

CC PHP : Session 1 - 1h30 - Sur machine

Téléchargez l'archive **cc-22-php-sujet.tgz** de l'espace Moodle. Après extraction, renommez le répertoire obtenu **nom-prenom** où **nom** et **prenom** sont vos nom et prénom écrits en minuscules et sans espaces (par ex. **delafontaine-jean**). A l'issue de l'examen, archivez ce répertoire (`tar czf nom-prenom.tgz nom-prenom`) et déposez l'archive sur Moodle.

Les exercices sont indépendants et peuvent être traités dans l'ordre de votre choix. Pour visualiser ce qui est attendu, consultez ce [démonstrateur](#).

Exercice 1. Le fichier **angers.php** définit 2 tableaux associatifs - `$labels` et `$angers` - et importe le gabarit **gabarit.php**. Ce dernier produit un tableau HTML dont les étiquettes proviennent de `$labels` et les données de `$angers` en s'adaptant au contenu pour masquer et colorer colonnes et cellules. L'objectif est de modifier la variable `$angers` en répondant aux questions qui suivent pour obtenir la page illustrée en Figure 1. La valeur finale à produire pour `$angers` vous est donnée en commentaires en fin de fichier.

1. Ajoutez en première position dans `$angers` un tableau associatif utilisant les mêmes clés que les autres et modélisant les données : année 2000, température minimum -6.3 , maximum 34.3 , et moyenne 12.7 , rafale maximum 94.5 , ensoleillement $1533h\ 6min$, précipitations 775.4 , pression minimale 971 et maximale 1036.9 .
2. Supprimez le dernier tableau d'`$angers`.
3. Supprimez tout tableau d'`$angers` ayant pour pression minimale la valeur $--$.
4. Supprimez la “colonne” d'`$angers` correspondant aux températures moyennes.
5. Triez les tableaux d'`$angers` par température maximum croissante et, en cas d'égalité, par précipitations décroissantes.
6. Créez un tableau `$soleils` contenant les valeurs d'ensoleillement arrondies à l'heure supérieure quand le nombre de minutes est non nul. Calculez-en la moyenne que vous stockerez dans la variable `$soleil`.
7. Remplacez chaque valeur d'ensoleillement dans `$angers` par une paire contenant la valeur arrondie (calculée dans `$soleils`) et le caractère `S` (respectivement, `I`) si cette valeur est supérieure (resp. inférieure) à la moyenne `$soleil`.

Année	Température minimum [°C]	Température maximum [°C]	Rafale maximum [km/h]	Ensoleillement [h]	Précipitations totales [mm]	Pression horaire minimale [hPa]	Pression horaire maximale [hPa]
2008	-4.8	31.1	103.7	1765	714.4	983	1044.2
2007	-6.9	32.4	90.7	1782	748.4	990.2	1042.8
2014	-6.6	33.7	75.9	1237	529.4	987.1	1043.4
2004	-4.9	33.8	94.5	1867	568.8	985.4	1037.4
2000	-6.3	34.3	94.5	1534	775.4	971	1036.9
2009	-25.5	34.9	103.7	1959	725.8	973.6	1035
2002	-5	35	107.4	1740	794.2	977.3	1036.9
2013	-4.5	35.1	83.3	1672	714.7	987	1042
2010	-6.3	35.1	100	1970	580.6	969.1	1038.3
2001	-7.4	35.2	96.3	1835	786.4	986.7	1041
2011	-5.4	35.5	96.3	1543	486.7	986.3	1039.8
2012	-10.5	35.7	88.9	1849	734.8	983.8	1039.7
2015	-5.3	36	87	1954	660.5	985.5	1042.5
2018	-7.5	36.1	46.3	2000	869.4	981.3	1039.9
2005	-10.6	36.1	90.7	1921	497.8	985.6	1041
2006	-25.5	36.9	107.4	1815	686.2	980.8	1039.7
2003	-7.9	38.3	96.3	2112	715.4	976.4	1040.2
2022	-5	40.7	46.3	2000	442.6	988.8	1040.7

FIGURE 1 – Données climatiques pour la ville d'Angers (**angers.php**).

Exercice 2. L'objectif de cet exercice est de créer une page d'affichage (voir Figure 2) d'un fichier de log contenant l'historique des commandes apt exécutées sur une machine (**history.log**). Pour ce faire vous allez compléter le fichier **history.php** en deux étapes pouvant être traitées séparément.

1. Conversion d'un fichier en tableau.

Transformez le fichier **history.log** en tableau (nommé \$logs) où chaque élément du tableau est un tableau clef-valeur. Le résultat attendu est donné dans **logs.php**.

history.log contient l'historique des commandes apt exécutées sur une machine. Chaque entrée est séparée par une ligne vide (uniquement un retour à la ligne) et contient différentes valeurs dont le libellé et les informations sont séparés par " : ".

Votre code devra contenir toutes les sécurités requises pour la manipulation de fichier.

2. Affichage des données d'un tableau en HTML. Si vous n'avez pas réussi à traiter la première partie suivez les instructions en commentaire dans **history.php**.

Affichez \$logs en HTML en respectant les consignes suivantes dans le but d'obtenir le résultat de la Figure 2 :

- Chaque élément doit être placé dans un conteneur <div>.
- Le conteneur <div> contient un titre h2 affichant la date et un tableau de 2 colonnes et 5 lignes.
- L'affichage de la date peut être géré à l'aide des fonctions **date** et **strtotime**.
- Les données de la première colonne du tableau sont des données d'entête.
- La classe **code** est ajoutée à la donnée liée au libellé **CommandLine**.

N'hésitez pas à vous appuyer sur le **démonstrateur** et analyser le code HTML à obtenir à l'aide des outils de votre navigateur.

Historique apt

Monday 02 January 2023

```
Commandline : apt upgrade
Requested-By : etudiant (1000)
  Upgrade : libcommons-net-java:amd64 (3.6-1, 3.6-1+deb11u1), gir1.2-javascriptcoregtk-4.0:amd64 (2.38.2-1~deb11u1,
  2.38.3-1~deb11u1), gir1.2-webkit2-4.0:amd64 (2.38.2-1~deb11u1, 2.38.3-1~deb11u1), libjavascriptcoregtk-4.0-18:amd64
  (2.38.2-1~deb11u1, 2.38.3-1~deb11u1), libwebkit2gtk-4.0-37:amd64 (2.38.2-1~deb11u1, 2.38.3-1~deb11u1)
```

Install :
Remove :

Thursday 05 January 2023

```
Commandline : apt upgrade
Requested-By : etudiant (1000)
  Upgrade : opera-stable:amd64 (94.0.4606.38, 94.0.4606.54)
```

Install :
Remove :

Monday 09 January 2023

```
Commandline : apt upgrade
Requested-By : etudiant (1000)
  Upgrade : libreoffice-java-amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-calc:amd64 (1:7.0.4-4+deb11u4,
  1:7.0.4-4+deb11u5), libreoffice-sdbc-postgresql:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-gnome:amd64
  (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), uno-lib-private:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-base-
  core:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-core:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5),
  libjurt-java:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), gir1.2-localeview-0.1:amd64 (1:7.0.4-4+deb11u4,
  1:7.0.4-4+deb11u5), libbrld-java:amd64 (1:7.0.4-4+deb11u5), libreoffice-common:amd64
  (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-report-builder-bin:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5),
  libreoffice-draw:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-krita:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5),
  libreoffice-draw:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-presentation:amd64 (1:7.0.4-4+deb11u4,
  1:7.0.4-4+deb11u5), libreoffice-draw:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-impress:amd64
  (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-impress:amd64 (1:7.0.4-4+deb11u4, 1:7.0.4-4+deb11u5), libreoffice-
```

FIGURE 2 – Affichage de l'historique des commandes apt (**history.php**).

Exercice 3. Dans cet exercice, vous traiterez un formulaire d'inscription à une compétition sportive (voir Figure 3). Le formulaire comprend 5 champs de saisie : 3 champs de texte (nom, prenom et club), 1 champ de saisie numérique (age) et 1 menu déroulant (categorie). L'affichage et le traitement du formulaire sont tous deux effectués par le fichier **inscription.php** et les données sont envoyées en HTTP POST.

Il y a deux conditions à valider : catégorie+age et club. La première validation (catégorie+age) doit vérifier que l'age saisi correspond à la catégorie saisie ; cette validation utilise le tableau associatif `$categorie_age` (fourni dans le code).

La deuxième validation se fait sur le club. Elle utilise le tableau `$club_exclus` (fourni dans le code) qui contient une liste de clubs qui ne peuvent pas participer. Vous devrez donc valider que le club saisi ne figure pas dans la liste. Si le club n'est pas valide, conservez le nom, prénom et age du compétiteur dans leurs champs de saisie correspondants.

En fonction du résultat des validations, il y a trois messages possibles à afficher :

- Pas d'erreur : “Compétiteur inscrit(e) : NOM, Prénom”, où le nom du compétiteur est en majuscules et le prénom est capitalisé (Figure 4).
- Erreur “catégorie+age” : “Compétiteur non inscrit. Erreur : L'age indiqué ne correspond pas à la catégorie demandée.” (Figure 5).
- Erreur “club” : “Compétiteur non inscrit. Erreur : Le club X ne participe pas à cette compétition.”, où ‘X’ est le nom du club (Figure 6).

Les messages doivent s'afficher dans un paragraphe `<p>`. Le premier utilise la classe `ok` et les autres utilisent la classe `error`.

The form is titled "Inscription compétiteur". It contains five input fields: Nom (empty), Prénom (empty), Age (empty dropdown), Catégorie (dropdown set to Benjamin), and Club (empty input). A single "Inscrire" button is at the bottom.

FIGURE 3 – Formulaire initial

The form is titled "Inscription compétiteur". All fields are filled: Nom (THOMAS), Prénom (Alban), Age (10), Catégorie (Benjamin), and Club (empty input). Below the form, a green message reads "Compétiteur inscrit(e): THOMAS, Alban". A single "Inscrire" button is at the bottom.

FIGURE 4 – Cas : Inscription valide

The form is titled "Inscription compétiteur". Fields are filled: Nom (Thomas), Prénom (Alban), Age (10), Catégorie (Benjamin), and Club (empty input). Below the form, a red message reads "Compétiteur non inscrit. Erreur: L'age indiqué ne correspond pas à la catégorie demandée." A single "Inscrire" button is at the bottom.

FIGURE 5 – Cas : Erreur “catégorie+age”

The form is titled "Inscription compétiteur". Fields are filled: Nom (Thomas), Prénom (Alban), Age (10), Catégorie (Benjamin), and Club (empty input). Below the form, a red message reads "Compétiteur non inscrit. Erreur: Le club CDK85 ne participe pas à cette compétition." A single "Inscrire" button is at the bottom.

FIGURE 6 – Cas : Erreur “club”