

Principe d'intégration de figures dans une page HTML.

Voici ci-dessous de code typique à implémenter dans une page html pour y incorporer une figure dynamique MathGraph32 gérée par le player MathGraph32 (typiquement dans le body de la page).

```
<div id = "mtg32"></div>
<script type="text/javascript" src="https://dev.mathgraph32.org/js/mtgLoad/mtgLoad.min.js"></script>
<script type="application/javascript">
    (function autostart() {
        var svgOptions = {
            idSvg: "mtg32svg", width : "1087", height : "654"
        }

        var mtgOptions = {
            fig:
            "TWF0aEdyYXB0SmF2YTEuMAAAABM+TMzNAAJmcv###wEA#wEAAAAAAAAAAAAAPxAAACHwAAAQE
            AAAAAAAAAQAAAAH#####AAAAAQAKQ0NhbGNDb25zdAD#####AAJwaQAWMy4xNDE1OTI2NTM1
            ODK3OTMyMzg0Nv###8AAAABAApDQ29uc3RhbnRIQAKh+1RELrj#####8=",
            isEditable: false,
            callBackAfterReady: function callBackAfterReady (app) {
                // C'est ici qu'il faut mettre toutes vos initialisations pour la figure dynamique.
            }
        }

        mtgLoad("mtg32", svgOptions, mtgOptions, function (error, mtgAppPlayer) {
            if (error) return console.error(error)
            console.log("mtg32 player loaded", mtgAppPlayer)
        })
    })()
</script>
```

Le div devant contenir la figure a ici pour id *mtg32*.

C'est à l'intérieur de ce div que MathGraph32 créera un svg qui aura ici pour id *mtg32svg* et pour dimensions 1087 et 654, ces valeurs étant fournies au player via la variable *svgOptions*.

La variable objet *mtgOptions* fournit quant à elle dans *fig* le code Base 64 de la figure à afficher.

isEditable doit être à *false* pour le player (sinon c'est l'éditeur de figure MathGraph32 qui sera lancé).

Et *callBackAfterReady* fournit une fonction qui sera appelée une fois que le player sera prêt. Une instance de ce player sera fournie lors de l'appel dans le paramètre *app*.

C'est dans cette fonction de *callBack* qu'il vous faudra mettre toutes les initialisations de votre figure.

Si vous voulez ajouter une autre figure à afficher dans la même page, créez un nouveau div destiné à la contenir, une nouvelle variable *svgOptions* et une nouvelle variable *mtgOptions* et procédez à un nouvel appel de *mtgLoad* avec ces paramètres.

Attention : N'utilisez pas les mêmes variables *svgOptions* et *mtgOptions* pour la nouvelle figure.

Autres paramètres qu'on peut passer via le paramètre *mtgOptions* lors de l'appel à *mtgLoad* :


displayOnLoad : Si *true*, la figure est affichée dès que possible, sinon il faudra l'afficher via les méthodes du player). *True* par défaut si le paramètre est absent.

isInteractive : Si *true*, la figure répondre aux actions de l'utilisateur (capture de points, exécution de macros, etc ...). *True* par défaut si le paramètre est absent.

Les fonctions que l'objet *mtg32AppPlayer* peut appeler.

Tous les appels aux fonctions de *mtg32App* ont comme premier paramètre l'identificateur du svg associé à la figure ici appelé *idDoc*.

Nom de la fonction et paramètres	Explications
----------------------------------	--------------

activateField (<i>idDoc, nomCalcul</i>) :	Donne le focus au premier champ d'édition de la figure associé au calcul (ou à la fonction) nommé <i>nomCalcul</i> dans la figure MathGraph32 associée au SVG d'id <i>idDoc</i> .
activateFirstEmptyField (<i>idDoc</i>)	Donne le focus au premier champ d'édition vide de la figure (s'il y en a un).
activateFirstInvalidField (<i>idDoc</i>)	Donne le focus au premier champ d'édition de la figure qui comporte une faute de syntaxe dans la formule qu'il contient.
activateFirstInvalidOrEmptyField (<i>idDoc</i>)	Donne le focus au premier champ d'édition de la figure qui est soit vide soit contient une formule dont la syntaxe est incorrecte.
addDoc (<i>idDoc, chdoc, displayOnLoad, isActive</i>)	Ajoute une figure dans le SVG d'id <i>idDoc</i> , dont le code base64 est contenu dans <i>chDoc</i> . Si <i>displayOnLoad</i> est à true, la figure sera affichée dès que le chargement de la page est effectué. Si <i>isActive</i> est à true, la figure répondra aux actions sur la figure (capture de points, exécution de macros etc...) Le code base64 de la figure est une chaîne de caractères obtenue dans MathGraph32 par l'icone  d'exportation de la figure. Cette fonction est dépréciée et il vaut mieux procéder à un nouvel appel de <code>mtgLoad</code> comme expliqué ci-dessus.
calculate (<i>idDoc, brandom</i>)	Recalcule la figure contenue dans le SVG d'id <i>idDoc</i> . Si <i>brandom</i> est true, les calculs aléatoires sont réinitialisés. Sinon la figure apparaît la première fois telle qu'elle a été enregistrée par MathGraph32.
calculateAndDisplayAll (<i>brandom</i>)	Recalcule et réaffiche toutes les figures gérées par l'objet <code>mtg32App</code> . Si <i>brandom</i> est true, les calculs aléatoires sont réinitialisés pour chaque figure. Sinon chaque figure apparaît la première fois telle qu'elle a été enregistrée par MathGraph32.
calculateFirstTime (<i>idDoc, brandom</i>)	Calcule la figure d'id <i>idDoc</i> pour la première fois. Si <i>brandom</i> est true, les calculs aléatoires sont réinitialisés. Sinon la figure apparaît la première fois telle qu'elle a été enregistrée par MathGraph32. Il vaut mieux appeler cette fonction pour le premier calcul de la figure (pour certaines figures contenant par exemple une macro de démarrage).
createList (<i>code</i>)	Crée une liste d'objets à partir du code Base64 d'une figure. Seuls les éléments de type calcul de cette figure pourront ensuite être utilisés.
display (<i>idDoc</i>)	Détruit du DOM tous les éléments graphiques créés pas la figure MathGraph32 dans le SVG d'id <i>idDoc</i> , puis réaffiche toute la figure associée à ce SVG.
executeMacro (<i>idDoc, nameMacro</i>)	Lance l'exécution de la macro d'intitulé <i>nameMacro</i> dans la figure associée au SVG d'id <i>idDoc</i> .
fieldValidation (<i>idDoc, nomCalcul</i>)	Fonction renvoyant le booléen true si le premier champ d'édition associé au calcul <i>nomCalcul</i> est non vide et correct sur le plan syntaxique et false si la syntaxe est incorrecte ou si le champ n'existe pas.
fieldsValidation (<i>idDoc</i>)	Fonction renvoyant le booléen true si tous les champs d'édition de la figure associée au SVG d'id <i>idDoc</i> sont non vides et corrects sur le plan syntaxique.
getDoc (<i>idDoc</i>)	Renvoie le document (objet de type <code>CMathGraphDoc</code>) associé à la figure contenue dans le SVG d'id <i>idDoc</i> .
getFieldValue (<i>idDoc, nomCalcul</i>)	Renvoie le contenu du premier champ d'édition associé au calcul (ou à la fonction) <i>nomCalcul</i> dans la figure associée au SVG d'id <i>idDoc</i> .
getFormula (<i>idDoc, nomCalcul</i>)	Renvoie une chaîne de caractères contenant la formule du calcul ou de la fonction de nom <i>nomCalcul</i> dans la figure associée au SVG d'id <i>idDoc</i> .
getLatexCode (<i>idDoc, ind</i>)	Renvoie une chaîne de caractères contenant le code LaTeX de l'affichage LaTeX d'indice <i>ind</i> dans la liste des objets créés (les indices commençant à zéro) dans la figure associée au SVG d'id <i>idDoc</i> . <i>ind</i> peut aussi être une chaîne de caractère commençant par # suivi du tag de l'affichage LaTeX (si le tag n'est pas une chaîne vide).
getLatexFormula (<i>idDoc, nomCalcul</i>)	Renvoie une chaîne de caractères contenant le code LaTeX permettant d'afficher la formule du calcul (ou de la fonction) de nom <i>nomCalcul</i> .
getList (<i>idDoc</i>)	Renvoie la liste (objet <code>CListeObjets</code> de <code>mathGraph32</code>) qui contient tous les objets gérés par la figure MathGraph32 associée au SVG d'id <i>idDoc</i> .
giveFormula2 (<i>idDoc, nomCalcul, formule</i>)	Donne au calcul (ou à la fonction) de nom <i>nomCalcul</i> une formule égale à celle contenue dans la chaîne de caractères <i>formule</i> . S'il s'agit d'une fonction, le paramètre doit être celui utilisé dans la figure MathGraph32 pour cette fonction.
removeAllDoc ()	Retire toutes les figures gérées par <code>Mtg32App</code> et détruit les éléments graphiques correspondants du DOM. Retire aussi tous les gestionnaires d'événements associés à ces figures.

removeDoc (<i>idDoc</i>)	Retire la figure associée au SVG d'id <i>idDoc</i> et détruit les éléments graphiques correspondants du DOM. Retire aussi tous les gestionnaires d'événements associés à cette figure.
setActive (<i>idDoc</i> , <i>ba</i>)	Etablit le statut d'activité de la figure MathGraph32 associée au SVG d'id <i>idDoc</i> . Si <i>ba</i> est true, la figure est activée et répond aux événements souris et clavier (les éditeurs de formules sont activés). Sinon tous ces éléments sont inactifs.
setEditorCallBackOK (<i>idDoc</i> , <i>nomCalcul</i> , <i>f</i>)	Fonction associant au premier éditeur associé à <i>nomCalcul</i> la fonction de <i>callBack</i> <i>f</i> qui sera appelée quand l'utilisateur valide par OK le contenu de l'éditeur.
setEditorCharset (<i>idDoc</i> , <i>nomCalcul</i> , <i>st</i>)	Fonction attribuant au premier éditeur de formule associé au calcul (ou à la fonction) de nom <i>nomCalcul</i> une chaîne de caractères contenant les caractères autorisés dans cet éditeur. Si cette chaîne est vide, tous les caractères seront autorisés.
setEditorValue (<i>idDoc</i> , <i>nomCalcul</i> , <i>st</i>)	Remplace le contenu du premier éditeur associé au calcul (ou la fonction) de nom <i>nomCalcul</i> par le contenu de la chaîne de caractères <i>st</i> .
setEditorsEmpty (<i>idDoc</i>)	Vide le contenu de tous les éditeurs de formules contenus dans la figure MathGraph32 associée au SVG d'id <i>idDoc</i> .
setEditorsSize (<i>idDoc</i> , <i>size</i>)	Donne à tous les éditeurs de formules de la figure associée au SVG d'id <i>idDoc</i> la taille de caractères <i>size</i> .
updateDisplay (<i>idDoc</i>)	Réactualise l'affichage de tous les éléments graphiques de la figure contenue dans le SVG d'id <i>idDoc</i> (sans recalcul).
updateFigure (<i>idDoc</i>)	Recalcule la figure et réactualise l'affichage. A utiliser de préférence à <i>updateDisplay</i> quand on a directement modifié des éléments de la figure ou affecté des event listeners sur un ou plusieurs éléments de la figure. A noter que <i>updateFigure</i> peut être plus rapide que <i>calculate()</i> suivi de <i>updateDisplay()</i> mais ne doit pas être appelé pour un premier affichage de la figure.
valueOf (<i>idDoc</i> , <i>nomCalcul</i>)	Renvoie la valeur actuelle du calcul (réel) de nom <i>nomCalcul</i> contenu dans la figure associée au SVG d'id <i>idDoc</i> .
setVisible (<i>idDoc</i> , <i>id</i> , <i>bVisible</i> , <i>bImmediat</i> = true)	Sert à masquer (ou démasquer) l'objet déterminé par <i>id</i> du document d'id <i>idDoc</i> . <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide) Si <i>bVisible</i> est à true, on démasque l'objet, sinon on le masque. Si <i>bImmediat</i> est à true, on réaffiche la figure sinon on ne la réaffiche pas (il faudra alors la réafficher en appelant <i>updateFigure</i> après d'autres actions sur la figure).
setText (<i>idDoc</i> , <i>id</i> , <i>txt</i> , <i>bImmediat</i> = true)	Modifie le texte d'un affichage de texte ou d'un affichage LaTeX dans le document d'id <i>idDoc</i> . <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). <i>txt</i> est le nouveau texte à affecter à l'affichage. Si <i>bImmediat</i> est true, la figure est tout de suite réaffichée. Sinon il faudra la réafficher après d'autres actions. Renvoie true si tout s'est bien passé, false sinon.
setPointPosition (<i>idDoc</i> , <i>name</i> , <i>x</i> , <i>y</i> , <i>bImmediat</i> = true)	Fonction modifiant la position d'un point libre dans la figure d'id <i>idDoc</i> . <i>name</i> est soit le nom du point soit l'indice html du point (entier) soit une chaîne de caractères formée du caractère # suivi du tag de l'objet (si celui-ci n'est pas une chaîne vide). <i>x</i> et <i>y</i> sont les nouvelles coordonnées du point relativement au svg de la figure. Si <i>bImmediat</i> est true, la figure est tout de suite réaffichée. Sinon il faudra la réafficher après d'autres actions. Renvoie true si tout s'est bien passé, false sinon. Le point doit être un point libre.
translatePoint (<i>idDoc</i> , <i>name</i> , <i>deltax</i> , <i>deltay</i> , <i>bImmediat</i> = true)	Fonction translatant un point libre dans la figure d'id <i>idDoc</i> d'un vecteur de coordonnées (<i>deltax</i> , <i>deltay</i>). <i>name</i> est soit le nom du point soit l'indice html du point (entier) soit une chaîne de caractères formée du caractère # suivi du tag de l'objet (si celui-ci n'est pas une chaîne vide). Si <i>bImmediat</i> est true, la figure est tout de suite réaffichée. Sinon il faudra la réafficher après d'autres actions.

	<p>Renvoie true si tout s'est bien passé, false sinon. Le point doit être un point libre.</p>
getPointPosition (<i>idDoc</i> , <i>name</i>)	<p>Revoie un objet du type { <i>x</i> : <i>valx</i>, <i>y</i> : <i>valy</i> } contenant les coordonnées d'un point relativement au svg de la figure d'id <i>idDoc</i>. <i>name</i> est soit le nom du point soit l'indice html du point (entier) soit une chaîne de caractères formée du caractère # suivi du tag de l'objet (si celui-ci n'est pas une chaîne vide). Si l'appel est incorrect (par exemple <i>name</i> n'est pas le nom d'un point ni un id valide pour un élément) la fonction renvoie false, sinon elle renvoie true.</p>
setLineStyle (<i>idDoc</i> , <i>id</i> , <i>style</i> , <i>thickness</i> , <i>bImmediat</i> = true)	<p>Modifie le style d'affichage d'un objet de style ligne en lui donnant le style de trait donné par <i>style</i> (entier, voir ci-dessous) et l'épaisseur de trait <i>thickness</i>. <i>idDoc</i> est l'id du SVG contenant la figure. <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). Si <i>bImmediat</i> est true, la figure est tout de suite réaffichée. Sinon il faudra la réafficher après d'autres actions. Renvoie true si tout s'est bien passé, false sinon. Le point doit être un point libre. Valeurs possibles pour <i>lineStyle</i> : voir palette de MathGraph32 0 : trait continu 1 : trait pointillé 2 : trait trait (même longueur) 3 : trait point point 4 : trait point point point 5 : trait long trait court Renvoie true si tout s'est bien passé, false sinon.</p>
setColor (<i>idDoc</i> , <i>id</i> , <i>r</i> , <i>v</i> , <i>b</i> , <i>bImmediat</i> = true)	<p>Modifie la couleur d'un objet graphique en lui donnant les composantes RVB <i>r,v,b</i> (pour red, green, blue). <i>idDoc</i> est l'id du SVG contenant la figure. <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). Si <i>bImmediat</i> est true, la figure est tout de suite réaffichée. Sinon il faudra la réafficher après d'autres actions. Renvoie true si tout s'est bien passé, false sinon.</p>
getSVGElement (<i>idDoc</i> , <i>id</i>)	<p>Renvoie l'élément SVG associé à un objet graphique dans la figure d'indice <i>idDoc</i>. <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). Renvoie null en cas de problème.</p>
addEventListener (<i>idDoc</i> , <i>ind</i> , <i>eventName</i> , <i>callBack</i>)	<p>Ajoute un event Listener sur le composant SVG représentant un objet dans la figure d'id <i>idDoc</i>. <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). <i>eventName</i> est une chaîne de caractères qui peut être « mousemove » ou tout autre nom valide pour un event Listener sur un SVG element. Associe à cet event Listener la fonction de <i>callBack</i> passée dans le paramètre <i>callBack</i> en retirant une éventuelle fonction de <i>callBack</i> déjà assignée au même Listener sur cet objet. Renvoie true si tout s'est bien passé, false sinon.</p>
removeEventListener (<i>idDoc</i> , <i>id</i> , <i>eventName</i>)	<p>Retire la fonction de <i>callBack</i> sur l'événement de nom <i>eventName</i> associée composant SVG représentant un objet dans la figure d'id <i>idDoc</i>. <i>id</i> peut être soit l'indice html de l'objet (entier) soit une chaîne de caractères suivie du tag de l'élément (si celui-ci n'est pas une chaîne vide). Ne marche que si la fonction de <i>callBack</i> a été affectée par la fonction <i>addEventListener</i> du player MathGraph32.</p>

Comment utiliser une figure MathGraph32 pour faire des calculs et les exploiter en JavaScript.

Dans la fonction de callback passée en paramètre à la fonction *mtgLoad*, on crée une liste d'objets à l'aide d'une instruction du type *list* = *mtg32AppPlayer.createList(code)* où *code* est la chaîne base64 d'une figure MathGraph32, obtenue dans le logiciel par Edition – Copier le code de la figure (pour html).
C'est cette figure qui contient les objets de type calcul qu'on souhaite exploiter.

On peut ensuite modifier les objets numériques de cette liste à l'aide des instructions ci-dessous à appeler par la liste.

calculateNG (<i>infoRandom</i>)	Recalcule tous les éléments de la liste. Si <i>infoRandom</i> est true, tous les calculs aléatoires de la figure sont réactualisés.
getLatexFormula (<i>nomCalcul</i>)	Renvoie une chaîne LaTeX contenant le code LaTeX représentant <i>nomCalcul</i> . <i>nomCalcul</i> peut être le nom d'un calcul réel, ou complexe ou le nom d'une fonction réelle ou complexe (de une ou plusieurs variables).
giveFormula2 (<i>nomCalcul</i> , <i>formule</i>)	Donne à l'objet de nom <i>nomCalcul</i> la formule contenue dans <i>formule</i> . <i>nomCalcul</i> peut être le nom d'un calcul réel, ou complexe ou le nom d'une fonction réelle ou complexe (de une ou plusieurs variables).
valueOf (<i>nomCalcul</i>)	Renvoie la valeur actuelle du calcul (réel) de nom <i>nomCalcul</i> contenu dans la liste.

Comment modifier temporairement le pointeur souris.

Cela concerne les utilisateurs avancés qui veulent gérer eux-mêmes certains éléments SVG de la figure, par exemple pour changer le pointeur souris au survol d'un polygone de la figure.

Pour modifier le pointeur souris dans la figure associée au SVG d'idDoc utilisez un code du type :

```
var doc = mtgAppPlayer.getDoc(idDoc)
```

```
doc.defaultCursor = 'pointer' (Ici pur avoir un pointeur souris représentant une main)
```

Pour redonner au pointeur souris son apparence par défaut , utilisez le code :

```
var doc = mtgApp.getDoc('svgMtg0')
```

```
doc.defaultCursor = null
```