

\ (namespaces/default.html)

\

NAMESPACE HIERARCHY
 ▶ \

# Functions

## addArticle()

addArticle(string \$Titre, float \$Prix, integer \$Cerise, integer \$CeriseBonus, boole  
 an \$Available, string \$LinkImg) : void

</>

Ajoute un article dans la boutique depuis le pannel admin.

### Parameters

string	\$Titre	(Ex: "Caisse de Cerise")
float	\$Prix	(Ex: 13.33)
integer	\$Cerise	(Ex: 1000)
integer	\$CeriseBonus	(Ex: 250)
boolean	\$Available	(Ex: true)
string	\$LinkImg	(Ex: "cerise-caisse.png")

FILE
 functions.php
 (files/functions.html)

### Tags

package Default

# addCerise()

```
addCerise(integer $nombreCerise, integer $idUser) : void
```

Ajoute des cerises sur le comptes d'un utilisateur

## Parameters

integer	\$nombreCerise	(Ex: 100)
integer	\$idUser	(Ex: 11)

**FILE**  
functions.php  
(files/functions.html)

## Tags

package Default

# addFiche()

```
addFiche(string $Filière, string $Correspondance, string $Titre, string $Text1, string $Text2, string $Text3, string $Text4, string $Text5, integer $IdUser) : void
```

Envoyer une fiche depuis un formulaire, permet d'alimenté le site, et de récompensé les utilisateurs

## Parameters

string	\$Filière	(Ex: "bac")
string	\$Correspondance	(Ex: "1.1.1")
string	\$Titre	(Ex: "illo nulla qui")
string	\$Text1	(Ex: "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Cupiditate aliquam iure assumenda beatae maxime ab inventore explicabo illo nulla qui? Maiores dicta consequatur tempore hic error similique voluptatibus nulla magni.")
string	\$Text2	(Ex: "consectetur adipisicing elit. Cupiditate aliquam iure assumenda beatae maxime ab inventore explicabo illo nulla qui? Maiores dicta consequatur tempore hic error similique voluptatibus nulla magni.")
string	\$Text3	(Ex: "Cupiditate aliquam iure assumenda beatae maxime ab inventore explicabo illo nulla qui? Maiores dicta consequatur tempore hic error similique voluptatibus nulla magni.")
string	\$Text4	(Ex: "assumenda beatae maxime ab inventore explicabo illo nulla qui? Maiores dicta consequatur tempore hic error similique voluptatibus nulla magni.")

string	\$Text5	(Ex: "sit amet consectetur adipisicing elit. Cupiditate aliquam iure assumenda beatae maxime ab inventore explicabo illo nulla qui? Maiores dicta consequatur tempore hic error similique voluptatibus nulla magni.")
integer	\$idUser	Identifiant de la session (Ex: 11)

**FILE**

functions.php  
(files/functions.html)

Tags

package	Default
---------	---------

## addFicheForUser()

```
addFicheForUser(integer $idUser, integer $IdFiche, integer $price) : void
```

</>

*Donnée l'accès à une fiche pour l'utilisateur, si il a suffisamment de cerise pour la débloquer.*

Parameters

integer	\$idUser	(Ex: 11)
integer	\$IdFiche	(Ex: 2)
integer	\$price	(Ex: 300)

**FILE**

functions.php  
(files/functions.html)

Tags

package	Default
---------	---------

## addLikeForFiche()

```
addLikeForFiche(integer $idUser, integer $IdFiche) : void
```

</>

*Ajoute la mention j'aime à une fiche*

Parameters

integer	\$idUser	(Ex: 11)
---------	----------	----------

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# addToken()

addToken(string \$email, integer \$date, string \$idToken) : void

</>

Créer un jeton de validation envoyé par mail pour confirmer le compte.

## Parameters

string	\$email	(Ex: "test@test.fr")
integer	\$date	(Ex: 1111002983)
string	\$idToken	(Ex: "p2z03iLp6ff7z2")

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# addUser()

addUser(string \$nom, string \$prénom, string \$email, string \$password, string \$avatar) : void

</>

Ajoute un nouveau utilisateur avec les données envoyé depuis le formulaire

## Parameters

string	\$nom	(Ex: "Dupon")
string	\$prénom	(Ex: "Pierre")
string	\$email	(Ex: "test@test.fr")

string	\$password	(Ex: "SHAfpz3iLp6@dffz2")
string	\$avatar	(Ex: "1292837647291083")

**FILE**  
functions.php  
(files/functions.html)

Tags

package Default

# addVente()

```
addVente(string $IdTransaction, string $IdPaypalPayer, integer $IdUser, string $Nom, string $Prenom, integer $IdArticle, string $Status, float $Prix, string $Devise, string $Pays, integer $Cerise, string $NumeroFacture, float $TauxTVA, string $Adresse) : void
```



Ajoute une vente, cette fonction permet de fabriquer les factures client.

## Parameters

string	\$IdTransaction	(Ex: "229842Q7918ZF53")
string	\$IdPaypalPayer	(Ex: "02093821")
integer	\$IdUser	(Ex: 11)
string	\$Nom	(Ex: "Dupon")
string	\$Prenom	(Ex: "Pierre")
integer	\$IdArticle	(Ex: 2)
string	\$Status	(Ex: "valid")
float	\$Prix	(Ex: 23.5)
string	\$Devise	(Ex: "EUR")
string	\$Pays	(Ex: "FR")
integer	\$Cerise	(Ex: 200)
string	\$NumeroFacture	(Ex: "DCP292873829827492")
float	\$TauxTVA	(Ex: 5.5)
string	\$Adresse	(Ex: "7 rue des fours, 11000 Carcassonne")

**FILE**  
functions.php  
(files/functions.html)

Tags

**package** Default

# checkPassword()

checkPassword(string \$email) : void

</>

Récupère le mot de passe en fonction de l'adresse mail envoyé.

## Parameters

string \$email (Ex: "test@test.fr")

**FILE**  
functions.php  
(files/functions.html)

Tags

**package** Default

# connexion\_db()

connexion\_db()

</>

**FILE**  
essentialFunction.php  
(files/essentialFunction.html)

Tags

**package** Default

# countSpecialChar()

countSpecialChar( \$chain)

</>

## Parameters

\$chain

### FILE

essentialFunction.php  
(files/essentialFunction.html)

### Tags

**package** Default

## cutPhrase()

```
cutPhrase( $string, $limit = 5, $fin = ' ...')
```

</>

## Parameters

\$string

\$limit

\$fin

### FILE

essentialFunction.php  
(files/essentialFunction.html)

### Tags

**package** Default

## dataUser()

```
dataUser(integer $id) : array
```

</>

*Selectionne toute les données d'un utilisateur avec son identifiant de session.*

## Parameters

integer \$id      identifiant de session (Ex: 34)

## Returns

array

### FILE

functions.php

Tags

package	Default
---------	---------

## date\_outil()

```
date_outil( $date, $nombre_jour)
```

</>

Parameters

\$date
\$nombre_jour

FILE

essentialFunction.php  
(files/essentialFunction.html)

Tags

package	Default
---------	---------

## deleteAccount()

```
deleteAccount(integer $IdUser) : void
```

</>

Supprime le compte de l'utilisateur, avec les fiches débloquent et les mentions j'aime.

Parameters

integer	\$IdUser	(Ex: 11)
---------	----------	----------

FILE

functions.php  
(files/functions.html)

Tags

package	Default
---------	---------



# deleteFiche()

```
deleteFiche(integer $id) : void
```

</>

Supprime une fiche en fonction de son identifiant

## Parameters

integer	\$id	identifiant de la fiche (Ex: 12)
---------	------	----------------------------------

**FILE**  
functions.php  
(files/functions.html)

## Tags

package	Default
---------	---------

# editArticle()

```
editArticle(string $Titre, float $Prix, integer $Cerise, integer $CeriseBonus, boolean $Available, $IdProduit) : void
```

</>

Modifie un article dans la boutique depuis le pannel admin

## Parameters

string	\$Titre	(Ex: "Caisse de Cerise")
float	\$Prix	(Ex: 13.33)
integer	\$Cerise	(Ex: 1000)
integer	\$CeriseBonus	(Ex: 250)
boolean	\$Available	(Ex: true)
	\$IdProduit	

**FILE**  
functions.php  
(files/functions.html)

## Tags

package	Default
---------	---------

# emailDoublon()

---

emailDoublon(string \$email) : array

</>

Vérifier si il n'y a pas d'adresse mail doublon

## Parameters

string \$email (Ex: "test@test.fr")

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

## Tags

package Default

# getIp()

---

getIp()

</>

**FILE**  
essentialFunction.php  
(files/essentialFunction.html)

## Tags

package Default

# getUrl()

---

getUrl( \$chemin = true)

</>

## Parameters

\$chemin

**FILE**  
essentialFunction.php  
(files/essentialFunction.html)

Tags

package

Default

# isAdmin()

isAdmin() : void

</>

Verifier si l'utilisateur est admin

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# isBan()

isBan() : void

</>

Verifier si l'utilisateur est ban de la plateform

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# isConnect()

isConnect()

</>

**FILE**  
essentialFunction.php  
(files/essentialFunction.html)

# likeFiche()

likeFiche(integer \$IdUser) : array

</>

Récupère tout les fiches avec la mention j'aime donnée par l'utilisateur pour la partie donnée personnelle.

## Parameters

integer

\$IdUser

(Ex: 11)

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

# likeNumberFiche()

likeNumberFiche(integer \$IdFiche) : integer

</>

Récupère le nombre total de mention j'aime de la fiche

## Parameters

integer

\$IdFiche

(Ex: 45)

## Returns

integer —  
443

**FILE**  
functions.php  
(files/functions.html)

Tags

package	Default
---------	---------

# MyDate()

MyDate( \$date)

</>

Parameters

\$date

FILE

essentialFunction.php  
(files/essentialFunction.html)

Tags

package	Default
---------	---------

# recupAllArticles()

recupAllArticles() : array

</>

Récupérer tout les articles de la boutique publique. (Certain article peuvent être masqué depuis l'espace Admin)

Returns

array

FILE

functions.php  
(files/functions.html)

Tags

package	Default
---------	---------

# recupAllArticlesForArmin()

recupAllArticlesForArmin() : array

</>

Récupérer tout les articles de la boutique pour le menu admin

## Returns

array

**FILE**

functions.php  
(files/functions.html)

## Tags

**package** Default

# recupAllFactures()

recupAllFactures() : array

</>

*Récupérer toute les factures (pour les admins)*

## Returns

array

**FILE**

functions.php  
(files/functions.html)

## Tags

**package** Default

# recupAllFacturesUser()

recupAllFacturesUser(integer \$IdUser) : array

</>

*Récupérer toute les factures d'un utilisateur*

## Parameters

integer \$IdUser (Ex: 11)

## Returns

array

**FILE**

functions.php  
(files/functions.html)

Tags

package

Default

# recupAllFichesOfUser()

recupAllFichesOfUser(integer \$IdUser, boolean \$public) : void

</>

Récupère tout les fiches valider ou non de l'utilisateur

## Parameters

integer	\$IdUser	(Ex: 11)
boolean	\$public	(Ex: true)

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# recupAllFichesUnlockOfUser()

recupAllFichesUnlockOfUser(integer \$IdUser) : array

</>

Récupère toute les fiches débloquée par l'utilsateur

## Parameters

integer	\$IdUser	(Ex: 11)
---------	----------	----------

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# recupAllUserForAdmin()

recupAllUserForAdmin() : array

</>

Récupère tout les utilisateurs du site avec leur dépenses totale en euro.

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

## Tags

package Default

# recupDataArticles()

recupDataArticles(integer \$IdProduit) : array

</>

Récupérer les données d'un article si il est visible (cela est pour la boutique client)

## Parameters

integer \$IdProduit (Ex: 4)

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

## Tags

package Default

# recupDataArticlesAdmin()

recupDataArticlesAdmin(integer \$IdProduit) : array

</>

Récupérer les données d'un article si il est visible ou non (cela est pour la vu admin)



## Parameters

integer                      \$IdProduit                      (Ex: 4)

## Returns

array

### FILE

functions.php  
(files/functions.html)

## Tags

**package**                      Default

## recupFacture()

---

```
recupFacture(string $NumeroFacture) : array
```

```
</>
```

*Récupérer les données de la facture avec son numéro.*

## Parameters

string                      \$NumeroFacture                      (Ex: "DCP299278329042")

## Returns

array

### FILE

functions.php  
(files/functions.html)

## Tags

**package**                      Default

## recupId()

---

```
recupId(string $email) : array
```

```
</>
```

*Récupérer l'identifiant associé à l'adresse email*

## Parameters

string                      \$email                      (Ex: "test@test.fr")

# Returns

array

**FILE**

functions.php  
(files/functions.html)

Tags

**package** Default

## recupLastIdFiche()

recupLastIdFiche() : integer

</>

Récupère le dernière identifiant insérer en base de donnée de la table fiche

# Returns

integer —  
34

**FILE**

functions.php  
(files/functions.html)

Tags

**package** Default

## recupPageView()

recupPageView(integer \$idUser) : array

</>

Récupère tout les pages visités de l'utilisateur pour la partie donnée personnelle.

# Parameters

integer \$idUser (Ex: 11)

# Returns

array

**FILE**

functions.php  
(files/functions.html)

Tags

package

Default

# recupRecetteMois()

recupRecetteMois(integer \$mois, integer \$annee) : float

</>

Récupère les recettes d'un mois choisi

## Parameters

integer	\$mois	(Ex: 12)
integer	\$annee	(Ex: 2022)

## Returns

float —  
234.45

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# recupToken()

recupToken(mixed \$id) : array

</>

Récupère les informations du token.

## Parameters

mixed	\$id	id du token (Ex: "p2z03iLp6ff7z2")
-------	------	------------------------------------

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# recupVenteMois()

recupVenteMois(integer \$mois, integer \$annee) : integer

</>

Récupère le nomnbre de vente d'un mois choisi

## Parameters

integer	\$mois	(Ex: 12)
integer	\$annee	(Ex: 2022)

## Returns

integer —  
2

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# recupViewPage()

recupViewPage(integer \$mois, integer \$annee) : integer

</>

Récupérer le nombre de visite du mois

## Parameters

integer	\$mois	(Ex: 12)
integer	\$annee	(Ex: 2022)

## Returns

integer —  
234

**FILE**

Tags

package

Default

# redirectUrl()

redirectUrl( **\$url**)

</>

Parameters

\$url

**FILE**  
essentialFunction.php  
(files/essentialFunction.html)

Tags

package

Default

# selectAllFiche()

selectAllFiche(boolean **\$public**, string **\$filiere** = NULL, integer **\$pole** = NULL) : integer

</>

*compte toute les fiches publique ou en attente en fonction de la filière et du pole*

Parameters

boolean	<b>\$public</b>	publique (Ex: true)
string	<b>\$filiere</b>	filière (Ex: "bac")
integer	<b>\$pole</b>	pole (Ex: 1)

Returns

integer —  
43

**FILE**  
functions.php  
(files/functions.html)

# selectAllFiches()

```
selectAllFiches(string $filiere, integer $pole, string $classement, integer $fiches
LimiteParPage, integer $ficheDeDepart, boolean $IsCount = false) : array
```

</>

Selectionne tout les fiche avec un filtrage, ou si \$IsCount est sur true compte le nombre de fiche avec les filtres.

## Parameters

string	\$filiere
integer	\$pole
string	\$classement
integer	\$fichesLimiteParPage
integer	\$ficheDeDepart
boolean	\$IsCount

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

# selectFicheOfId()

```
selectFicheOfId(integer $IdFiche) : array
```

</>

Selectionne les données d'une fiche en fonction d'un identifiant.

## Parameters

integer	\$IdFiche	identifiant de la fiche (Ex: 12)
---------	-----------	----------------------------------

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

Tags

package Default

# selectLastFicheSend()

selectLastFicheSend() : array

</>

Permet de selectionner la plus veille fiche non publique

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

Tags

package Default

# unlockedFiche()

unlockedFiche(integer \$IdUser) : array

</>

Récupère tout les fiches débloquées de l'utilisateur pour la partie donnée personnelle.

## Parameters

integer \$IdUser (Ex: 11)

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

# unlockedNumberFiche()

unlockedNumberFiche(integer \$IdFiche) : integer

</>

Récupère le nombre de fois que la fiche à été débloquent

## Parameters

integer

\$IdFiche

(Ex: 11)

## Returns

integer —  
7

**FILE**  
functions.php  
(files/functions.html)

# updateExpireToken()

updateExpireToken(integer \$date, string \$id, string \$newIdT) : void

</>

Change le token et la date si expiré.

## Parameters

integer	\$date	(Ex: 1111002983)
string	\$id	identifiant du token (Ex: "p2z03iLp6ff7z2")
string	\$newIdT	identifiant du nouveau token (Ex: "md7pc5lsoz2uf")

**FILE**  
functions.php  
(files/functions.html)



Tags

package

Default

# updateFiche()

updateFiche(integer \$id, integer \$quality) : void

</>

Passe une fiche en format publique avec la qualité choisi.

Parameters

integer	\$id	identifiant de la fiche (Ex: 12)
integer	\$quality	qualité de la fiche (Ex: 0,1 ou 2)

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# validAccount()

validAccount( \$idUser) : void

</>

valide le compte de l'utilisateur depuis le pannel admin.

Parameters

\$idUser

**FILE**  
functions.php  
(files/functions.html)

Tags

package

Default

# validAccountToken()

```
validAccountToken(string $email) : void
```

</>

Confirme le compte si le token est validé

## Parameters

string	\$email	(Ex: "test@test.fr")
--------	---------	----------------------

**FILE**  
functions.php  
(files/functions.html)

## Tags

package	Default
---------	---------

# verifFicheUnlocked()

```
verifFicheUnlocked(integer $idUser, integer $idFiche) : array
```

</>

Vérifie si la fiche est débloquent pour l'utilisateur

## Parameters

integer	\$idUser	(Ex: 11)
integer	\$idFiche	(Ex : 3)

## Returns

array

**FILE**  
functions.php  
(files/functions.html)

## Tags

package	Default
---------	---------

# verifInt()

```
verifInt( $chain)
```

</>

## Parameters

\$chain

### FILE

essentialFunction.php  
(files/essentialFunction.html)

### Tags

package Default

## verifLikeFiche()

```
verifLikeFiche(mixed $idUser, mixed $idFiche) : array
```

</>

*Vérifie si la fiche est débloquent pour l'utilisateur*

## Parameters

mixed	\$idUser	(Ex: 11)
mixed	\$idFiche	(Ex: 45)

## Returns

array

### FILE

functions.php  
(files/functions.html)

### Tags

package Default

## verifValidUser()

```
verifValidUser(string $email) : array
```

</>

*Vérifier si le compte de l'utilisateur a été validé par mail.*

## Parameters

string	\$email	(Ex: "test@test.fr")
--------	---------	----------------------

## Returns

array

FILE

functions.php  
(files/functions.html)

Tags

package Default

# viewNumberFiche()

```
viewNumberFiche(integer $IdFiche) : integer
```

</>

Récupère le nombre de fois que la fiche à été vu

## Parameters

integer \$IdFiche (Ex: 11)

## Returns

integer —  
23

FILE

functions.php  
(files/functions.html)

Tags

package Default

# viewPage()

```
viewPage(string $NamePage) : void
```

</>

Ajoute un vu sur la page pour faire des statistiques. cette fonction évite les doublons.

## Parameters

string \$NamePage Nom de la page ("accueil")

FILE

functions.php  
(files/functions.html)

## Tags

package

Default

# writeCorrespondance()

```
writeCorrespondance(string $arg) : string
```

```
</>
```

*Permet de convertir le code correspondance en correspondance de toute lettre.*

## Parameters

string      \$arg      correspondance (Ex: "1.1.1")

## Returns

string —

"1.1.1 Tenue des dossiers fournisseurs et sous-traitants"

### FILE

functions.php  
(files/functions.html)

## Tags

package

Default

# writeQuality()

```
writeQuality(integer $quality) : string
```

```
</>
```

*Permet de convertir le chiffre de la qualité en toute lettre.*

## Parameters

integer      \$quality      qualité (Ex: 0)

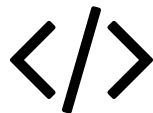
## Returns

string —

"classique"

### FILE

functions.php  
(files/functions.html)



Class Hierarchy Diagram (<graphs/class.html>)



Errors (<reports/errors.html>)

Markers (<reports/markers.html>)