

# Javaramach

*Javaramach* affiche des images *et des textes purs* sous forme de diaporama. Il existe en 3 variantes qui diffèrent dans l'organisation temporelle du déroulement des objets et les transitions possibles.

<u>Nom</u>	<u>Caractéristique</u>	<u>Version traitée</u>
<b>JavaramaS</b>	simple (transitions en fondu)	<b>2.02</b>
<b>JavaramaM</b>	mutiple (effets multiples de transitions)	<b>2.02</b>
<b>JavaramaX</b>	étendu (transitions en fondu mais gestion étendue des départs de transition)	<b>2.02</b>

## Caractéristiques communes aux 3 *Javaramach*

- ▶ Pour chaque objet, indépendamment de tous les autres, précédents et suivants, le créateur gère totalement les durées :
  - de la transition d'apparition, "transition entrante" ;
  - d'affichage à 100% ;
  - de la transition de disparition, "transition sortante".
- ▶ Tous les temps sont en dixièmes de seconde.
- ▶ Le diaporama global est contenu dans une boîte (ou conteneur) principale qui le positionne.  
Dans cette boîte, sont déclarées les boîtes contenant les objets à animer par *Javaramach*. Cet objet peut être une image, un texte, une image ET un texte... Si l'objet est unique (image ou texte), il peut être déclaré directement dans le conteneur du diaporama, sa déclaration formant "boîte".
- ▶ Des objets à animer peuvent se situer n'importe où dans la page, hors de la boîte principale. Ils sont pris en charge comme les autres par *Javaramach*.
- ▶ Dans la page HTML, la boîte du diaporama est positionnée par le créateur de façon absolue ou dans le flux. Son identifiant "**boitJVRM**" lui donne le style attendu par *Javaramach*, avec si nécessaire, des compléments de style. À l'intérieur, les objets du diaporama sont créés dans un ordre qui contribue au rendu final. Si les objets A, B et C sont déclarés dans cet ordre, C recouvrira B et A quand C sera visible, B recouvrira A quand B sera visible.
- ▶ Conformément aux normes du W3C, les fichiers HTML et CSS contiennent la déclaration des objets (contenu et architecture) et les fichiers JavaScript contiennent les définitions du comportement.
- ▶ Écrits en JavaScript, les 3 moteurs ne contiennent aucune personnalisation et ainsi peuvent être communs à un site entier. Ils sont compatibles avec tous les navigateurs qui respectent les standards du W3C en HTML, CSS et JavaScript.

### Notas :

- *Internet Explorer* demande des ajustements (CSS) à cause de son traitement incorrect des boîtes ;
- *Javaramach* ne fonctionne pas avec *Konqueror (GNU-Linux + KDE)* qui ne gère pas l'opacité.
- ▶ Fonctionnement avec un pupitre de commande : la PAUSE  
Quand on passe en pause, si des transitions sont en cours, la transition entrante s'interrompt et la sortante s'achève. Au retour en marche ("dé-pause"), l'entrante reprend au point d'interruption.

## Recommandation générale

Pas d'espace, ni apostrophe, ni caractères accentués, ni signes diacritiques dans les noms de fichiers et d'images, dans les identifiants ("id") de objets HTML. Que des lettres non accentuées, des chiffres, tirets et soulignés.

# JavaramaS

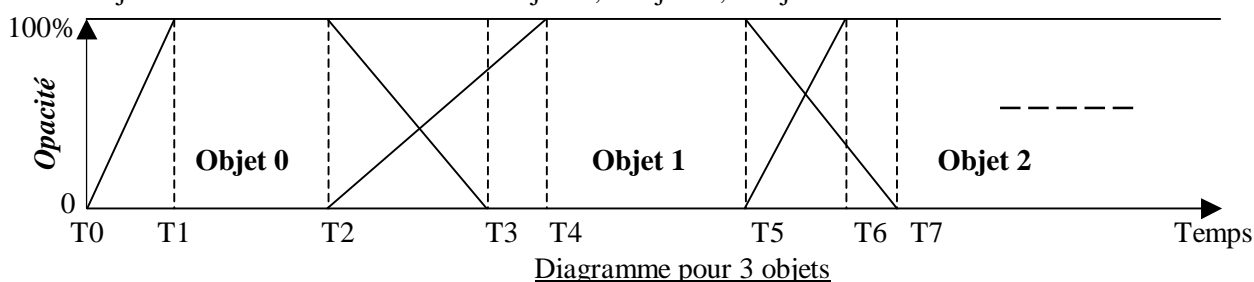
## Fonctionnement de JavaramaS

### Deux règles

- \* Le seul effet de transition est un fondu réalisé par gestion de l'**opacité**. C'est-à-dire qu'avant de commencer d'afficher l'objet (une image par exemple), à l'état d'opacité nulle, l'objet est invisible et ce qui se trouve "sous" (déclaré avant lui ou avec un "z-index" inférieur) cet objet est visible. Cette notion est primordiale pour définir l'ensemble des paramètres qui détermineront le déroulement du diaporama.
- \* L'apparition de l'objet "N+1" démarre quand commence la disparition de l'objet "N".

### Diagramme du déroulement d'un diaporama mû par JavaramaS

Les objets sont déclarés dans l'ordre : "Objet 0", "Objet 1", "Objet 2"...



avant T0 -----> on voit le fond (couleur, image, texte...)

à T0 -----> Objet 0 : début de transition "entrante" vers l'apparition

à T1 -----> Objet 0 : fin de transition "entrante" vers l'apparition, objet totalement visible

de T1 à T2 -----> Objet 0 : affichage à 100%

à T2 -----> Objet 0 : début de transition "sortante" vers la disparition

> Objet 1 : **simultanément** début de transition "entrante" vers l'apparition

à T3 -----> Objet 0 : fin de transition "sortante" vers la disparition

à T4 -----> Objet 1 : fin de transition "entrante" vers l'apparition, objet totalement visible

Dans l'exemple, l'**Objet 0** disparaît plus vite que l'**Objet 1** apparaît, donc avec une opacité totale plus faible.

de T4 à T5 -----> Objet 1 : affichage à 100%

à T5 -----> Objet 1 : début de transition "sortante" vers la disparition

> Objet 2 : **simultanément** début de transition "entrante" vers l'apparition

à T6 -----> Objet 2 : fin de transition "entrante" vers l'apparition

à T7 -----> Objet 1 : fin de transition "sortante" vers la disparition

Dans l'exemple, l'**Objet 2** disparaît plus lentement que l'**Objet 2** apparaît, donc avec une opacité totale plus grande.

de T6 à ... -----> OBJET 2 : affichage à 100% ...

## Composants de JavaramaS

Le moteur JavaramaS : JavaramaS.js

Le fichier de style pour JavaramaS : JVRMstyle.css

Il contient 2 styles à utiliser dans le fichier HTML.

*#boitJVRM* : "boîte" (ou "conteneur") du diaporama entier

*.objJVRM* : objets du diaporama ou "boîte" (ou "conteneur") de l'objet du diaporama

Le fichier de personnalisation pour JavaramaS : JVRMvar.js

Il contient TOUTES les personnalisations du diaporama créé.

## Création avec JavaramaS

Les parties en **orange** sont à personnaliser.

### Le fichier HTML

Il décrit la page où UN SEUL diaporama **Javaramach** apparaît. Il appelle les 3 composants de **Javaramach**.

Dans l'en-tête (entre les balises <head> et </head>) :

- Appel de la feuille de style :

```
<link rel="stylesheet" href="JVRMstyle.css"/>
```

- Dans l'ordre

- > appel du fichier de personnalisation :

```
<script language="JavaScript" src="JVRMvar.js"></script>
```

- > appel du moteur **JavaramaS** :

```
<script language="JavaScript" src="JavaramaS.js"></script>
```

Dans le corps du fichier HTML :

- Initialiser le diaporama avec dans la balise **body** (**modification 2.00**)

```
<body onload="init()" XXXX>
```

- Déclaration de la **boîte contenant le diaporama**

```
<div id="boitJVRM">
```

- Déclaration des **boîtes contenant les objets** composant le diaporama. Autant de boîtes (**div**) que d'objets du diaporama. Chaque boîte est identifiée "**Jvrm**" avec un numéro de 0 à N : "**Jvrm0**", "**Jvrm1**", "**Jvrm2**"...

```
<div id="Jvrm0" class="objJVRM"><br/>TEXTE</div>
```

Si l'objet est une image ou un texte seul, il est possible de le déclarer directement comme suit :

```

```

Si les objets à animer par **Javaramach** sont plus petits que la boîte du diaporama, les positionner précisément dans celle-ci à l'aide de paramètres de style ajoutés à la boîte des objets. Exemple :

```
<div id="Jvrm1" class="objJVRM" style="left:17px; top:29px">
```

Dans ce cas, si du texte accompagne les images repositionnées, son positionnement sera incorrect par rapport aux autres et devra être redéfini.

- **Pupitre de commande** OPTIONNEL

**Indispensable** si le diaporama n'est pas en démarrage automatique. Il peut se limiter au bouton de lancement, ou aux boutons de lancement et de pause, ou aux boutons de lancement et d'arrêt.

Il peut aussi être présent en démarrage automatique. Il servira alors à passer en pause, redémarrer et arrêter le déroulement.

Voir les déclarations complètes dans le fichier HTML d'exemple : **tstJVRM.html**

Apparence résultante (les boutons sont fournis pour l'exemple et toutes autres images peuvent être utilisées par le créateur du diaporama)



: lance le diaporama. S'affiche pendant le déroulement.



: met en pause le déroulement. En pause, clignote : / pour attirer l'attention. Pour relancer le déroulement, appuyer de nouveau sur la pause.



: arrête le déroulement automatique. Pour reprendre le déroulement, appuyer sur le bouton de lancement.

### Personnalisation

Les paramètres personnalisant le déroulement du diaporama se trouvent dans le fichier **JVRMvar.js** où des paramètres sont obligatoires, ici précédés de \* et d'autres subsidiaires, ici précédés de °

#### \* Boîte du diaporama

Déclarer sa largeur et sa hauteur en pixels

```
var JVRMlarg=360;
```

```
var JVRMhaut=280;
```

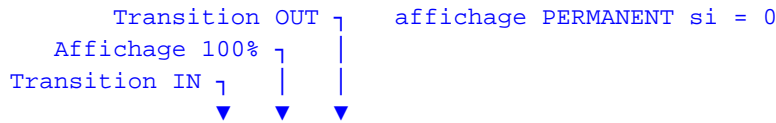
#### \* Boîtes contenant les objets du diaporama

Chaque boîte possède 3 paramètres temporels.

- **Durée de transition entrante**.....Peut être nulle
- **Durée d'affichage à 100%** .....Peut être nulle

- **Durée de transition sortante** ..... Une valeur nulle rend l'affichage permanent ..... Pour une transition immédiate, déclarer "1" par exemple
- SEULE NOTION FIGÉE : la disparition de l'objet "N" et l'apparition de l'objet "N+1" commencent au même instant.

Les paramètres de chaque objet sont déclarés sous forme de tableau.



```
var obj00=new Array( 20, 25, 10);
```

Le fichier **JVRMvar.js** contient les commentaires complétant ces explications. Le nom de chaque objet n'a aucune importance. Pour la lisibilité, donner des noms en suivant "obj00", "obj01", "obj02"...

Le nom INVARIABLE est celui du tableau (*obj*) où tous les tableaux sont déclarés.

```
var obj=new Array(obj00,obj01,obj02,obj03,obj04);
```

#### ° Nom des boîtes contenant les objets du diaporama

Ce nom est utilisé dans la déclaration des boîtes dans le fichier HTML :

```
<div id="Jvrm0" class="objJVRM"></div>
```

Déclaré comme "Jvrm" SANS NUMÉRO dans l'exemple, ce nom est libre.

```
var nomObj="Jvrm";
```

#### ° Pas des transitions

Temps exprimé en millisecondes. Déterminant le nombre d'images par seconde affichées durant les transitions, ce paramètre est important pour obtenir un fondu fluide ou saccadé à l'inverse. Toute valeur inférieure à **40** sera ramenée à **40**.

Par exemple, déclaré à **55**, il produit ~18 images/seconde.

```
var pasTr=55;
```

#### ° Démarrage automatique

À l'ouverture de la page, le diaporama peut démarrer automatiquement (**OUI**) ou non (**NON**). Si le démarrage n'est pas automatique, il est **INDISPENSABLE** de créer un pupitre de commande dans le fichier HTML.

```
var demarAuto="OUI";
```

#### ° Bouclage

Ce paramètre indique si arrivé en fin du diaporama, le déroulement reprend au début avec les objets réinitialisés dans leur état au premier départ. Valeur acceptées : **OUI** ou **NON**. Toute autre valeur que **OUI** est prise comme **NON**.

```
var bouclage="OUI";
```

#### ° Fonction spéciale de fin (*Ce paramétrage AVANCÉ demande d'écrire des instructions en JavaScript*)

Si le diaporama n'est pas en bouclage, cette fonction sera déclenchée **APRÈS LA DISPARITION** du dernier objet.

On peut utiliser cet automatisme pour vider la page du diaporama, la changer pour en appeler une autre...

Les parties en **VERT** sont des commentaires.

```
// fonction spéciale de fin activée à l'extinction fin automatique du diaporama (sans
bouclage)
function foncSpeFin() { // créer ci-dessous les actions à réaliser
}
```

# JavaramaM

## *Fonctionnement du JavaramaM*

### Trois règles

- \* Les effets de transition sont multiples.
- \* La transition entrante (apparition) de l'objet "N+1" commence au départ la sortante (disparition) de l'objet "N".
- \* Au 1<sup>er</sup> départ ainsi qu'à chaque relance sur bouclage, tous les objets sont réinitialisés en position (x=0 et y=0) et opacité.

### Diagramme du déroulement d'un diaporama mû par JavaramaM

Le diagramme est identique à celui de *JavaramaS*.

## *Composants de JavaramaM*

Le moteur pour JavaramaM : JavaramaM.js

Le fichier de style pour JavaramaM : JVRMstyle.css

Il contient 2 styles à utiliser dans le fichier HTML.

*#boitJVRM* : "boîte" (ou "conteneur") du diaporama entier

*.objJVRM* : objets du diaporama ou "boîte" (ou "conteneur") de l'objet du diaporama

Le fichier de personnalisation pour JavaramaM : JVRMvar.js

Il contient TOUTES les personnalisations du diaporama créé.

## *Création avec JavaramaM*

Les parties en **orange** sont à personnaliser.

### Le fichier HTML

Il décrit la page où UN SEUL diaporama *Javaramach* apparaît. Il appelle les 3 composants de *Javaramach*.

Dans l'en-tête (entre les balises <head> et </head>) :

- Appel de la feuille de style :

```
<link rel="stylesheet" href="JVRMstyle.css"/>
```

- Dans l'ordre

- > appel du fichier de personnalisation :

```
<script language="JavaScript" src="JVRMvar.js"></script>
```

- > appel du moteur *JavaramaM* :

```
<script language="JavaScript" src="JavaramaM.js"></script>
```

Dans le corps du fichier HTML :

- Initialiser le diaporama avec dans la balise **body** (**modification 2.00**)

```
<body onload="init()" XXXX>
```

- Déclaration de la **boîte contenant le diaporama**

```
<div id="boitJVRM">
```

- Déclaration des **boîtes contenant les objets** composant le diaporama. Autant de boîtes (**div**) que d'objets du diaporama. Chaque boîte est identifiée "*Jvrm*" avec un numéro de 0 à N : "*Jvrm0*", "*Jvrm1*", "*Jvrm2*"...

```
<div id="Jvrm0" class="objJVRM"><br/>TEXTE</div>
```

- Pour définir un objet **HORS** de la boîte du diaporama, créer d'abord une boîte positionnée à l'endroit souhaité, y insérer la boîte de l'objet piloté par *JavaramaM* en l'identifiant ("**id**") dans la suite "*JvrmN*" puis le déclarer dans **JVRMvar.js** pour définir son comportement. Exemple :

```
<div style="position:absolute; left:96px; top:62px;"><!--boîte protectrice de l'objet-
->
<div id="Jvrm0" class="objJVRM"><!--boîte de l'objet piloté par JavaramaM-->
  <div clas="xxxx">TEXTE</div><!--contenu de l'objet-->
</div>
</div>
```

- Déclaration du **pupitre de commande** OPTIONNEL  
Exactement comme avec *JavaramaS*.

### Personnalisation

Les paramètres personnalisant le déroulement du diaporama se trouvent dans le fichier **JVRMvar.js** où des paramètres sont obligatoires, ici précédés de \* et d'autres subsidiaires, ici précédés de °

#### \* **Boîte du diaporama**

Déclarer sa largeur et sa hauteur en pixels

```
var JVRMlarg=360;
var JVRMhaut=280;
```

#### \* **Boîtes contenant les objets du diaporama**

Chaque boîte possède 5 paramètres.

- **Transition entrante** ..... Valeur de 0 à 13 ; forcée à 0 (fondu) pour toute autre valeur
- **Durée de la transition entrante** ..... Peut être nulle
- **Durée d'affichage à 100%** ..... Peut être nulle
- **Transition sortante** ..... Valeur de 0 à 13 ; forcée à 0 (fondu) pour toute autre valeur
- **Durée de la transition sortante** ..... Une valeur nulle rend l'affichage permanent  
..... Pour une transition immédiate, déclarer "1" par exemple

Transitions possibles pour des objets IMAGES, des objets TEXTES ou des objets COMPLEXES (images ET textes)

0. **Fondu** par variation de l'opacité
1. **Glisse du haut vers le bas**  
Apparition, l'objet entier glisse du haut vers le bas.  
Disparition, l'objet entier glisse du bas vers le haut.
2. **Glisse du bas vers le haut**  
Apparition, l'objet entier glisse du bas vers le haut.  
Disparition, l'objet entier glisse du haut vers le bas.
3. **Glisse de la gauche vers la droite**  
Apparition, l'objet entier glisse de la gauche vers la droite.  
Disparition, l'objet entier glisse de la droite vers la gauche.
4. **Glisse de la droite vers la gauche**  
Apparition, l'objet entier glisse de la droite vers la gauche.  
Disparition, l'objet entier glisse de la gauche vers la droite.

Transitions possibles pour des objets IMAGES PURES

5. **Déplie du haut vers le bas**  
Apparition, l'objet se "déplie" du haut vers le bas.  
Disparition, l'objet se "replie" du bas vers le haut.
6. **Déplie du bas vers le haut**  
Apparition, l'objet se "déplie" du bas vers le haut.  
Disparition, l'objet se "replie" du haut vers le bas.
7. **Déplie de la gauche vers la droite**  
Apparition, l'objet se "déplie" de la gauche vers la droite.  
Disparition, l'objet se "replie" de la droite vers la gauche.
8. **Déplie de la droite vers la gauche**  
Apparition, l'objet se "déplie" de la droite vers la gauche.  
Disparition, l'objet se "replie" de la gauche vers la droite.
9. **Déplie du coin haut-gauche vers le coin bas-droite**  
Apparition, l'objet se "déplie" en diagonale du coin haut-gauche vers le coin bas-droite.  
Disparition, l'objet se "replie" en diagonale du coin bas-droite vers le coin haut-gauche.

### 10. Déplie du coin haut-droite vers le coin bas-gauche

Apparition, l'objet se "déplie" en diagonale du coin haut-droite vers le coin bas-gauche.

Disparition, l'objet se "replie" en diagonale du coin bas-gauche vers le coin haut-droite.

### 11. Déplie du coin bas-gauche vers le coin haut-droite

Apparition, l'objet se "déplie" en diagonale du coin bas-gauche vers le coin haut-droite.

Disparition, l'objet se "replie" en diagonale du coin haut-droite vers le coin bas-gauche.

### 12. Déplie du coin bas-droite vers le coin haut-gauche

Apparition, l'objet se "déplie" en diagonale du coin bas-droite vers le coin haut-gauche.

Disparition, l'objet se "replie" en diagonale du coin haut-gauche vers le coin bas-droite.

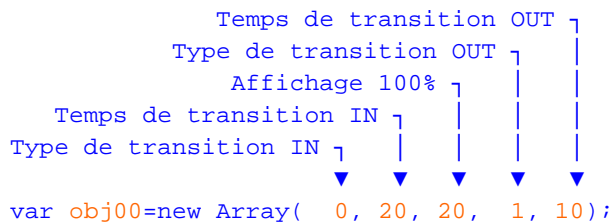
### 13. Déplie du centre vers les 4 coins

Apparition, l'objet se "déplie" du centre vers les 4 coins.

Disparition, l'objet se "replie" des 4 coins vers le centre.

**SEULE NOTION FIGÉE** : la transition sortante de l'objet "N" et la transition entrante de l'objet "N+1" commencent au même instant.

Les paramètres de chaque objet sont déclarés sous forme de tableau.



Le fichier **JVRMvar.js** contient les commentaires complétant ces explications. Le nom de chaque objet n'a aucune importance. Pour la lisibilité, donner des noms en suivant "obj00", "obj01", "obj02"...

Le nom INVARIABLE est celui du tableau (*obj*) où tous les tableaux sont déclarés.

```
var obj=new Array(obj00,obj01,obj02,obj03,obj04);
```

#### ◦ Boîtes contenant les objets du diaporama

Ce nom est utilisé dans la déclaration des boîtes dans le fichier HTML :

```
<div id="Jvrm0" class="objJVRM"></div>
```

Déclaré comme "Jvrm" dans l'exemple, ce nom est libre.

```
var nomObj="Jvrm";
```

#### ◦ Pas des transitions

Temps exprimé en millisecondes. Déterminant le nombre d'images par seconde affichées durant les transitions, ce paramètre est important pour obtenir un fondu fluide ou saccadé à l'inverse. Toute valeur inférieure à **40** sera ramenée à **40**.

Par exemple, déclaré à **55**, il produit ~18 images/seconde.

```
var pasTr=55;
```

#### \* Démarrage automatique

À l'ouverture de la page, le diaporama peut démarrer automatiquement (**OUI**) ou non (**NON**). Si le démarrage n'est pas automatique, il est **INDISPENSABLE** de créer le pupitre de commande dans le fichier HTML.

```
var demarAuto="OUI";
```

#### \* Bouclage

Ce paramètre indique si arrivé en fin du diaporama, le déroulement repart au début avec les objets réinitialisés dans leur état au premier départ. Valeur acceptées : **OUI** ou **NON**. Toute autre valeur que **OUI** est prise comme **NON**.

```
var bouclage="OUI";
```

#### ◦ Fonction spéciale de fin (Ce paramétrage AVANCÉ demande d'écrire des instructions en JavaScript)

Si le diaporama n'est pas en bouclage, cette fonction sera déclenchée **APRÈS LA DISPARITION** du dernier objet.

On peut utiliser cet automatisme pour vider la page du diaporama, la changer pour en appeler une autre...

Les parties en **VERT** sont des commentaires.

```
// fonction spéciale de fin activée à la fin du diaporama (sans bouclage)
function foncSpeFin() { // créer ci-dessous les actions à réaliser
}
```

# JavaramaX

## Fonctionnement du JavaramaX

### Deux règles

- \* Le seul effet de transition est le **fondue** réalisé par gestion de l'**opacité**. C'est-à-dire qu'avant de commencer d'afficher l'objet (une image par exemple), à l'état d'opacité nulle, l'objet est invisible et ce qui se trouve "sous" cet objet est visible. Cette notion est primordiale pour définir l'ensemble des paramètres qui feront le diaporama.
- \* Le déroulement de chaque objet (apparition, affichage à 100% et disparition) est indépendant du précédent et du suivant.

### Diagramme du déroulement d'un diaporama mû par *JavaramaX*

Le diagramme est basé sur le temps ZÉRO de l'ensemble du déroulement. Le départ de chaque objet est donné par rapport à ce T0, totalement libre par rapport au début de la transition sortante du précédent. Chaque durée de transition est indépendante de toutes les autres. Ces conditions donnent une indépendance intégrale des différents objets entre eux.

Un objet peut apparaître hors de la boîte du diaporama et accompagner le diaporama d'une légende par exemple.

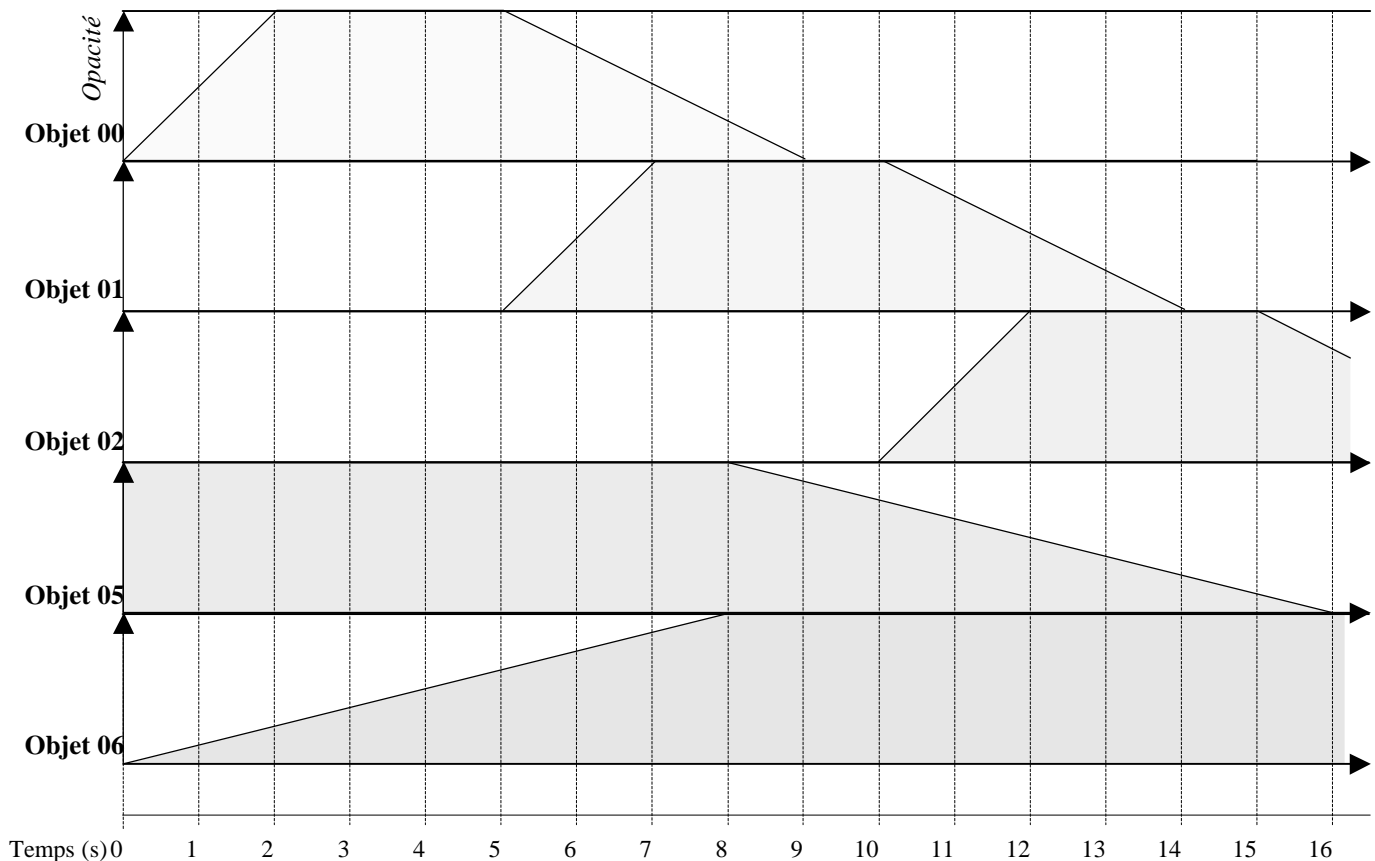


Diagramme réduit à 5 des 7 objets de l'exemple fourni

## Composants de JavaramaX

Le moteur *JavaramaX* : JavaramaX.js

Le fichier de style pour *JavaramaX* : JVRMstyle.css

Il contient 2 styles à utiliser dans le fichier HTML.

*#boitJVRM* : "boîte" (ou "conteneur") du diaporama entier

*.objJVRM* : objets du diaporama ou "boîte" (ou "conteneur") de l'objet du diaporama



Le fichier de personnalisation pour **JavaramaX** : JVRMvar.js  
Il contient TOUTES les personnalisations du diaporama créé.

## *Création avec JavaramaX*

Les parties en **orange** sont à personnaliser.

### Le fichier HTML

Il décrit la page où UN SEUL diaporama **Javaramach** apparaît. Il appelle les 3 composants de **Javaramach**.

Dans l'en-tête (entre les balises <head> et </head>) :

- Appel de la feuille de style :  
`<link rel="stylesheet" href="JVRMstyle.css"/>`
- Dans l'ordre
  - > appel du fichier de personnalisation :  
`<script language="JavaScript" src="JVRMvar.js"></script>`
  - > appel du moteur **JavaramaX** :  
`<script language="JavaScript" src="JavaramaX.js"></script>`

Dans le corps du fichier HTML :

- Initialiser le diaporama avec dans la balise **body** (**modifications 2.00 / 2.01**)  
`<body onload="initJVRM()" XXXX>`
- Déclaration de la **boîte contenant le diaporama**. C'est une boîte "**div**" libre.
- Déclaration des **boîtes contenant les objets** composant le diaporama. Autant de boîtes (**div**) que d'objets du diaporama. Chaque boîte est identifiée "**Jvrm**" avec un numéro de 0 à N : "**Jvrm0**", "**Jvrm1**", "**Jvrm2**"...  
`<div id="Jvrm0" class="objJVRM"><br/>TEXTE</div>`
- Un objet peut être créé **HORS** de la boîte du diaporama. Il suffit de l'identifier ("**id**") dans la suite : "**JvrmN**" puis de le déclarer dans **JVRMvar.js** pour définir son comportement.

### Personnalisation

Les paramètres personnalisant le déroulement du diaporama se trouvent dans le fichier **JVRMvar.js** où des paramètres sont obligatoires, ici précédés de \* et d'autres subsidiaires, ici précédés de °

#### \* **Boîtes contenant les objets du diaporama**

Chaque boîte possède 4 paramètres temporels.

- **Instant de départ de l'objet** par rapport au T0 de l'ensemble du diaporama..... Peut être nul
- **Durée de transition entrante**..... Peut être nulle
- **Durée d'affichage à 100%** ..... Une valeur nulle rend l'affichage permanent  
..... Pour une durée minimale d'affichage, déclarer "1" par exemple
- **Durée de transition sortante**..... Peut être nulle

Les paramètres de chaque objet sont déclarés sous forme de tableau.

```

      Durée transition OUT  |
      Durée affichage 100% | INFINIE SI = 0
      Durée transition IN  |
Top départ de l'objet     |
      ▼                   |
var obj00=new Array( 50, 20, 30, 40);
    
```

Le fichier **JVRMvar.js** contient les commentaires complétant ces explications. Le nom de chaque objet n'a aucune importance. Pour la lisibilité, donner des noms en suivant "**obj00**", "**obj01**", "**obj02**"...

Le nom INVARIABLE est celui du tableau (**obj**) où tous les tableaux sont déclarés.

```
var obj=new Array(obj00,obj01,obj02,obj03,obj04);
```

#### \* **Bouclage** : la commande indique à quel instant le diaporama sera relancé au premier objet, instant calculé à partir du début T0 du diaporama. Déclarer "0" pour ne pas reboucler.

NOTA : le BOUCLAGE est réalisé SANS RÉINITIALISATION des objets.

```
var boucle=260;
```

- \* **Nombre d'objets à ne pas relancer** : après le 1<sup>er</sup> déroulement qui traite tous les objets, on peut ne pas relancer un certain nombre d'objets. Dans ce cas, le nombre d'objets à relancer sera réduit du nombre déclaré. Dans l'exemple fourni, on relancera tous les objets sauf les 2 derniers.

```
var NePasReLanC=2;
```

- ° **Nom des boîtes contenant les objets du diaporama**

Ce nom est utilisé dans la déclaration des boîtes dans le fichier HTML :

```
<div id="Jvrm0" class="objJVRM"></div>
```

Déclaré comme "*Jvrm*" dans l'exemple, ce nom est libre.

```
var nomObj="Jvrm";
```

- \* **Pas des transitions**

Temps exprimé en millisecondes. Déterminant le nombre d'images par seconde affichées durant les transitions, ce paramètre est important pour obtenir un fondu fluide ou saccadé à l'inverse. Toute valeur inférieure à **40** sera ramenée à **40**.

Par exemple, déclaré à **55**, il produit ~18 images/seconde pendant les transitions.

```
var pasTr=55;
```

- ° **Fonction spéciale de fin de 1<sup>er</sup> tour (modification 2.02)**

*Ce paramétrage AVANCÉ demande d'écrire des instructions en JavaScript.*

Même si le diaporama est en bouclage, cette fonction sera déclenchée **APRÈS LA FIN DU PREMIER TOUR** du diaporama entier.






On peut utiliser cet automatisme pour corriger le défaut des "**Internet Explorer**" affichant de façon **BAVEUSE** les textes soumis au paramètre d'opacité variable, ou toute autre action souhaitée indépendante d'un paramètre d'opacité.

Les parties en **VERT** sont des commentaires.




```
// fonction spéciale de FIN DE PREMIER TOUR du diaporama
function foncSpeTour1() {           // créer ci-dessous les actions à réaliser
}
```

## *Quelques couleurs courantes*




### Couleurs pures

	= FFFFFFFF (Blanc)
	= FF0000 (Rouge)
	= 00FF00 (Vert)
	= 0000FF (Bleu)
	= 000000 (Noir)




### Couleurs plus sombres




	= 800000
	= 008000
	= 000080

### Couleurs mélangées

	= 00FFFF (Cyan)
	= FFFF00 (Jaune)
	= FF00FF (Magenta)

### Couleurs plus sombres

	= 008080
	= 808000
	= 800080

	= 808080 (Gris)
	= C0C0C0 (Gris)
	= E0E0E0 (Gris)