



I1 - ISCC

I-SCC-000

ISCC Day 06

PHP 02



SOMMAIRE

- SOMMAIRE
- JOUR 06
- EXERCICE 00 - DOSSIER DE TRAVAIL
- EXERCICE 01 - CREATION DE FONCTION
 - FONCTIONS AFFICHAGE
 - FONCTION MANIPULATION
- EXERCICE 02 - UTILISATION FONCTIONS
 - MANIPULATION DE CHAINE
- EXERCICE 03 - BOUCLES
 - BOUCLE SIMPLE
 - BOUCLE SUITE
 - BOUCLES IMBRIQUEES
- EXERCICE 04 - PROJET



JOUR 06

Aujourd'hui vous allez continuer d'expérimenter les différentes mécaniques du PHP.
Après les variables et les conditions, vous allez voir les fonctions et les boucles.

EXERCICE 00 - DOSSIER DE TRAVAIL

Si vous n'avez pas préparé le repos de travail "ISCC-2020-J06" pendant le jour 01.

- Créez un nouveau repos "ISCC-2020-J06" dans votre dossier "ISCC-2020".
- Publiez le repos "ISCC-2020-J06" sur GitHub.
- **Invitez** les membres de votre équipe pédagogique de l'ISCC à collaborer avec vous sur votre repos "ISCC-2020-J06".

C'est dans ce dossier, que vous publierez vos travaux de cette journée. N'oubliez pas d'envoyer régulièrement sur GitHub votre travail. Ceci pourra vous servir de sauvegarde en cas d'erreur ou de perte de données.

EXERCICE 01 - CREATION DE FONCTION

Cette série d'exercice va vous permettre de prendre en main les fonctions en PHP.
Vous allez voir la déclaration et l'utilisation d'une fonction ainsi que les paramètres et les valeurs de retour.

- Créez un dossier EX_01 où vous travaillerez pour les sous-exercices suivants.
- Créez un fichier "test-fonctions.php" qui vous servira à tester les fonctions que vous allez créer.
- Dans le fichier "test-fonctions.php" déclarez les variables suivantes et assignez les valeurs correspondantes.

nom-produit	T-shirt femme
couleur	Rouge
prix	15.50
disponible	true
quantite	10

FONCTIONS AFFICHAGE

- Dans le dossier “EX_01” Créez un fichier “affichage.php”
- Importez le fichier “affichage.php” dans le fichier “test-fonctions.php”

Dans le fichier “affichage.php”, créez les trois fonctions suivantes :

- fonction : afficher-titre
paramètres: nom-produit
Valeur de retour : Rien
Action: Affiche “Fiche produit : X” dans une balise <h1>, X étant le nom du produit passé en paramètre.
- fonction : afficher-description
paramètres: couleur, prix
Valeur de retour : Rien
Action: Affiche “Couleur: X, Prix: Y” dans une balise <p>. X étant la couleur passée en paramètre, Y étant le prix passé en paramètre.
- fonction : afficher-produit
paramètres: nom-produit, couleur, prix, disponible
Valeur de retour : Rien
Action: Si le paramètre disponible est égale à True, appelez les fonctions afficher-titre et afficher-description à la suite sinon affiche “Le produit X n'est malheureusement plus disponible” dans une balise <p>. X étant le titre produit passé en paramètre.

Dans le fichier “test-fonctions.php”, à la suite des déclarations de variables :

* appelez la fonction “afficher-produit” en lui fournissant les paramètres appropriés.



N'hésitez pas à changer les valeurs des paramètres pour tester le fonctionnement de vos fonctions, faites simplement attention à bien remettre les valeurs demandées avant de rendre votre travail.

FONCTION MANIPULATION

- Dans le dossier “EX_01” Créez un fichier “gestion-produit.php”
- Importez le fichier “gestion-produit.php” dans le fichier “test-fonctions.php”

Dans le fichier “gestion-produit.php”, créez les trois fonctions suivantes:

- fonction : update-dispo
paramètres: quantite
Valeur de retour : True/False
Action: Si la quantité passée en paramètre est supérieure à 0 la fonction renvoie True sinon elle renvoie False
- fonction : restockage
paramètres: quantite, nb-ajout
Valeur de retour : nombre (quantité précédente de produit plus l'ajout de nouveaux produits)
Action: Affiche “X produits ont été ajoutés au stock” dans une balise <p> X étant le nombre de produit ajouté. Renvoiez la quantité précédente de produit plus l'ajout de nouveaux produits.

- fonction : achat
paramètres: quantité, nb-achat
Valeur de retour : nombre (quantité précédente de produit moins le nombre de produit acheté)
Action: Affiche “X produits ont été ajoutés acheté” dans une balise <p> X étant le nombre de produit acheté. Renvoie la quantité précédente de produit moins le nombre de produit acheté.

Dans le fichier “test-fonctions.php” et à la suite des appels de fonction précédentes, utilisez les fonctions que vous venez de créer pour :

- Afficher le produit.
- Acheter 5 produits
- Mettre à jour la disponibilité
- Afficher le produit.
- Acheter 5 produits
- Mettre à jour la disponibilité
- Afficher le produit.
- Restocker 5 produits
- Mettre à jour la disponibilité
- Afficher le produits

EXERCICE 02 - UTILISATION FONCTIONS

Vous allez maintenant voir que de nombreuses fonctions existent déjà dans le langage PHP. Que ce soit des opérations mathématiques complexes, des traitements de chaînes de caractères, des manipulations de données comme des dates. Ces fonctions vous permettront de simplifier votre travail dans beaucoup de situation.

Il est donc essentiel quand vous utilisez le PHP d'apprendre à chercher si une fonction PHP n'existe pas déjà pour vous simplifier le travail.

- Créez un dossier EX_02 où vous travaillerez pour les sous-exercices suivants.



Pour réaliser les exercices suivants vous n'aurez pas besoin de créer de fonction. Votre travail est de trouver quelle fonction PHP utiliser pour répondre à la consigne, lire la documentation de ces fonctions et les utiliser correctement pour répondre à l'exercice

MANIPULATION DE CHAÎNE

- Dans le dossier "EX_02", créez un fichier "chaîne.php"
- Déclarez une variable **str1** qui a pour valeur "La maîtrise des fondamentaux du code et de la programmation dans les domaines du business, du marketing et de la communication est un avantage sérieux et utile."

A partir de cette variable et de fonction PHP vous devez maintenant :

- Affichez dans une balise <p>. "La chaîne **str1** contient X caractère" ou X est le nombre de caractère contenu dans la variable **str1**.
- Affichez dans une balise <p>. "La chaîne **str1** contient X mots" ou X est le nombre de mots contenu dans la variable **str1**.
- Affichez dans une balise <p>. La chaîne **str1** mais entièrement en minuscule.
- Affichez dans une balise <p>. La chaîne **str1** mais entièrement en majuscule.
- Affichez dans une balise <p>. La chaîne **str1** mais entièrement mélangé.
- Affichez dans une balise <p>. La chaîne **str1** mais en remplaçant tout les " la " par des " LA ".
- Affichez dans une balise <p>. La chaîne **str1** mais en supprimant la fin " est un avantage sérieux et utile."

EXERCICE 03 - BOUCLES

Troisième notion pour cette journée, les boucles. Ces **structures de contrôle** vont vous permettre de simplifier votre travail encore une fois. Le but d'une boucle est simple : répéter X fois une instruction mais comme pour les conditions les applications sont extrêmement nombreuses. Les conditions et les boucles sont des éléments que vous pourrez retrouver dans de nombreux langage (à quelques exceptions prêt) et sont donc la base même de la logique de programmation.

- Créez un dossier EX_03 où vous travaillerez pour les sous-exercices suivants.

BOUCLE SIMPLE

- Dans le dossier "EX_03", créez un fichier "boucles-simples.php"
- Créez une fonction "compter"
Cette fonction prend en paramètre une variable "maximum".
Cette fonction ne renvoie rien.
Cette fonction doit afficher chaque chiffre, de 1 à la valeur de "maximum", suivi d'un retour à la ligne.
Vous devez utiliser une boucle **while** dans la fonction "compter"
- Créez une fonction "compter-for"
Cette fonction prend en paramètre une variable "maximum".
Cette fonction ne renvoie rien.
Cette fonction doit afficher chaque chiffre, de 1 à la valeur de "maximum", suivi d'un retour à la ligne.
Vous devez utiliser une boucle **for** dans la fonction "compter-for"
- Appelez la fonction "compter" avec 10 en paramètre.
- Appelez la fonction "compter-for" avec 10 en paramètre.

BOUCLE SUITE

- Dans le dossier "EX_03", créez un fichier "boucles-suite.php"
- Créez une fonction "compter-dizaine"
Cette fonction prend en paramètre une variable "maximum".
Cette fonction ne renvoie rien.
Cette fonction doit afficher chaque chiffre, de 1 à la valeur de "maximum", suivi d'un retour à la ligne.
Tous les multiples de 10, la fonction doit afficher "Ceci est une dizaine." suivi d'un retour à la ligne.
Tous les multiples de 100, la fonction doit afficher "Ceci est une centaine." suivi d'un retour à la ligne, mais pas la phrase précédente.
- Appelez la fonction "compter-dizaine" avec 100 en paramètre.



Utilisez le modulo pour les multiples de 10 et de 100.

Résultat attendu pour \$maximum = 100 :

```
Terminal
~/I-SCC-000>
1
2
3
...
9
10
Ceci est une dizaine.
11
12
...
99
100
Ceci est une centaine.
```

BOUCLES IMBRIQUEES

- Dans le dossier “EX_03”, créez un fichier “boucles-imbriquees.php”
- Créez une fonction “clock”
Cette fonction ne prend pas de paramètres et ne renvoie rien.
Cette fonction doit simuler le défilement des minutes et des heures comme sur un réveil.
Les minutes commencent donc à 00 et vont jusqu’à 59, produisant ensuite un changement d’heure, qui quant à elles vont de 00 à 23.
Une fois 23:59 atteint, votre fonction doit s’achever (donc vos boucles doivent être terminées également).



Utilisez deux boucles imbriquées

Voici un exemple de ce que vous devez afficher :

```
Terminal
~/I-SCC-000>
00:00
00:01
00:02
...
00:59
01:00
01:01
...
23:59
```




EXERCICE 04 - PROJET

Vous pouvez utiliser le temps qui vous reste pour continuer l'avancement de votre projet.