Plan de test
Orinoco

FLORAND Grégory

17 Février 2021

Introduction

Vous retrouverez dans ce plan de test, l'intégralité des fonctions composant le code de l'application.

Pour plus de clarté, les fonctions ont été réunies par feuille de scripts.

Chaque test est établi selon la même structure, ce qui permet une meilleure compréhension.

Les tests

home.js

Affiche tous les produits

- ***Description :*** récupère tous les produits de l'api et les itèrent à l'intérieur de la cible passée en paramètre
- ***Fonction à tester :*** Home.showAllProducts (ligne 16 à 30)
- ***Déclencheur :*** au chargement de la page
- ***Arguments :*** la cible où afficher le contenu
- ***Retour attendu :*** doit afficher tous les produits sur la page d'accueil
- ***Problème possible :*** Ne reçoit pas les données de l'API ou du dataManager. Consulter le log d'erreur dans la console.

Affiche un produit sous forme de carte

- ***Description :*** Génère le HTML du produit sous forme de carte et affiche les propriétés de l'objet passé en paramètre
- ***Fonction à tester :*** Home.productHtml (ligne 39 à 56)
- ***Déclencheur :*** la méthode showAllProducts() au chargement de la page
- ***Arguments :*** les propriétés de l'objet à afficher
- ***Retour attendu :*** une carte avec la photo, le nom, le prix et autres propriétés de l'objet
- ***Problème possible :*** Ne reçoit pas les propriétés de l'objet à afficher

Affiche les couleurs sous forme de pastille

- ***Description :*** réalise une boucle sur la longueur de liste de couleur et la renvoit sous forme de pastille colorée
- ***Fonction à tester :*** Home.showColor (ligne 65 à 71)
- ***Déclencheur :*** la méthode productHtml() au chargement de la page
- ***Arguments :*** la liste de couleur de l'objet à aficher
- ***Retour attendu :*** une pastille de couleur pour chaque couleur trouvée dans la liste correspondante
- ***Problème possible :*** Ne reçoit pas les couleurs de l'objet à afficher

Transforme un nom de couleur en nom de classe

- ***Description :*** prend le nom en paramètre, le transforme en camelCase et ajoute "Color" à la fin du mot
- ***Fonction à tester :*** Home.convertToClassName (ligne 80 à 92)
- ***Déclencheur :*** la méthode showColor() au chargement de la page

- ***Arguments :*** la couleur à modifier
- ***Retour attendu :*** un couleur telle que “Dark brown” doit retourner “darkBrownColor”
- ***Problème possible :*** Ne reçoit pas la couleur à transformer

product.js

Affiche un produit unique sous forme de carte

- ***Description :*** Récupère le produit grâce à l’id passé en 2ème paramètre et l’affiche dans la cible passé en premier paramètre
- ***Fonction à tester :*** Product.showProduct (ligne 16 à 20)
- ***Déclencheur :*** Au chargement de la page
- ***Arguments :*** la cible en 1er argument et l’id du produit en second
- ***Retour attendu :*** une carte avec la photo, le nom, le prix et autres propriétés de l’objet spécifié
- ***Problème possible :*** Ne reçoit pas l’id de l’objet à afficher

Surveille le changement de quantités

- ***Description :*** Écoute le clic sur les boutons “+”, “-” et “ajouter au panier” et lance la méthode appropriée
- ***Fonction à tester :*** Product.initQtySelector (ligne 27 à 35)
- ***Déclencheur :*** la méthode showProduct() au chargement de la page
- ***Arguments :*** aucun
- ***Retour attendu :*** doit lancer la méthode correspondant au bouton cliqué
- ***Problème possible :*** rien ne se produit au clic sur les boutons

Génère le HTML du produit

- ***Description :*** affiche une carte avec les infos correspondant aux propriétés passées en paramètre
- ***Fonction à tester :*** Product.productHtml (ligne 44 à 78)
- ***Déclencheur :*** la méthode showProduct() au chargement de la page
- ***Arguments :*** les propriétés de l’objet à afficher
- ***Retour attendu :*** une carte avec l’image, le nom et autres propriétés, un choix de quantité et un bouton pour ajouter au panier
- ***Problème possible:*** Ne reçoit pas les infos du produit à afficher

Affiche les couleurs sous forme de menu déroulant

- ***Description :*** réalise une boucle sur la longueur de liste de couleur et et renvoit chaque itération sous forme d'option
- ***Fonction à tester :*** Product.showOptionColor (ligne 88 à 94)
- ***Déclencheur :*** la méthode productHtml() au chargement de la page
- ***Arguments :*** la liste de couleur de l'objet à aficher
- ***Retour attendu :*** un menu déroulant avec toutes les options correspondant aux couleurs du produit
- ***Problème possible :*** Ne reçoit pas les couleurs de l'objet à afficher

Ajoute une majuscule à chaque début de mot

- ***Description :*** Met tout en minuscule puis change la première lettre de chaque mot en majuscule
- ***Fonction à tester :*** Product.convertToDisplayName (ligne 103 à 114)
- ***Déclencheur :*** La méthode showOptionColor() au chargement de la page
- ***Arguments :*** Le nom de la couleur à modifier
- ***Retour attendu :*** un nom comme "Pale brown" doit retourner "Pale Brown"
- ***Problème possible :*** Ne reçoit pas les couleurs de l'objet à modifier

Affiche les couleurs sous forme de pastille

- ***Description :*** réalise une boucle sur la longueur de liste de couleur et la renvoit sous forme de pastille colorée
- ***Fonction à tester :*** Product.showColor (ligne 123 à 129)
- ***Déclencheur :*** la méthode productHtml() au chargement de la page
- ***Arguments :*** la liste de couleur de l'objet à aficher
- ***Retour attendu :*** une pastille de couleur pour chaque couleur trouvée dans la liste correspondante
- ***Problème possible :*** Ne reçoit pas les couleurs de l'objet à afficher

Transforme un nom de couleur en nom de classe

- ***Description :*** prend le nom en paramètre, le transforme en camelCase et ajoute "Color" à la fin du mot
- ***Fonction à tester :*** Product.convertToClassName (ligne 138 à 149)
- ***Déclencheur :*** la méthode showColor() au chargement de la page
- ***Arguments :*** la couleur à modifier
- ***Retour attendu :*** un couleur telle que "Dark brown" doit retourner "darkBrownColor"

- ***Problème possible :*** Ne reçoit pas la couleur à transformer

Réduit de 1 la valeur du nombre

- ***Description :*** récupère le chiffre présent dans le champ "field" et le réduit de 1 jusqu'à 1 minimum, sinon ne fait rien
- ***Fonction à tester :*** Product.decrementInput (ligne 154 à 161)
- ***Déclencheur :*** Au clic sur le bouton “-” grâce à la méthode initQtySelector()
- ***Arguments :*** aucun
- ***Retour attendu :*** le nombre affiché dans le champ avec l’id “field” descend de 1 au clic jusqu’à 1 minimum puis ne descend plus
- ***Problème possible :*** la valeur ne change pas au clic

Augmente de 1 la valeur du nombre

- ***Description :*** récupère le chiffre présent dans le champ "field" et l’augmente de 1
- ***Fonction à tester :*** Product.incrementInput (ligne 166 à 171)
- ***Déclencheur :*** Au clic sur le bouton “+” grâce à la méthode initQtySelector()
- ***Arguments :*** aucun
- ***Retour attendu :*** le nombre affiché dans le champ avec l’id “field” monte de 1 au clic
- ***Problème possible :*** la valeur ne change pas au clic

Ajoute le produit et sa quantité au panier

- ***Description :*** Récupère l’id dans les paramètres URl ainsi que la valeur champ portant l’id “field” et l’ajoute au composant panier. Puis affiche une fenêtre modal pour orienter l’utilisateur
- ***Fonction à tester :*** Product.addToCart (ligne 177 à 198)
- ***Déclencheur :*** Au clic sur le bouton “ajouter au panier” grâce à la méthode initQtySelector()
- ***Arguments :*** aucun

- ***Retour attendu :*** la valeur du panier en haut à droite doit augmenter du même montant que le champ à l'id "field". Une fenêtre modal doit s'ouvrir pour confirmer le nombre d'objets ajoutés
- ***Problème possible :*** le nombre sur le panier n'augmente pas lorsqu'on appuie sur "ajouter au panier"

panier.js

Lance la méthode de template selon l'état du panier

- ***Description :*** Si le panier est vide, lance la méthode templateEmptyCart(), sinon la méthode templateProduct(). Calcule le total du panier et passe la valeur à la méthode displayTotal()
- ***Fonction à tester :*** Panier.displayCart (ligne 22 à 50)
- ***Déclencheur :*** Au chargement de la page ainsi qu'avec les méthodes increment(), decrement() et deleteLine()
- ***Arguments :*** aucun
- ***Retour attendu :*** affiche le panier si au moins un objet, sinon une phrase qui décrit l'état du panier
- ***Problème possible :*** Ne peut pas récupérer les produits du panier. Si une erreur arrive, alors la méthode templateError() est appliquée

Transforme un tableau en objet et lui attribue une quantité

- ***Description :*** Récupère le contenu du composant panier sous forme d'objet et ajuste sa propriété "qté" en fonction du nombre d'id similaires dans la liste
- ***Fonction à tester :*** Panier.arrayToObject (ligne 59 à 68)
- ***Déclencheur :*** Au chargement de la page
- ***Arguments :*** Le contenu du composant panier
- ***Retour attendu :*** Un objet avec l'id du produit et une propriété "qté" et le nombre

- ***Problème possible :*** N'arrive pas à récupérer le tableau du composant panier

Affiche une ligne avec les infos du produit

- ***Description :*** renvoie un bloc de code HTML
- ***Fonction à tester :*** Panier.templateProduct (ligne 83 à 111)
- ***Déclencheur :*** Dans la méthode displayCart() si il y a au moins un produit dans le composant panier
- ***Arguments :*** un objet contenant les propriétés du produit
- ***Retour attendu :*** une ligne de tableau avec les infos propres au produit
- ***Problème possible :*** n'affiche rien alors qu'il y a un produit dans le panier

Affiche une ligne expliquant que le panier est vide

- ***Description :*** renvoie un bloc de code HTML
- ***Fonction à tester :*** Panier.templateEmptyCart (ligne 116 à 120)
- ***Déclencheur :*** Dans la méthode displayCart() si il n'y a aucun produit dans le composant panier
- ***Arguments :*** rien
- ***Retour attendu :*** une ligne de texte explicatif à la place du tableau
- ***Problème possible :*** affiche le tableau alors que le panier est vide

Affiche une ligne avec un message d'erreur

- ***Description :*** renvoie un bloc de code HTML
- ***Fonction à tester :*** Panier.templateError (ligne 125 à 129)
- ***Déclencheur :*** Dans la méthode displayCart() si le catch a été exécuté
- ***Arguments :*** aucun
- ***Retour attendu :*** un message d'erreur à la place du tableau
- ***Problème possible :*** affiche le tableau alors qu'une erreur a été relevée

Affiche le prix total du panier

- ***Description :*** affiche une ligne au tableau avec un nombre qui représente la valeur totale du panier et passe la valeur du total à la méthode wachClick()
- ***Fonction à tester :*** Panier.displayTotal (ligne 138 à 147)
- ***Déclencheur :*** Dans la méthode displayCart()
- ***Arguments :*** La valeur total du panier
- ***Retour attendu :*** Une ligne de tableau avec “Total du panier =” et la valeur totale
- ***Problème possible :*** Ne reçoit pas de valeur sous forme de nombre

Augmente 1 produit au panier

- ***Description :*** ajoute l’id du produit ciblé en paramètre au composant panier. Incrémente de 1 la propriété “qté” du produit ciblé et actualise l’affichage
- ***Fonction à tester :*** Panier.increment (ligne 154 à 158)
- ***Déclencheur :*** au clic sur le “+” de la ligne grâce à la méthode templateProduct()
- ***Arguments :*** l’id du produit concerné
- ***Retour attendu :*** lorsqu’on clique, augmente de 1 la valeur du champ et du composant panier (en haut à droite)
- ***Problème possible :*** L’argument n’est pas un string ou ne renvoie à aucun id

Enlève 1 produit au panier

- ***Description :*** retire l'id du produit ciblé en paramètre au composant panier. Décrémente de 1 la propriété "qté" du produit ciblé, exécute la méthode deletLine()si la quantité tombe à 0, puis actualise l'affichage
- ***Fonction à tester :*** Panier.decrement (ligne 165 à 170)
- ***Déclencheur :*** au clic sur le "-" de la ligne grâce à la méthode templateProduct()
- ***Arguments :*** l'id du produit concerné
- ***Retour attendu :*** lorsqu'on clique, diminue de 1 la valeur du champ et du composant panier (en haut à droite)
- ***Problème possible :*** L'argument n'est pas un string ou ne renvoie à aucun id

Retire la ligne produit du panier

- ***Description :*** ouvre un modal pour confirmer le choix, puis supprime tous les id qui correspondent à celui passé en paramètre du composant panier, retire la ligne du produit du tableau et actualise l'affichage
- ***Fonction à tester :*** Panier.deleteLine (ligne 177 à 193)
- ***Déclencheur :*** au clic sur l'icône poubelle de la ligne grâce à la méthode templateProduct() ainsi que dans la méthode decrement() si la quantité descend à 0
- ***Arguments :*** l'id du produit concerné
- ***Retour attendu :*** lorsqu'on clique et que l'on valide le modal qui s'ouvre, cela retire la ligne du tableau ainsi que la quantité du composant panier (en haut à droite)

- ***Problème possible :*** L'argument n'est pas un string ou ne renvoie à aucun id

Vérifie que le champs est correct

- ***Description :*** Vérifie si le champ requis est valide. Sinon, affiche un message explicatif
- ***Fonction à tester :*** Panier.checkField (ligne 203 à 205)
- ***Déclencheur :*** à la saisie d'un des champs du formulaire par la méthode displayForm()
- ***Arguments :*** l'input ciblée en 1ère argument et le message à afficher en second paramètre
- ***Retour attendu :*** Rien si le champs est bon, un message explicatif en dessous du champs si cela ne correspond pas
- ***Problème possible :*** n'affiche pas le message s'il y a une erreur

Vérifie si le navigateur possède déjà des infos de contact

- ***Description :*** Vérifie si un objet "contact" existe déjà dans localStorage et affiche ses infos si c'est le cas
- ***Fonction à tester :*** Panier.isAlreadyCustomer (ligne 211 à 220)
- ***Déclencheur :*** Dans la méthode displayForm()
- ***Arguments :*** aucun
- ***Retour attendu :*** Si un"contact" existe, affiche les valeurs pour chaque input
- ***Problème possible :*** n'arrive pas à interroger localStorage

Efface les données du champ de formulaire

- ***Description :*** supprime l'objet "contact" du localStorage et actualise l'affichage du formulaire
- ***Fonction à tester :*** Panier.resetInputs (ligne 225 à 230)
- ***Déclencheur :*** Au clic sur "Ce n'est pas vous ?" au dessus du formulaire
- ***Arguments :*** aucun
- ***Retour attendu :*** efface les données des champs du formulaire
- ***Problème possible :*** les champs reste remplis après avoir cliqué

Affiche le formulaire

- ***Description :*** insère dans la cible un bloc de code HTML
- ***Fonction à tester :*** Panier.displayForm (ligne 235 à 263)
- ***Déclencheur :*** au chargement de la page et dans la méthode resetInputs()
- ***Arguments :*** aucun
- ***Retour attendu :*** Doit afficher le formulaire avec les champs à remplir, leurs labels et le bouton pour "passer commande" qui valide le formulaire
- ***Problème possible :*** N'affiche pas le formulaire

Envoie les infos de commande à l'api

- ***Description :*** crée un objet contact avec les valeurs du formulaire et récupère le contenu du panier pour l'afficher dans une variable. ces deux valeurs sont stringifiées et envoyer à l'API
- ***Fonction à tester :*** Panier.sendForm (ligne 270 à 286)

- ***Déclencheur :*** A la soumission du formulaire grâce à la méthode watchClick()
- ***Arguments :*** aucun
- ***Retour attendu :*** On obtient une chaîne de caractères que l'on peut vérifier en entrant un console.log
- ***Problème possible :*** Envoie le formulaire sans produits dans le panier

Surveille la soumission du formulaire

- ***Description :*** Écoute la soumission du formulaire. Si le panier a au moins 1 produit, le montant total est sauvegardé dans localStorage et applique la méthode sendForm(). Sinon affiche un modal d'avertissement
- ***Fonction à tester :*** Panier.watchClick (ligne 293 à 308)
- ***Déclencheur :*** au chargement de la page ainsi qu'avec la méthode displayTotal()
- ***Arguments :*** un number représentant le total du panier
- ***Retour attendu :*** sauvegarde le total dans localStorage à la soumission du formulaire. Affiche un modal s'il n'y a pas d'article
- ***Problème possible :*** La clé/valeur n'est pas sauvé dans localStorage à la soumission

confirmation.js

Vide le panier et affiche les remerciements

- ***Description :*** Supprime le panier du localStorage et vide le composant panier. Puis exécute la méthode displayMessage()
- ***Fonction à tester :*** Confirmation.clearCart (ligne 12 à 16)
- ***Déclencheur :*** Au chargement de la page

- ***Arguments :*** La cible où afficher le message de remerciements
- ***Retour attendu :*** L'icône panier en haut à droite du site doit revenir à 0 et un message de remerciements doit être affiché
- ***Problème possible :*** Le panier ne se vide pas au chargement de la page

Affiche un message de remerciements

- ***Description :*** Récupère les infos de contact et le montant total dans localStorage. Récupère le numéro de commande dans les paramètres URL et affiche un message de remerciements personnalisé
- ***Fonction à tester :*** Confirmation.displayMessage (ligne 21 à 49)
- ***Déclencheur :*** Au chargement de la page avec la méthode clearCart()
- ***Arguments :*** La cible où afficher le message de remerciements
- ***Retour attendu :*** Un message reprenant le prénom et l'adresse mail de l'utilisateur, le montant total de la transaction ainsi que le numéro de commande retourné par l'API
- ***Problème possible :*** N'arrive pas à récupérer les infos dans localStorage

datamanager.js

Récupère tous les produits de l'API

- ***Description :*** Envoie une requête fetch à l'API et récupère les produits sous forme de tableau
- ***Fonction à tester :*** Datamanager.getAllProducts (ligne 14 à 19)

- ***Déclencheur :*** La méthode getProduct() ainsi que la méthode de la class Home showAllProducts()
- ***Arguments :*** aucun
- ***Retour attendu :*** Doit retourner les produits de l'API sous forme de tableau
- ***Problème possible :*** Ne peut pas récupérer les infos depuis l'API

Récupère un produit unique

- ***Description :*** Récupère tous les produits grâce à la méthode getAllProducts() et applique la méthode extractFromArray()
- ***Fonction à tester :*** DataManager.getProduct (ligne 28 à 31)
- ***Déclencheur :*** La méthode de la class Product showProduct() et la méthode de la class Panier displayCart()
- ***Arguments :*** L'id du produit à récupérer
- ***Retour attendu :*** Le produit sous forme d'objet avec toutes ses propriétés
- ***Problème possible :*** l'id spécifié en paramètre ne correspond pas à l'API

Extrait un produit du tableau

- ***Description :*** Parcourt le tableau contenant les produits et retourne celui correspondant à l'id passé en paramètre
- ***Fonction à tester :*** DataManager.extractFromArray (ligne 40 à 45)

- ***Déclencheur :*** La méthode getProduct()
- ***Arguments :*** L'id du produit à retourner
- ***Retour attendu :*** un objet correspondant au produit spécifié
- ***Problème possible :*** L'id en paramètre ne correspond pas à un id du tableau

Sauvegarde le panier dans localStorage

- ***Description :*** Enregistre le panier sous forme de chaîne de caractères
- ***Fonction à tester :*** DataManager.saveCart (ligne 54 à 56)
- ***Déclencheur :*** Les méthodes add(), remove(), delete() et deleteAll() du composant panier (class Cart)
- ***Arguments :*** Le tableau du panier avec tous les ids des produits ajoutés
- ***Retour attendu :*** enregistre le panier sous forme de chaîne de caractères dans localStorage
- ***Problème possible :*** Ne récupère pas le tableau du composant panier

Recherche si le panier est déjà présent

- ***Description :*** Interroge localStorage pour savoir si un panier est déjà présent
- ***Fonction à tester :*** Datamanager.getCart (ligne 63 à 65)
- ***Déclencheur :*** À l'affichage du composant panier sur la page
- ***Arguments :*** Aucun

- ***Retour attendu :*** Si un panier existe, retourne le panier sous forme de tableau. Sinon, renvoie un tableau vide
- ***Problème possible :*** N'arrive pas à interroger le localStorage

Envoie la commande à l'API et traite la réponse

- ***Description :*** Envoie une requête fetch avec les infos nécessaires. Reçoit la réponse, enregistre les infos de contact retournées dans localStorage et redirige vers la page de confirmation en passant le numéro de commande en paramètre URL
- ***Fonction à tester :*** Datamanager.postOrder (ligne 175 à 93)
- ***Déclencheur :*** lors de l'exécution de la méthode sendForm() de la class Panier
- ***Arguments :*** un string contenant les infos de contact et les produits du panier
- ***Retour attendu :*** une réponse au format json de l'API puis un objet contact sous forme de chaîne de caractères dans localStorage. Enfin, une réorientation vers la page de confirmation
- ***Problème possible :*** Si les infos en paramètre ne sont pas au bon format, le catch renvoie un message d'erreur dans la console

cart.js

Affiche l'icône panier et le nombre de produits qu'il contient

- ***Description :*** Retourne un bloc HTML
- ***Fonction à tester :*** Cart.render (ligne 28 à 35)
- ***Déclencheur :*** au chargement de la page ainsi que dans les méthodes add(), remove(), delete() et deleteAll()
- ***Arguments :*** aucun
- ***Retour attendu :*** Un icône de panier et un chiffre qui s'actualise à chaque changement de valeur
- ***Problème possible :*** L'icône ne s'affiche pas sur la page

Ajoute un ou plusieurs ids au tableau

- ***Description :*** pousse à la fin du tableau l'id en 1er paramètre, le nombre de fois spécifié en 2ème paramètre. Actualise le contenu et enregistre le changement dans localStorage
- ***Fonction à tester :*** Cart.add (ligne 41 à 47)
- ***Déclencheur :*** la méthode addToCart() de la class Product et la méthode increment() de la class Panier
- ***Arguments :*** l'id à ajouter en 1er paramètre. Le nombre d'id à ajouter (1 par défaut)

- ***Retour attendu :*** doit ajouter le/les produit(s) au tableau et actualiser l'affichage du composant
- ***Problème possible :*** l'affichage ne change pas lorsque les déclencheurs sont exécutés

Retire un id du tableau

- ***Description :*** Cherche la première occurrence de l'id dans le tableau et l'enlève, puis actualise l'affichage et sauvegarde le tableau dans localStorage
- ***Fonction à tester :*** Cart.remove (ligne 53 à 58)
- ***Déclencheur :*** La méthode decrement() dans la class Panier
- ***Arguments :*** L'id à retirer
- ***Retour attendu :*** doit retirer un produit du tableau et actualiser son affichage
- ***Problème possible :*** L'affichage du composant ne change pas lorsque l'on retire un produit sur la page panier

Supprime toutes les occurrences de l'id dans le tableau

- ***Description :*** Filtre l'id spécifié en ajoutant dans un tableau seulement les ids ne correspondant pas. Enfin, actualise l'affichage et sauvegarde dans localStorage
- ***Fonction à tester :*** Cart.delete (ligne 64 à 72)
- ***Déclencheur :*** La méthode deleteLine() de la Class Panier

- ***Arguments :*** L'id à filtrer
- ***Retour attendu :*** Un tableau ne contenant plus l'id spécifié. L'affichage du chiffre sur le composant panier doit s'actualiser
- ***Problème possible :*** L'id reste présent dans le tableau lorsque le déclencheur est exécuté

Supprime tout le contenu du tableau

- ***Description :*** Assigne un tableau vide au composant panier. Ensuite, actualise l'affichage et sauvegarde dans localStorage
- ***Fonction à tester :*** Cart.deleteAll (ligne 78 à 82)
- ***Déclencheur :*** La méthode clearCart() de la Class Confirmation
- ***Arguments :*** aucun
- ***Retour attendu :*** Un tableau vide. L'affichage du chiffre sur le composant panier doit être 0
- ***Problème possible :*** Le ne se vide pas lorsque le déclencheur est exécuté

main.js

Définit quel script exécuté sur la page courante

- ***Description :*** Récupère l'URL de la page actuelle, la compare aux différents cas et retourne l'instance de la class appropriée
- ***Fonction à tester :*** definePage (ligne 12 à 26)

- ***Déclencheur :*** Au chargement de la page
- ***Arguments :*** Aucun
- ***Retour attendu :*** doit exécuter l'instance de class

  correspondant à la page HTML concernée
- ***Problème possible :*** N'exécute pas la bonne instance sur la page actuelle