

PLAN DE TESTS

TESTS FONCTIONNELS.....	2
TEST INTERACTION AVEC LE TABLEAU.....	2
TEST VÉRIFICATION DU TABLEAU.....	6
TEST MANIPULATION DE LA ZONE TEXTUELLE.....	10
TEST AFFICHAGE DE L'AIDE.....	11
TESTS UNITAIRES.....	12
VÉRIFICATION ET VALIDATION DU TABLEAU.....	12
Comparer les valeurs d'une ligne de tableau.....	12
Validation de la fonction principale.....	13
Vérification de la dérivée.....	14
COMMANDE DEPUIS LA ZONE TEXTUELLE.....	14
Création d'une fonction pour exécution.....	14
Exécution d'une fonction dans le module tabvar depuis une entrée textuelle.....	15

TESTS FONCTIONNELS

TEST INTERACTION AVEC LE TABLEAU

Intitulée du test :	Entrée de la fonction principale
Scénarios :	<ol style="list-style-type: none">1. Entrer un nom pour la fonction.2. Entrer un nom pour la variable.3. Entrer une expression de cette fonction.
Attendu :	<ol style="list-style-type: none">1. Tous les éléments du tableau utilisant ce nom de fonction doivent prendre ce nom.2. Tous les éléments du tableau utilisant ce nom de variable doivent prendre ce nom.3. La fonction entrée doit utiliser la variable définie.
Succès/Échec :	<ol style="list-style-type: none">1. Succès.2. Succès.3. Échec.
Description du résultat :	<ol style="list-style-type: none">1. Les noms changent visuellement.2. Les noms changent visuellement.3. Cette étape ne permet pas cette vérification (nécessite la commande "valider main fonction" dans les test de vérification du tableau.

Intitulée du test :	Entrée de la dérivée
Scénario :	<ol style="list-style-type: none">1. Entrer l'expression de la dérivée.
Attendu :	<ol style="list-style-type: none">1. Rien.
Succès/Échec :	<ol style="list-style-type: none">1. Succès.
Description du résultat :	<ol style="list-style-type: none">1. Une dérivée est entrée dans la zone dédiée (aucune vérification à cette étape).

Intitulée du test :	Changer la valeur d'une cellule
Scénarios :	<ol style="list-style-type: none"> 1. Changer les valeurs des cellules de la première ligne. 2. Changer les valeurs des cellules de la deuxième ligne. 3. Changer les valeurs des cellules de la troisième ligne.
Attendu :	<ol style="list-style-type: none"> 1. Affichage de lien cliquable (changer) au survole des valeurs. Change la valeur en la valeur éditée. 2. Affichage de lien cliquable proposant les signes et les identités des valeurs de x (extremum, valeurs interdite) au survole des valeurs. Remplace la valeur du bouton avec l'option choisie. 3. Affichage de lien cliquable proposant les variations de la fonction. Remplace la valeur du bouton avec l'option choisie.
Succès/Échec :	<ol style="list-style-type: none"> 1. Échec. 2. Succès. 3. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. L'ajout d'un caractère vide supprime la zone d'édition. 2. Les changements sont exécutés sans problème (visuellement). 3. Les changements sont exécutés sans problème (visuellement).

Intitulée du test :	Ajouter une colonne
Scénario :	<ol style="list-style-type: none"> 1. Cliquez sur le bouton (+) de la première ligne.
Attendu :	<ol style="list-style-type: none"> 1. Ajout d'une colonne avec une valeur et d'une autre colonne avec le bouton (+).
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. L'ajout de colonne ajoute une valeur 0 et les boutons des colonnes n'ont pas de valeur initial

Intitulée du test :	Supprimer une colonne
Scénario :	1. En survolant une cellule de la première ligne, cliquer sur l'option (supprimer).
Attendu :	1. La colonne correspondant à la position de la cellule supprimée doit disparaître.
Succès/Échec :	1. Succès.
Description du résultat :	1. Les changements sont exécutés comme attendu.

Intitulée du test :	Ajouter et supprimer une sous-fonction
Scénarios :	<ol style="list-style-type: none"> 1. Cliquer sur le bouton (Ajouter une ligne). 2. Ajouter une expression dans la première cellule de ligne ajoutée. 3. Supprimer une sous-fonction
Attendu :	<ol style="list-style-type: none"> 1. Une ligne doit être ajoutée à en deuxième position dans le tableau. La première cellule de cette ligne contient le nom d'une nouvelle expression. 2. Au survol de la première cellule, il devrait être proposé une option de changement (changer). 3. Au survol de la première cellule, il devrait être proposé une option de suppression (supprimer). Le clique de ce lien devrait supprimer la ligne.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Échec. 3. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. La ligne est correctement ajoutée en dupliquant les éléments de la ligne précédente et avec un nouveau nom de fonction dans la première cellule. 2. Suppression de la zone d'édition quand l'expression saisie est vide. 3. La suppression de la ligne courante s'exécute.

Intitulée du test :	Supprimer toutes les sous-fonctions
Scénario :	1. Supprimer toutes les lignes ajoutées pour les sous-fonction en cliquant sur le bouton (Supprimer les lignes intermédiaires).
Attendu :	1. A l'affichage, seules les lignes liées à la fonction principale et à la dérivée devraient être présentes sur la tableau (les valeurs de la variable, les signes de la dérivée, les variations de la fonction).
Succès/Échec :	1. Succès.
Description du résultat :	1. Les modifications s'exécutent comme prévu.

Intitulée du test :	Sauvegarder les données
Scénario :	1. Sauvegarder les données du tableau actuel en cliquant sur le bouton (Sauvegarder) en haut à gauche.
Attendu :	1. Un fichier .JSON devra être créé contenant les données du tableau.
Succès/Échec :	1. Échec.
Description du résultat :	1. Ne sauvegarde pas toutes les données du tableau. Seules les données des valeurs de la première ligne, les données contenant les signes de la dérivée et celles contenant les variations de la fonction sont récupérées dans le JSON pour le moment.

TEST VÉRIFICATION DU TABLEAU

Intitulée du test :	Valider la fonction principale
Scénarios :	<p>S'assurer que l'expression de la fonction principal soit validé par le modul de tabvar</p> <ol style="list-style-type: none"> 1. Entré une expression avec comme la variable celle saisi juste avant (x par défaut). Cliquer sur le bouton (" valider main function") 2. Entré une expression avec comme la variable celle qui n'est pas saisi juste avant. Cliquer sur le bouton (" valider main function") 3. Mettre une expression invalide avec des caractères spécifiques. Cliquer sur le bouton ("valider main function").
Attendu :	<ol style="list-style-type: none"> 1. Message de validation + Expression de la dérivée saisie. 2. Message de non validation. 3. Message de non validation.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Succès. 3. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. Le message attendu est affiché. 2. Le message attendu est affiché. 3. Le message attendu est affiché.

Intitulée du test :	Valider la dérivée
Scénarios :	<p>S'assurer que la dérivée entrée est une expression valide en cliquant sur le bouton ("valider deriv function").</p> <ol style="list-style-type: none"> 1. Mettre une expression contenant la même variable que celle de la fonction principale. Cliquer sur le bouton ("valider deriv function"). 2. Mettre une expression contenant une variable différente de celle de la fonction principale. Cliquer sur le bouton ("valider deriv function"). 3. Mettre une expression invalide avec des caractères spécifiques. Cliquer sur le bouton ("valider deriv function").

Attendu :	<ol style="list-style-type: none"> 1. Message de validation + Expression de la dérivée saisie. 2. Message de non validation. 3. Message de non validation.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Succès. 3. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. Le message attendu est affiché. 2. Le message attendu est affiché. 3. Le message attendu est affiché.

Intitulée du test :	Vérifier la dérivée
Scénarios :	<p>S'assurer que la dérivée entrée est bonne en cliquant sur le bouton ("verifier deriv function"). Saisir l'expression de la fonction principale.</p> <ol style="list-style-type: none"> 1. Mettre une expression dans la zone dédiée à la dérivée contenant la même variable que la fonction principale. Cliquer sur le bouton ("verifier deriv function"). 2. Mettre la dérivée exacte de l'expression de la fonction principale. Cliquer sur le bouton ("verifier deriv function"). 3. Mettre la bonne dérivée de la fonction principale mais la fonction principale n'a pas la même variable.
Attendu :	<ol style="list-style-type: none"> 1. _ Message de validation si la dérivée est bonne . _ Message de non validation si la dérivée n'est pas bonne. 2. Message de validation. 3. Message d'erreur.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Succès. 3. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. Le message attendu est affiché. 2. Le message attendu est affiché. 3. Le message d'erreur attendu est affiché

Intitulée du test :	Comparer les valeurs de x
Scénarios :	<p>S'assurer que les valeurs de x sont dans l'ordre croissant.</p> <ol style="list-style-type: none"> 1. Dans les cases prévues à cet effet, entrer des valeurs croissantes, puis cliquer sur "comparer les valeurs de x". 2. Dans les cases prévues à cet effet, entrer des valeurs décroissantes, puis cliquer sur "comparer les valeurs de x". 3. Dans les cases prévues à cet effet, entrer des valeurs équivalentes, puis cliquer sur "comparer les valeurs de x". 4. Dans les cases prévues à cet effet, entrer autre que des valeurs, puis cliquer sur "comparer les valeurs de x".
Attendu :	<ol style="list-style-type: none"> 1. Les cases sont surlignées en vert. 2. Les cases sont surlignées en rose, sauf la première car valeur de référence. 3. Les cases sont surlignées en jaune, sauf la première car valeur de référence. 4. Les cases sont surlignées en rouge, sauf la première car valeur de référence.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Succès. 3. Succès. 4. Échec.
Description du résultat :	<ol style="list-style-type: none"> 1. Toutes les cases des valeurs sont bien surlignées en vert. 2. Toutes les cases des valeurs sont bien surlignées en rose, sauf la première. 3. Toutes les cases des valeurs sont bien surlignées en jaune, sauf la première. 4. Toutes les cases des valeurs sont bien surlignées en rouge sauf la première.

Intitulée du test :	Valider les sous-fonctions
Scénarios :	<p>Entrer les expressions des sous fonctions dans la première case des nouvelles lignes et s'assurer qu'elles soient valides en cliquant sur le bouton "Les sous fonctions".</p> <ol style="list-style-type: none"> 1. Une expression des sous fonctions contient la bonne variable ou aucune. 2. Une expression des sous fonctions contient la mauvaise variable ou aucune.
Attendu :	<ol style="list-style-type: none"> 1. _ Sous fonction valide : Elle est surlignée en vert et un message contenant l'expression apparaît. _ Sous fonction non valide : Elle est surlignée en rose et un message d'erreur apparaît. 2. Elle est surlignée en rose et un message d'erreur apparaît.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès / Échec 2. Échec.
Description du résultat :	<ol style="list-style-type: none"> 1. Succès quand l'expression est valide sinon échec car le message n'apparaît pas mais surligné en rose. 2. Le message n'apparaît pas, mais surligné en rose.

Intitulée du test :	Validation totale du tableau
Scénarios :	<p>S'assurer que tout ce qui concerne le tableau de variation soit valide et vérifié.</p> <ol style="list-style-type: none"> 1. La fonction principale n'est pas valide mais le reste oui. 2. La dérivé n'est pas valide mais le reste oui. 3. La dérivé n'est pas vérifié mais le reste oui. 4. Rien n'est ni validé ni vérifié. 5. Tout est validé et vérifié.
Attendu :	<ol style="list-style-type: none"> 1. Traitement seulement de la validation de la fonction principale. 2. Traitement seulement de la validation de la fonction principale et de la dérivée.

	<ol style="list-style-type: none"> 3. Traitement de toutes les validations et des vérifications. 4. Traitement juste de la fonction principale. 5. Traitement de toutes les validations et des vérifications.
Succès/Échec :	<ol style="list-style-type: none"> 1. Succès. 2. Succès. 3. Succès. 4. Succès. 5. Succès.
Description du résultat :	<ol style="list-style-type: none"> 1. Affichage du message attendu uniquement en rapport avec la fonction principale. 2. Affichage du message de la validation de la fonction principale et de la dérivé et message d'erreur. 3. Affichage des messages des validations et surlignage des cases. 4. Affichage du message attendu uniquement en rapport avec la fonction principale. 5. Affichage des messages des validations et surlignage des cases.

TEST MANIPULATION DE LA ZONE TEXTUELLE

Intitulée du test :	Entrée une commande
Scénarios :	<p>Aller dans la partie de l'interface dédiée à la manipulation des commandes de manière textuelle (en bas à gauche).</p> <ol style="list-style-type: none"> 1. Saisir une chaîne de caractères sans choisir de commandes parmi celles proposées s'il y en a. Exécuter avec la touche "Entrée" 2. Saisir une commande existante. Possibilité d'auto-complétion en choisissant une des commandes proposées. Exécuter avec la touche "Entrée"
Attendu :	<ol style="list-style-type: none"> 1. [Message d'erreur 1.1 : Commande inexistante.] 2. _ Si la commande ne nécessite pas de paramètres alors : [Exécution de la fonction + Message de retour de fonction du module]

	tabvar associé.] _Si la commande nécessite des paramètres alors : [Message d'erreur 1.1 : Paramètres manquants]
Succès/Échec :	1. Échec. 2. Succès.
Description du résultat :	1. Affichage du message d'erreur en doublon. 2. Le message attendu est affiché. (RMQ : des messages d'erreur, liés à l'exécution de la fonction dans le module tabvar peuvent s'afficher).

Intitulée du test :	Exécuter une commande existante
Scénario :	Aller dans la partie de l'interface dédiée à la manipulation des commandes de manière textuelle (en bas à gauche). 1. Saisir une commande existante. Possibilité d'auto-complétion en choisissant une des commandes proposées avec les paramètres nécessaires. Exécuter avec la touche "Entrée"
Attendu :	1. Affichage sans erreur du retour de la fonction depuis le module tabvar.
Succès/Échec :	1. Échec.
Description du résultat :	1. _ Certaines fonctions retournent des erreurs indéfinies en dépit de l'application du paramétrage adéquat.

TEST AFFICHAGE DE L'AIDE

Intitulée du test :	Rechercher une fonction
Scénarios :	Aller dans la partie de l'interface dédiée à la recherche d'information sur les fonctions (en bas à droite). 1. Saisir une commande existante. Possibilité d'auto-complétion en

	choisissant une des commandes proposées avec les paramètres nécessaires. Exécuter avec la touche "Entrée" ou Appuyer sur le bouton à côté de la zone de texte. 2. Exécuter une recherche pour une chaîne vide ou une chaîne inexistante.
Attendu :	1. Affichage des informations sur la fonction recherchée. 2. Message d'erreur [1.1 : Commande inexistante] + Structure d'affichage vide.
Succès/Échec :	1. Succès. 2. Succès.
Description du résultat :	1. Retourne l'affichage attendu. 2. Retourne l'affichage attendu + Message d'erreur.

TESTS UNITAIRES

VÉRIFICATION ET VALIDATION DU TABLEAU

Comparer les valeurs d'une ligne de tableau.

Identification:

Fonction → `compareValuesInRow()`

Utilisation dans le programme : Comparaison des valeurs de la première ligne du tableau de variation.

Utilisation dans le module de gestion interne : Module de vérification et de validation du tableau (voir documentation interne).

Description:

Vérifie que les valeurs d'une ligne sont comparées deux à deux.

Contraintes:

Importation du module `tabvar`. Utilisation des fonctions du module `tabvar`.

Dépendances:

- _ Importation du module tabvar. Utilisation des fonctions du module tabvar.
- _ Utilisation de la fonction tabvar.compare_values() du module tabvar.
- _ Utilisation de la fonction getValues() du module de gestion du tableau (voir documentation interne) pour la récupération des valeurs nécessaires.

Procédure de Test:

1. Vérification de l'existence du module tabvar contenant la fonction de comparaison. **géré**
2. Vérification de l'existence du tableau (défini par l'id 'tabvar'). **géré**
3. Vérification des retours de comparaison. **géré**

Validation de la fonction principale

Identification:

Fonction → validation_funMain()

Utilisation dans le programme : Validation de la fonction saisie par l'utilisateur depuis la zone d'édition.

Utilisation dans le module de gestion interne : Module de vérification et de validation du tableau (voir documentation interne).

Description:

Vérifie si la fonction à traiter est correctement définie.

Contraintes:

Importation du module tabvar. Utilisation des fonctions du module tabvar.

Dépendances:

- _ Importation du module tabvar. Utilisation des fonctions du module tabvar.
- _ Utilisation de la fonction tabvar.validate() du module tabvar.
- _ Utilisation de la fonction tabvar.suppose_var() du module tabvar.
- _ Utilisation de la fonction tabvar.function_definition() du module tabvar.

Procédure de Test:

1. Vérification de l'existence du module tabvar contenant les différentes fonctions utilisées. **géré**
2. Vérification de l'existence des éléments nécessaires pour l'exécution des fonctions du module tabvar. **géré**
3. Vérification des retours de validation. **géré**
4. Vérification des retours de définition de fonction. **géré**

Vérification de la dérivée

Identification:

Fonction → `verificatio_funDeriv()`

Utilisation dans le programme : Vérification de l'expression de la dérivée saisie par l'utilisateur depuis la zone d'édition.

Utilisation dans le module de gestion interne : Module de vérification et de validation du tableau (voir documentation interne).

Description:

Vérifie que la dérivée saisie est bien une dérivée de la fonction principale.

Contraintes:

Importation du module `tabvar`. Utilisation des fonctions du module `tabvar`.

Dépendances:

- _ Importation du module `tabvar`. Utilisation des fonctions du module `tabvar`.
- _ Utilisation de la fonction `tabvar.deriv()` du module `tabvar`.
- _ Utilisation de la fonction `tabvar.compare_values()` du module `tabvar`.
- _ Utilisation de la fonction `tabvar.develop()` du module `tabvar`.

Procédure de Test:

1. Vérification de l'existence du module `tabvar` contenant les différentes fonctions utilisées. **géré**
2. Vérification de l'existence des éléments nécessaires pour l'exécution des fonctions du module `tabvar`. **géré**
3. Vérification des retours de vérification de la dérivée. **géré**

COMMANDE DEPUIS LA ZONE TEXTUELLE

Création d'une fonction pour exécution.

Identification:

Fonction → `parseFunctionString()`

Utilisation dans le programme : Entrée d'une commande dans la zone textuelle.

Utilisation dans le module de gestion interne : Module de gestion de la manipulation de la gestion de la zone textuelle (voir documentation interne).

Description:

Vérifier si le nom d'une fonction avec ses paramètres matche. Si on entre une chaîne de caractères dans une zone de texte, on veut qu'elle soit considérée comme une fonction.

Contraintes:

Nécessite une fonction de récupération des données saisies dans le champ de texte dédié. Dans le programme, nous utilisons une fonction d'autocomplétion via une bibliothèque externe JQuery.

Dépendances:

_ Importation d'une bibliothèque externe dans le programme final.

Procédure de Test:

1. Vérification du format de texte saisi. **géré**

Exécution d'une fonction dans le module tabvar depuis une entrée textuelle.

Identification:

Fonction → `parseFunctionString()`

Utilisation dans le programme : Exécution de la commande saisie dans la zone textuelle.

Utilisation dans le module de gestion interne : Module de gestion de la manipulation de la gestion de la zone textuelle (voir documentation interne).

Description:

On souhaite que si la fonction saisie est une fonction du module tabvar avec les bons paramètres, alors elle soit exécutée.

Contraintes:

Importation du module tabvar. Utilisation des fonctions du module tabvar. Doit utiliser la fonction de création de nom de fonction `parseFunctionStr()`.

Dépendances:

_ Importation du module tabvar. Utilisation des fonctions du module tabvar. (Celles dont on connaît le fonctionnement).

Procédure de Test:

1. Vérification de l'existence du module tabvar contenant les différentes fonctions utilisées. **géré**
2. Vérification de l'existence des éléments nécessaires pour l'exécution des fonctions du module tabvar. **géré**
3. Vérification des retours possibles. **non géré**