUT, sortie);
function survol (evt :MouseEvent) {
    this.zone1.scaleX =1.25;
    this.zone1.scaleY =1.25;
}
function sortie (evt :MouseEvent) {
    this.zone1.scaleX =1;
    this.zone1.scaleY =1;
}
```

L'objet à surveiller est : **this.zone1**, c'est-à-dire l'occurrence zone1 placé dans l'animation principale (this ou Object(root), suivant que la référence est relative ou absolue).
L'événement est de type **MouseEvent**, événement lié à la souris, et c'est l'événement **MOUSE_OVER** qui est attendu, et la fonction qui sera exécutée dans ce cas sera **survol.**
Il reste évidemment à expliciter la fonction survol().

Pour terminer, une précision : les gestionnaires ne gèrent pas uniquement les événements liés à la souris, il en existe d'autres, comme le montre le fichier Exercice_6.SWF ...

### 3.1 Ne pas utiliser de gestionnaire

Dans certains cas, vous n'aurez pas besoin d'utiliser de gestionnaire. Cela n'est vrai que dans le cas où vous placez un script sur une image clé. L'exécution du script se fait alors une seule fois au moment où la tête de lecture passe, ou repasse, sur l'image clé qui contient le script. On rencontre généralement ce cas lorsque des variables ou des propriétés d'occurrences doivent être initialisées. Vous pouvez bien entendu combiner ces lignes d'instructions qui ne se trouvent pas dans un gestionnaire avec d'autres qui se trouvent dans un gestionnaire. L'instruction qui arrête le déroulement de la tête de lecture dans le scénario se résume à une fonction : stop(). Cette dernière ne doit pas se trouver dans un gestionnaire si l'arrêt doit être fait automatiquement (au passage de la tête de lecture sur une image clé).
L'exemple de script ci-dessus illustre bien la chose :

```
stop();
var leNom:String = new String() ;
Object(root).cadreSelection.X = 10 ;
Object(root).panneauAide.visible = false ;
Object(root).vignette1.addEventListener(MouseEvent.CLICK,
afficheGrande) ;
function afficheGrande (evt :MouseEvent) {
    Object(root).vignette1.scaleX =1 ;
    Object(root).vignette1.scaleY =1 ;
    Object(root).vignette1.alpha =0.7 ;
}
```

Dans cet exemple, une variable, **leNom**, est initialisée et deux propriétés, **X** de l'occurrence cadreSélection et **visible** de panneauAide, sont initialisées,respectivement à **25** et **false**. Un gestionnaire gère quant à lui les lignes d'instructions qu'il faudra exécuter à l'instant où l'utilisateur cliquera sur l'occurrence intitulée vignette1.

## 4 Nomenclature

Vous allez nommer des fichiers, des occurrences, des variables, des fonctions, des images et bien d'autres choses, vous ne pouvez donc pas choisir des noms au hasard pour définir tous les éléments d'une application.

### 4.1 Noms de fichiers

Si vous lisez un nom de fichier comme celui-ci, anim3.swf, il y a de fortes chances pour que vous ne puissiez deviner le contenu de l'animation. En revanche, si vous lisez credits.swf ou accueil.swf, il n'y a pas de doute sur le contenu de ces animations ! Vous l'aurez compris, choisissez toujours des noms représentatifs. Certaines règles supplémentaires s'appliquent également aux noms de fichiers comme celles qui suivent dans la partie ci-dessous.

## 5 Noms d'occurrence, de variable, fonction et image-clé

Respectez les règles suivantes :

- Choisir un nom représentatif et donc ne pas utiliser d'abréviations ou de termes trop génériques (ex. : boutonSortie ou btnSortie et non bts, boutonS, btnSort, etc.)
- Ne pas hésiter à utiliser des noms longs et composés (ex. : curseurDeSon, coinDeLaPage)
- Ne pas utiliser les caractères accentués ou spéciaux
- Ne pas utiliser d'espace (ex. : boutonSortie et non bouton sortie)
- Composez vos noms en utilisant des majuscules (ex. : curseurDeSon et non curseurdeson). Cela facilite la lecture, il s'agit de l'écriture en dosDeChameau.
- Evitez de commencer un nom par une majuscule

A présent, vous pouvez démarrer la programmation d'une animation. Que ce soit une occurrence à contrôler sur la scène, ou une occurrence sur laquelle l'utilisateur va devoir cliquer pour déclencher une action, elle devra toujours être nommée.

## 6 Qu'est-ce que la syntaxe pointée

Pour la rédaction des scripts, vous devez respecter impérativement une syntaxe précise au même titre que vous devez respecter la structure d'une phrase (en français par exemple) pour vous faire comprendre. Sujet verbe complément ! Vous devez remplacer les espaces conventionnels par des points.

Ex. : Placer l'occurrence à 3 pixels du bord gauche de la scène >
Placer.ocurrence.3pixelsde.bordgauchescene

L'exemple donné ci-dessus se traduirait par : *nomoccurrence._x=3*

L'explication d'un tel concept est légèrement abstraite, plutôt difficile à comprendre pour des novices en programmation. Choisissons alors d'analyser plusieurs exemples afin de mieux comprendre. Imaginons que les lignes d'instructions ci-dessous se trouvent sur un bouton qui se trouve lui-même sur la scène.

```
1    this.balle1.X = 12 ;
```

Signification : Une occurrence intitulée balle1 est placée à 12 pixels du bord gauche de la scène.

```
1    this.tableau.balle1.X = 12 ;
```

Signification : Une occurrence intitulée tableau se trouve sur la scène, elle contient une occurrence intitulée balle1 qui est placée à 12 pixels (vers la droite) du centre de l'occurrence Tableau.

```
1    this.balle1.X = this.balle1.X + 12 ;
```

Signification : On repositionne, horizontalement, l'occurrence balle1 à 12 pixels de sa position actuelle

```
1    this.balle1.score = 12 ;
```

Signification : Une variable intitulée score contenue dans l'occurrence intitulée balle1 est affectée de la valeur 12

```
1    this.balle1.gotoAndStop(12) ;
```

Signification : La tête de lecture d'une occurrence intitulée balle1 est déplacée sur l'image 12 et s'y arrête.

La lecture se fait de gauche à droite. Vous trouvez à gauche de la ligne d'instruction, le nom d'un conteneur d'objet (très souvent la scène principale ( this ou Object(root)), puis le nom d'occurrence de l'objet, puis peuvent suivre ensuite un autre nom d'occurrence (dans le cas d'objets imbriqués, voire de plusieurs), tous séparés par un point (.), puis une propriété, un nom de variable, une fonction ou une commande. On parle aussi de "méthode", il s'agit d'une "fonction".
Les éléments propres au langage s'affichent en caractères bleus dans la fenêtre de saisie des actions.
Les éléments qui vous sont propres sont affichés en caractères noirs.
Cela permet de retrouver d'éventuelles erreurs de syntaxe-saisie : length est un mot-clé et sera affiché en bleu tandis que Length sera considéré comme une variable ou un nom d'occurrence, et donc en noir.

# 7 Apprendre à utiliser les raccourcis clavier

Avant tout propos, comprenez bien qu'il est indispensable de connaître les techniques qui vous permettent de gagner du temps en production, c'est pourquoi voici une présentation des raccourcis clavier et autres techniques permettant d'être plus rapide dans Flash.

## 7.1 ESC-X-X

Lorsque vous devez rédiger vos propres scripts, cela sous entend bien évidemment que vous connaissez le langage, mais connaissez vous la saisie "simplifiée" ? Il est possible d'insérer dans vos lignes d'instructions, certaines commandes/fonctions par un simple raccourci clavier.

Ex. : gotoAndStop(5) contient 14 caractères, donc 14 touches du clavier à taper.
Pour gagner du temps, vous pouvez procéder ainsi :

Alors que vous vous apprêtez à taper la lettre "g" de gotoAndStop(5), tapez plutôt sur la touche **Escape** (ou Echappe) de votre clavier*. Tapez ensuite **successivement** sur les 2 touches (lettres) du clavier qui permettent d'insérer la commande automatiquement. (Ex. : g et s).

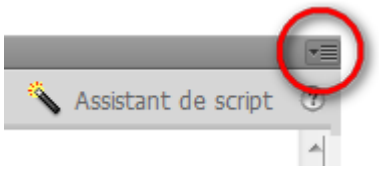
* *Relâchez cette touche, vous ne devez surtout pas la garder enfoncée comme vous le feriez dans le cas d'un raccourci clavier classique (Ex. : CTRL-S ou Commande-S pour sauvegarder un document).*

Certaines commandes et fonctions peuvent donc être saisies très rapidement grâce à ce type de raccourci. Voici quelques exemples supplémentaires :

ESC - G - S : gotoAndStop()
ESC - G - U : getURL()
ESC - S - T : stop()

ESC - F - N : function() { }
ESC - I - F : if() { }
ESC - F - R : for() { }
Les raccourcis clavier sont indiqués, lorsqu'ils existent, à la suite de l'instruction ActionScript.

Pour que les raccourcis clavier fonctionnent, il faut d'abord les avoir activés dans le menu de la fenêtre Actions : clic sur le bouton Menu de la fenêtre puis Cocher la commande Touches de raccourcis Echap .




Attention : il semblerait que certains raccourcis clavier ne soient plus disponibles dans la version CS5. Mais une extension proposé sur le site yazo.net permet de remédier à ce désagrément. : http://yazo.net/racine/download/RaccourcisYazo.mxp .

Pour une plus grande optimisation de votre travail (production) vous pouvez aussi utiliser les touches ci-dessous pour la sélection de vos outils, la manipulation de votre timeline et celle de vos occurrences sur la scène. Bien sûr, il ne s'agit pas d'une liste exhaustive mais au contraire des raccourcis "minimums" à connaître.

*Raccourcis Mac et PC :* **CTRL** *pour PC et* ***Commande*** *pour Mac. Ex. :* **CTRL-Entrée** *(PC) ou* **Commande-Entrée** *(Mac)*

**CTRL-Entrée** : Lecture de l'animation dans une nouvelle fenêtre.
OUTILS (Tapez uniquement les lettres ci-dessous sans CTRL (PC) ou Commande (Mac))
**V** : Flèche noire (sélecteur)
**T** : Outil texte
**O** : Ovale
**R** : Rectangle
TIMELINE
**F6** : Créer une image clé
**F7** : Créer une image clé vide
**F5** : Créer/Insérer des images
SCENE
**F8** : Créer un symbole (à partir d'une sélection)
**Shift-F8** : Créer un symbole vide (A composer juste après l'exécution du raccourci)
**CTRL-Shift-V** : Coller à la même place que sa copie (Commande-C)
MISE EN FORME DU TEXTE
**CTRL-Shift-B** : Mettre une sélection de texte en gras (**P** > Normal- **I** > Italique)
**CTRL-Shift-C** : Centrer un texte dans sa zone de texte (**J** > Justifier - **L** > Gauche)

## 7.2 MENU CONTEXTUEL DANS LA FENETRE D'ACTIONS

Il existe une autre technique qui permet d'accéder à une aide supplémentaire relative aux méthodes et propriétés que vous pouvez associer à un nom d'instance ou _root. Cette technique est très contraignante car cela vous oblige à nommer vos noms d'instances avec un suffixe supplémentaire. Le principe reste simple. La liste ci-dessous répertorie toutes les classes et leurs suffixes disponibles dans Flash MX et MX 2004.

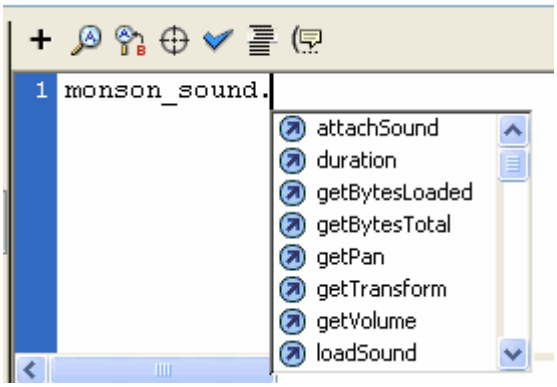
Liste des suffixes à placer derrière le nom de l'instance :

| **Objets** | **Suffixes** | **Objets** | **Suffixes** |
|---|---|---|---|
| Array | _array | **MovieClip** | **_mc** |
| Button | _btn | MovieClipLoader | _mcl |
| Camera | _cam | PrintJob | _pj |
| Color | _color | NetConnection | _nc |
| ContextMenu | _cm | NetStream | _ns |
| ContextMenuItem | _cmi | **SharedObject** | **_so** |
| Date | _date | Sound | _sound |
| Error | _err | **String** | **_str** |

| | | | |
|---|---|---|---|
| LoadVars | _lv | Video | _video |
| LocalConnection | _lc | XML | _xml |
| **TextField** | **_txt** | XMLNode | _xmlnode |
| **TextFormat** | **_fmt** | XMLSocket | _xmlsocket |

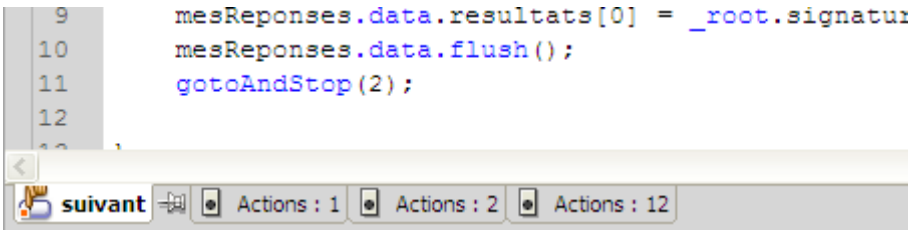
*Liste extraite de l'aide en ligne de Flash MX 2004*

Il est clair que vous devez impérativement donner des noms représentatifs pour vos noms d'instance et de variable dans vos développements, surtout si vous devez travailler en équipe et/ou travailler sur des projets contenants de nombreuses lignes de code. Alors un peu plus ou un peu moins, on est pas à quelques lettres... Au lieu de nommer votre instance sonFond = new Sound(), vous pouvez la nommer sonFond_sound = new Sound(). Vous pourrez alors obtenir le menu ci-dessous. A vous de voir si vous voulez vous contraindre à ajouter des suffixes à vos noms d'instances pour obtenir ce menu dès que vous saisissez le point situé derrière le nom d'instance. Selon vos habitudes de développement, selon la nature des développements, il est intéressant de se contraindre à une telle nomenclature. Les suffixes des classes ci-dessus figurant en gras valent la peine d'être retenus et utilisés.




## 8 Les onglets de la fenêtre Actions

Lorsque vous réaliserez des projets contenants de nombreuses lignes de code avec de nombreux scripts, vous rencontrerez alors l'obligation de vous déplacer de scripts en scripts (pas obligatoirement sur la même timeline). Vous allez donc passer une bonne partie de votre temps à rechercher et sélectionner les images clés (parfois cachées dans la timeline des occurrences) qui contiennent les scripts à modifier, puis vous rechercherez enfin la bonne ligne d'instruction en utilisant l'ascenseur de la fenêtre Actions. Pour vous éviter toutes ces manipulations inutiles, vous pouvez utiliser la punaise posée à côté de l'onglet : une fois l'onglet punaisé, il suffit de cliquer dessus pour accéder au script correspondant.




Le panneau de l'explorateur d'animation (ALT+F3) est aussi un excellent outil pour retrouver un script « perdu » …