DOCUMENTATION

**Html**

Classe du titre : Nom-entreprise

Id conteneur ajout : zone-saisie

Id texte dépôt id : champ-id

Id texte dépôt fruit : champ-fruit

Id texte dépôt quantité : champ-quantité

Id dépôt bouton : bouton-valider

Id conteneur détails (none) : zone-detail

Id sortie nom : sortie-nom

Id sortie quantité : sortie-Qte

Id sélecteur : selecteur-fruit

Id sortie résultat : resultat-recherche

Id conteneur retrait : zone-retrait

Id texte retrait id : champ-id

Id texte retrait fruit : champ-fruit-retrait

Id texte retrait quantité : champ-quantite-retrait

Id retrait bouton : bonton-retrait

This : appeler la classe lui-même (fruit)

DOCUMENTATION

**JavaScript**

**l.1-40**

Au click du bouton (#bouton-valider) du conteneur d’entrée de dépôt

Déclaration de variable : **l.1-7**

‘’id pour value de id champ-id

‘’nom’’ pour value de id champ-fruit

‘’qt’’ pour value de id champ-quantité

Création de donnée de la fonction (addfruit) **l.9**

id pour id

nom pour nom

quantité pour qt–

Ajout de texte sur la zone id ‘’sortie-Nom’’ retour de valeur ‘’Nom’’ : $nom  **l.14-16**

Ajout de texte sur la zone id ‘’sortie-Qte’’ retour de valeur ‘’Nom :’’ $quantité **l.19-21**

Déclaration de variable selectElement qui correspond à l’id selecteur-fruit **l.24**

selectElement s’ajoute ‘’ ’’ **l.25**

Déclaration de variable : **L28**

‘’storedNames’’ pour transformation chaine en objet et que localstorage reçoit l’objet ‘’fruit’’ (JSON.parse)

Condition si storedNames et storedNames.length (longueur) > 0 **l.31-40**

Recherche dans storedNames pour chaque element ‘’name’’ **l.33-40**

‘’option’’ pour création d’élément ‘’option’’

Déclaration de variable x pour id selecteur-fruit

Création d’élément option avec ajout de texte ‘’nom :’’ $name.nom et ‘’quantité :’’ $name.quantité

Sur x création d’un nœud enfant ‘’option’’

Au changement du sélecteur **l.43-50**

Déclaration de variable ‘’str’’ texte vide

Recherche pour chaque option du sélectionner du sélecteur **l.49-51**

str s’ajoute ce texte + ‘’\_’’

#resultat-recherche prend le texte str

S’exécute pour tout changement du sélecteur

Au click du bouton (#bouton-retrait) se passe   **l.52-75**

Declaration de variable

qt\_rt pour la value de id #champ-quantite-retrait

id pour la value de id #champ-id-retrait

console log pour vérifier id

Création de donnée de la fonction (retireStock)  **l.56**

Id pour id et qt\_rt

Déclaration de variable selectElement qui correspond à l’id selecteur-fruit **l.59**

selectElement s’ajoute ‘’ ’’ **l.60**

Déclaration de variable : **L63**

‘’storedNames’’ pour transformation chaine en objet et que localstorage reçoit l’objet ‘’fruit’’ (JSON.parse)

Condition si storedNames et storedNames.length (longueur) > 0 **l.66-75**

Recherche dans storedNames pour chaque element ‘’name’’ **l.68-75**

‘’option’’ pour création d’élément ‘’option’’

Déclaration de variable x pour id selecteur-fruit

Création d’élément option avec ajout de texte ‘’nom :’’ $name.nom et ‘’quantité :’’ $name.quantité

Sur x création d’un nœud enfant ‘’option’’

Fonction saveFruit avec paramètre ‘’fruit’’ **l.84**

Qui dans le localStorage prend la valeur de fruit transformer en chaine de caractère **l.89**

Fonction getFruit **l.93-101**

Avec une variable fruit qui est le localStorage qui reçoit le jeux de clé ‘’fruit’’

Si ‘’fruit’’ est null (nexiste pas) retour un tableau vide

Sinon retour une transformation de chaine charactere en objet pour la clé ‘’fruit’’ (recuperer ses valeur)

Fonction addFruit en parametre ‘’liste et qt’’ **l.105-120**

Déclaration de variable ‘’fruit’’ qui est la fonction getFruit

Déclaration de variable rechFruit qui prend comme valeur le jeux de clé (fruit) /valeur, puis fais une recherche par méthode find() qui renvoie la valeur du 1er élément trouver qui respecte la condition donner par rapport a la fonction de teste de passer en argument. Si dans le cas où la méthode retourne undifined, c’est que la valeur comparer n’a pas été trouver **l.109**

Si rechFruit est different de undefined (donc valeur trouvé) donc ça va ajouter la nouvelle quantité en argument de la fonction addfruit() dans la quantité déjà existante dans le jeu de clé(fruit)/valeur **l.110-111**

Sinon si rechfruit retourne undefined cest que la valeur n’existe pas dans le jeu de clé(fruit)/valeur alors on ajoute avec la méthode push(), le nouvelle enregistrement dans le jeu clé(fruit)/valeur **l.114**

Sauvegarde des modification sur la clé(fruit)/valeur avec la fonction saveFruit  **l.117-119**

Renvoie les valeur de la clé(fruit)/valeur

Fonction retireStock en parametre ‘’liste et qt’’ **l.122-128**

Déclaration de variable ‘’fruit’’ qui est la fonction getFruit

Déclaration de variable rechFruit qui prend comme valeur le jeux de clé (fruit) /valeur, puis fais une recherche par méthode find() qui renvoie la valeur du 1er élément trouver qui respecte la condition donner par rapport a la fonction de teste de passer en argument. Si dans le cas où la méthode retourne undifined, c’est que la valeur comparer n’a pas été trouver **l.124**

Si rechFruit est different de undefined (donc valeur trouvé) donc ça va ajouter la nouvelle quantité en argument de la fonction addfruit() dans la quantité déjà existante dans le jeu de clé(fruit)/valeur **l.125-126**

Sinon si rechfruit retourne undefined cest que la valeur n’existe pas dans le jeu de clé(fruit)/valeur alors on ajoute avec la méthode push(), le nouvelle enregistrement dans le jeu clé(fruit)/valeur

Sauvegarde des modification sur la clé(fruit)/valeur avec la fonction saveFruit  **l.128**

Renvoie les valeur de la clé(fruit)/valeur