

Cours de découverte métiers devops

Par Pr. Samuel OUYA

2022

SOMMAIRE

Cours du vendredi 25 Février	2
Séparation des métiers :	2
3 compétences à développer :	2
Organisation du travail.....	3
Mise en œuvre et leçons retenues	3
Evaluations :.....	3
Séquence 1 : Editeurs de textes.....	3
Séquence 2 : Découverte de bases de données	5
Partie 1	5
Partie II : Services web : installation, hébergement, formulaire html, langage php	15
Cours du Vendredi 25 mars 2022	27
Objectif :	27
Organisation	27
Cours du 1 avril 2022.....	31
Rappels sur la manipulation de bases de données avec php	31
Méthode de travail	32
Conception du programme.....	33
Cours samedi 02 avril 2022	35
Objectif :	35
Utilisation de la curl dans php.....	39
Initiation à javascript (jquery, ajax)	40
Sécurité et ajax	42
Bilan général	47
Compétences à développer pour être développeur frontend	47
Compétences à développer pour être développeur backend	48
Conclusion	48

Cours du vendredi 25 Février

Au début :

Séparation des métiers :

- développeurs d'applications
- techniciens systèmes et réseaux (Ops)
- techniciens en sécurité

Il faut une triple compétence pour accélérer le travail en entreprise

DevSecOps : techniciens capables de prendre en compte les besoins technologiques de l'entreprise

3 compétences à développer :

1- Développement :

a- Il me faut un éditeur de textes : nano, vim, sublime text

NB : éviter d'utiliser un logiciel de traitement de textes pour écrire de programmes

(LibreOffice, Microsoft office) car un logiciel de traitement de textes ajoute d'autres caractères à ce qui est saisi

b- il me faut un serveur qui héberge les applications qu'on met à disposition des utilisateurs et les utilisateurs utilisent des programmes clients pour accéder aux applications

Par exemple :

un navigateur est un client Web qu'on peut utiliser pour accéder à un site web hébergé sur un serveur

NB : un protocole est un ensemble de règles de communication

Un **service** est composé au minimum d'un **client** qui fait des demandes (requêtes) et un **serveur** qui traite ces requêtes et qui répondent au client.

Chaque service a ses propres règles dont l'ensemble constitue un protocole du service

Service	Protocole	Commentaire
Web	HTTP	Service permettant de gérer des sites Web et des API ()
Transfert de fichiers	FTP	Service d'envoi et de réception de fichiers
Messagerie classique	SMTP, POP, IMAP	SMTP : envoi ; POP télécharge ; IMAP donne accès aux mails
Messagerie instantanée (IM)	XMPP	Permet de faire du Chat ; audio, vidéo, transfert de fichiers, partage d'écran ; webconférences
TOIP	SIP, RTP, RTCP	SIP gère la signalisation ; RTP pour le transport de la voix et vidéo sur un réseau IP ; RTCP pour le contrôle des flux (données ayant même origine)

		et même destination)
--	--	----------------------

Client léger : un client d'un service est dit léger, s'il ne nécessite pas une installation côté de l'utilisateur ; généralement, les utilisateurs utilisent un navigateur pour accéder à un client léger que l'administrateur a déjà installé et paramétré

Organisation du travail

Les séquences du cours se trouvent sur la plateforme Moodle

Préparer les séquences :

- Objectif de la séquence
- Outils à installer

Mise en œuvre et leçons retenues

- Faire les TP
- Faire le rapport
- Donner les grandes lignes des leçons à retenir

Evaluations :

- Evaluations en ligne
- Les TP à rendre
- Examen final

Séquence 1 : Editeurs de textes

On doit les éditeurs de textes tels que nano et vim

1- Ce qu'on peut faire avec un éditeur de textes

- créer un fichier
- ouvrir un fichier
- écrire dans un fichier et sauvegarder
- écrire dans un fichier et ne pas sauvegarder
- rechercher un mot dans un fichier
- aller à une ligne dont le numéro est fourni

Pour éditer un fichier sous Linux

`noméditeur nomfichier`

L'éditeur nano est installé par défaut sous Linux ; ce qui n'est pas le cas de vim

Si on veut utiliser vim, il faut l'installer par la commande :

`apt install vim`

NB : la machine Linux n'arrive pas à se connecter à internet

Pour être dans un réseau, une machine a besoin des éléments TCP/IP :

- une adresse IP qui l'identifie de manière unique dans le réseau
- un masque de réseau pour lui permettre de savoir quelles sont les machines qui sont dans son réseau ou celles qui ne sont pas dans son réseau et qui nécessitent une passerelle pour les joindre
- une passerelle si on veut que notre machine communique avec une autre se trouvant dans un autre
- un DNS pour convertir des noms en adresse IP

NB : si on parvient à communiquer avec le serveur DNS de google par son adresse IP qui est 8.8.8.8

Mais on n'arrive pas à accéder des sites web, il y a un problème de conversion (résolution) de noms car on n'a pas le bon serveur DNS.

Pour préciser un bon DNS sous Linux, il faut éditer le fichier `/etc/resolv.conf` et on y met `nameserver 8.8.8.8`

Exemple

`nameserver 8.8.8.8`

Ainsi un éditeur de textes est important aussi bien pour le développeur et le technicien réseau

Utilisation de l'éditeur vim

1- il y a 2 modes dans vim

- mode insertion permettant d'écrire
- mode commande permettant de donner des ordres tel que :

w pour sauvegarder

wq pour sauvegarder et quitter

q pour quitter alors qu'on n'a pas modifié le fichier

q ! pour quitter sans sauvegarder

set num

nano +num nomfichier ; permet d'aller à la ligne num

vim +num nomfichier ; permet d'aller à la ligne

Avec vim, on fait de recherche en mode commande par /motrecherche

NB : vim est pratique pour écrire des programmes en langage python qui nécessite un décalage de 4 unités dans décalage certaines conditions car il le fait automatiquement à la place du développeur

Cours du vendredi 04 Mars 2022

Séquence 2 : Découverte de bases de données

Partie 1

Il y a 2 grandes familles de suites bureautiques

1- Suite bureautique de Microsoft (éditeur de textes avec traitement : **Word**, un tableur permettant de faire de traitement de données : **Excel**, un outil de présentation : **PowerPoint**)

2- Suite bureautique LibreOffice avec plusieurs composants dont :

- un traitement de textes : **writer**
- un tableur : **cal**
- un outil de présentation : **presenter**

Une base de données sert à organiser des données de manière à les manipuler facilement.

*Une base de données **relationnelle** est comme un classeur d'un tableur et contenant des tables.*

Chaque table a des champs

Règle : chaque champ d'une table dans une base de données doit avoir un type

Les principaux types :

- entier : **int** en anglais ;
- chaines de caractères : **varchar** ; **NB** ; il faut donner le nombre maximal de caractères prévu ; **varchar(30)**
- réel : **float**
- date
- heure

Les principaux serveurs de bases de données relationnelles sont :

- **mysql-server**
- **mariadb-server**
- **oracle**
- **postgresql**
- **sql server de Microsoft**

On utilise un langage appelé **SQL** pour manipuler une base de données relationnelles

Un **service de bases de données** comporte un **client** et un **serveur** de bases de données

Le client est utilisé pour :

- se **connecter** au serveur de base de données
- faire des **requêtes sql**
- se **déconnecter**

Tâches à faire :

- installer le programme client : **mysql-client**
- installer le programme serveur : **mysql-server**
- se connecter au serveur en utilisant le programme client (**mysql -u root -p**)
- montrer les bases de données disponibles (**show databases;**)
- créer une base de données appelée ecole (**create database ecole;**)
- accéder ou utiliser la base ecole (**use ecole;**)
- créer la table etudiant avec les champs **id** int, **prenom** varchar(30), **nom** varchar(30), **note** varchar(5) (**create table etudiant(id int, prenom varchar(30), nom varchar(30), note varchar(5));**)
- description de la table etudiant pour voir ses différents champs (**desc etudiant;**)
- Insérer l'enregistrement (1,Macky,Sall,12) (**insert into etudiant(id,prenom,nom,note) values (1, « Macky », »Sall », « 12 »);**)
- Afficher ou sélectionner tous les enregistrements de la table etudiant

(**select * from etudiant;**)

- modifier la note de l'enregistrement dont l'id est 1 en mettant sa note à 15

update etudiant set note= « 15 » where id=1 ;

- supprimer l'enregistrement de la table etudiant dont l'id est 1

delete from etudiant where id=1 ;

NB : les valeurs chaines de caractères sont entourés de « »

On vient de se rendre compte qu'il est possible de donner le même id à plusieurs enregistrements de la table etudiant.

Comment faire pour que l'identifiant soit unique : c'est la notion de **clé primaire** en bases de données ;

Une **clé primaire** d'une table est **un champ de la table** qui identifie de manière unique un enregistrement pour déclarer qu'un champ d'une table est une clé primaire on met : id int **primary key**.

Si on veut que la clé s'auto-incrémente on ajoute auto_increment

Exemple

créons une table president ayant comme clé id qui s'auto-incrémente

```
create table president(id int primary key auto_increment, prenom varchar(20), nom varchar(20), note varchar(5)) ;
```

Comme id est auto_increment, on a besoin de le préciser au moment de l'insertion des données

par exemple : nm

```
insert into president(prenom, nom, note) values(« macky », « Sall », « 14 ») ;
```

Comment modifier les champs d'une table ?

Comment altérer une table pour ajouter un nouveau champ ?

```
alter table president ADD email varchar(40) ;
```

comment altérer une table pour modifier le type d'un champ ?

```
alter table etudiant MODIFY id int primary key auto_increment ;
```

Comment supprimer un champ d'une table ?

```
Alter table etudiant DROP prenom ;
```

Mise en œuvre

Connectons-nous à la base de données avec le compte root

```
root@freewitch:~# mysql -u root -p
```

Enter password:

Affichons toutes les bases du serveur

```
mysql> show databases;
```

```
+-----+  
| Database |
```



```
+-----+
| information_schema |
| mysql              |
| performance_schema |
| roundcube          |
| sys                 |
+-----+
5 rows in set (0,16 sec)
```

Créons la base ecole

```
mysql> create database ecole
-> ;
Query OK, 1 row affected (0,00 sec)
```

```
mysql> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| ecole          |
| mysql          |
| performance_schema |
| roundcube      |
| sys            |
+-----+
6 rows in set (0,00 sec)
```

Connectons-nous à la base ecole

```
mysql> use ecole;
Database changed
mysql> show tables;
Empty set (0,00 sec)
```

créons la table étudiant avec ses différents champs

```
mysql> create table etudiant(id int, prenom varchar(30),nom varchar(30),note varchar(5));
Query OK, 0 rows affected (0,31 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_ecole |
+-----+
| etudiant         |
+-----+
1 row in set (0,00 sec)
```

Décrivons la table étudiant pour voir ses champs

```
mysql> desc etudiant;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
prenom	varchar(30)	YES		NULL	
nom	varchar(30)	YES		NULL	
note	varchar(5)	YES		NULL	

```
4 rows in set (0,01 sec)
```

Insérons quelques enregistrements dans la table etudiant

```
mysql> insert into etudiant(id,prenom,nom,note) values(1,"Macky","Sall","12");
```

```
Query OK, 1 row affected (0,10 sec)
```

```
mysql> insert into etudiant(id,prenom,nom,note) values(2,"Ousmane","Sonko","11");
```

```
Query OK, 1 row affected (0,12 sec)
```

```
mysql> insert into etudiant(id,prenom,nom,note) values(3,"Abdoulaye","Wade","13");
```

```
Query OK, 1 row affected (0,03 sec)
```

```
mysql> select * from etudiant;
```

id	prenom	nom	note
1	Macky	Sall	12
2	Ousmane	Sonko	11
3	Abdoulaye	Wade	13

```
3 rows in set (0,00 sec)
```

Modifions la note de Macky en lui donnant 15

```
mysql> update etudiant set note="15" where id=1;
```

```
Query OK, 1 row affected (0,12 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from etudiant;
```

id	prenom	nom	note
1	Macky	Sall	15
2	Ousmane	Sonko	11
3	Abdoulaye	Wade	13

```
| 1 | Macky | Sall | 15 |
| 2 | Ousmane | Sonko | 11 |
| 3 | Abdoulaye | Wade | 13 |
+-----+-----+-----+-----+
3 rows in set (0,00 sec)
```

Supprimons l'enregistrement dont l'id est 1

```
mysql> delete from etudiant where id=1;
Query OK, 1 row affected (0,07 sec)
```

```
mysql> select * from etudiant;
+-----+-----+-----+-----+
| id | prenom | nom | note |
+-----+-----+-----+-----+
| 2 | Ousmane | Sonko | 11 |
| 3 | Abdoulaye | Wade | 13 |
+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

création de la table president en précisant une clé primaire auto_increment

```
mysql> create table president(id int primary key auto_increment,prenom varchar(20),nom
varchar(20),note varchar(5));
Query OK, 0 rows affected (0,07 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_ecole |
+-----+
| etudiant |
| president |
+-----+
2 rows in set (0,00 sec)
```

Insérons plusieurs enregistrements en 1 seule requete sql

```
mysql> insert into president(prenom,nom,note)
values("Macky","Sall","10"),("Ousmane","Sonko","15"),("Ali","Bongo","12"),("Denis","Sassou","
17");
Query OK, 4 rows affected (0,03 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from president;
```

```
+----+-----+-----+-----+
| id | prenom | nom  | note |
+----+-----+-----+-----+
| 1 | Macky  | Sall | 10  |
| 2 | Ousmane | Sonko | 15  |
| 3 | Ali    | Bongo | 12  |
| 4 | Denis  | Sassou | 17  |
+----+-----+-----+-----+
4 rows in set (0,00 sec)
```

Ajoutons le champ email à la table president

```
mysql> alter table president ADD email varchar(40);
Query OK, 0 rows affected (0,17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc president;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | NO   | PRI | NULL    | auto_increment |
| prenom | varchar(20) | YES  |     | NULL    |                |
| nom   | varchar(20) | YES  |     | NULL    |                |
| note  | varchar(5) | YES  |     | NULL    |                |
| email | varchar(40) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0,00 sec)
```

```
mysql> select * from president;
+----+-----+-----+-----+-----+
| id | prenom | nom  | note | email |
+----+-----+-----+-----+-----+
| 1 | Macky  | Sall | 10  | NULL  |
| 2 | Ousmane | Sonko | 15  | NULL  |
| 3 | Ali    | Bongo | 12  | NULL  |
| 4 | Denis  | Sassou | 17  | NULL  |
+----+-----+-----+-----+-----+
4 rows in set (0,00 sec)
```

Ajoutons un email à l'enregistrement dont l'id vaut 1

```
mysql> update president set email="macky.sall@gouv.sn" where id=1;
Query OK, 1 row affected (0,05 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from president;
```

```
+----+-----+-----+-----+-----+
| id | prenom | nom  | note | email          |
+----+-----+-----+-----+-----+
| 1  | Macky  | Sall | 10   | macky.sall@gouv.sn |
| 2  | Ousmane | Sonko | 15   | NULL            |
| 3  | Ali    | Bongo | 12   | NULL            |
| 4  | Denis  | Sassou | 17   | NULL            |
+----+-----+-----+-----+-----+
4 rows in set (0,00 sec)
```

```
mysql> desc etudiant;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)   | NO   | PRI | NULL    | auto_increment |
| nom   | varchar(30) | YES  |     | NULL    |              |
| note  | varchar(5) | YES  |     | NULL    |              |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0,01 sec)
```

Supprimons la table etudiant

```
mysql> drop table etudiant;
```

Query OK, 0 rows affected (0,06 sec)

```
mysql> show tables;
```

```
+-----+
| Tables_in_ecole |
+-----+
| president       |
+-----+
1 row in set (0,00 sec)
```

Calcul de la somme des notes

```
mysql> Select SUM(note) AS TotalItemsOrdered From president;
```

```
+-----+
| TotalItemsOrdered |
+-----+
| 54                |
+-----+
1 row in set (0,05 sec)
```

```
mysql> Select SUM(note) AS Total From president;
+-----+
| Total |
+-----+
|  54 |
+-----+
1 row in set (0,00 sec)
```

Calcul de la moyenne des notes des presidents

```
mysql> select AVG(note) AS moyenne from president;
+-----+
| moyenne |
+-----+
|  13.5 |
+-----+
1 row in set (0,00 sec)
```

Calcul de la moyenne maximale

```
mysql> select MAX(note) AS max  from president;
+-----+
| max |
+-----+
| 17 |
+-----+
1 row in set (0,00 sec)
```

Détermination de la note minimale

```
mysql> select MIN(note) AS max  from president;
+-----+
| max |
+-----+
| 10 |
+-----+
1 row in set (0,00 sec)
```

```
mysql>
```

```
mysql> select MIN(note) AS max  from president;
+-----+
| max |
+-----+
| 10 |
```

+-----+

1 row in set (0,00 sec)

Afficher les présidents qui n'ont pas d'email

```
mysql> select * From president where email IS NULL;
```

+----+-----+-----+-----+-----+

| id | prenom | nom | note | email |

+----+-----+-----+-----+-----+

| 2 | Ousmane | Sonko | 15 | NULL |

| 3 | Ali | Bongo | 12 | NULL |

| 4 | Denis | Sassou | 17 | NULL |

+----+-----+-----+-----+-----+

3 rows in set (0,00 sec)

Sélectionner les présidents qui ont une note supérieure à 11

```
mysql> select * From president where note >11;
```

+----+-----+-----+-----+-----+

| id | prenom | nom | note | email |

+----+-----+-----+-----+-----+

| 2 | Ousmane | Sonko | 15 | NULL |

| 3 | Ali | Bongo | 12 | NULL |

| 4 | Denis | Sassou | 17 | NULL |

+----+-----+-----+-----+-----+

3 rows in set (0,00 sec)

Sélectionner les présidents dont les notes sont comprises entre 15 et 18

```
mysql> select * From president where note >=15 and note <=18;
```

+----+-----+-----+-----+-----+

| id | prenom | nom | note | email |

+----+-----+-----+-----+-----+

| 2 | Ousmane | Sonko | 15 | NULL |

| 4 | Denis | Sassou | 17 | NULL |

+----+-----+-----+-----+-----+

2 rows in set (0,06 sec)

Utiliser BETWEEN

```
mysql> select * From president where note BETWEEN 15 AND 18;
```

+----+-----+-----+-----+-----+

| id | prenom | nom | note | email |

+----+-----+-----+-----+-----+

```
| 2 | Ousmane | Sonko | 15 | NULL |
| 4 | Denis   | Sassou | 17 | NULL |
+---+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

Liste des president dont le nom se termine par o
%

```
mysql> insert into president(prenom,nom,note) values("Mathieu","Kerekou","13");
Query OK, 1 row affected (0,06 sec)
```

```
mysql> select * From president where nom LIKE("%u");
+---+-----+-----+-----+-----+
| id | prenom | nom   | note | email |
+---+-----+-----+-----+-----+
| 4 | Denis  | Sassou | 17   | NULL  |
| 5 | Mathieu | Kerekou | 13   | NULL  |
+---+-----+-----+-----+-----+
2 rows in set (0,00 sec)
```

Exercice :

- 1- afficher les presidents dont les noms commencent par S ;
- 2- Afficher par ordre alphabetique les presidents par nom et par prenom

Gestion des utilisateurs et privilèges dans mysql

Pour créer l'utilisateur bouki dans mysql, on utilise la requête :
create user bouki identified by « motdepasse » ;

Comment donner tous les privileges à bouki sur toutes les tables de la base ecole ?

```
grant all privileges on ecole.* to bouki ;
```

On applique les privileges par la commande
flush privileges ;

Exercice sur la gestion des utilisateurs Mysql

- 1- afficher tous les utilisateurs de mysql
- 2- afficher les droits d'un utilisateur
- 3- supprimer un utilisateur
- 4- révoquer(annuler) les privilèges d'un utilisateur

Partie II : Services web : installation, hébergement, formulaire html, langage php

Objectif :

1- Savoir installer le serveur web apache2

apt install apache2

2- identifier le dossier dans lequel mettre le contenu de son site, ce dossier porte le nom de **DocumentRoot** du site

/var/www/html

3- connaître la structure générale d'une page html

4- apprendre à accéder à site web dont le dossier se trouve dans **/var/www/html** à travers un navigateur par exemple si le dossier est **/var/www/html/tp1**

http://localhost/tp1/nomfichier

5- quelle la structure d'un programme en langage php ?

6- Que faut il installer pour que son serveur web apache puisse executer du code php ?

libapache2-mod-php

```
mysql> insert into president(prenom,nom,note,email)
```

```
values("Macky","Sall","13","macky.sall@gouv.sn"),("Idrissa","Seck","12","idy.seck@gmail.com"),
("Ousmane","Sonko","11","ousmane.sonko@hotmail.com"),("Denis","Sassou","14","denis.sassou@gouv.cg");
```

```
Query OK, 4 rows affected (0,01 sec)
```

```
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from president;
```

```
+----+-----+-----+-----+-----+
| id | prenom | nom  | note | email                |
+----+-----+-----+-----+-----+
| 1  | Macky  | Sall | 13   | maccky.sall@gouv.sn  |
| 2  | Idrissa | Seck | 12   | idy.seck@gmail.com   |
| 3  | Ousmane | Sonko | 11   | ousmane.sonko@hotmail.com |
| 4  | Denis  | Sassou | 14   | denis.sassou@gouv.cg |
+----+-----+-----+-----+-----+
```

```
4 rows in set (0,00 sec)
```

```
mysql> create user bouki identified by "passer";
```

```
Query OK, 0 rows affected (0,00 sec)
```

```
root@membre:~# mysql -u toto -p
```

```
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 4
```

```
Server version: 5.7.37-0ubuntu0.18.04.1 (Ubuntu)
```

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or 'h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
+-----+
1 row in set (0,00 sec)
root@membre:~# apt install apache2
root@membre:~# cd /var/www/html/
```

Créons un dossier tp1 dans ce repertoire
root@membre:/var/www/html# mkdir tp1
root@membre:/var/www/html# ls
index.html tp1
root@membre:/var/www/html#
nano premier.html

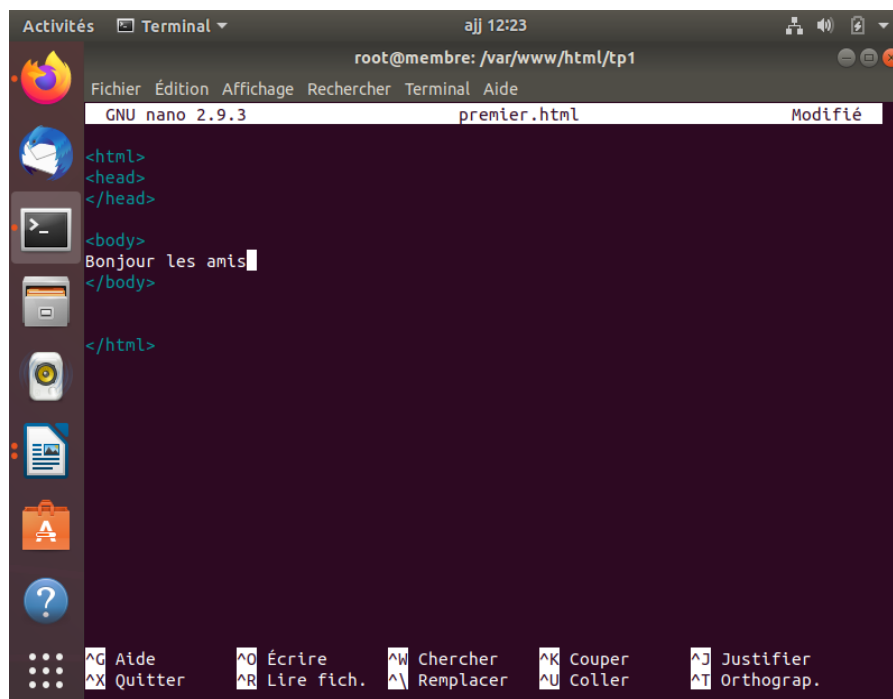


Fig 2.1 Premier.html

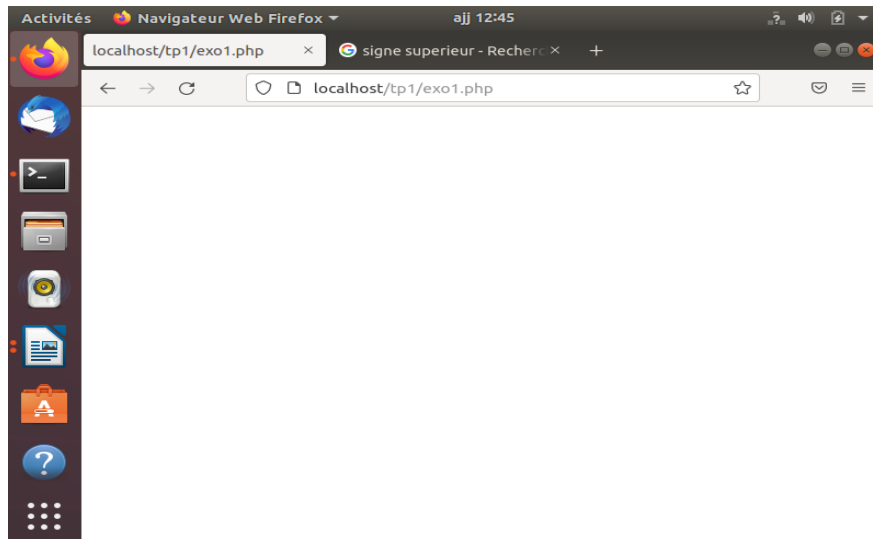
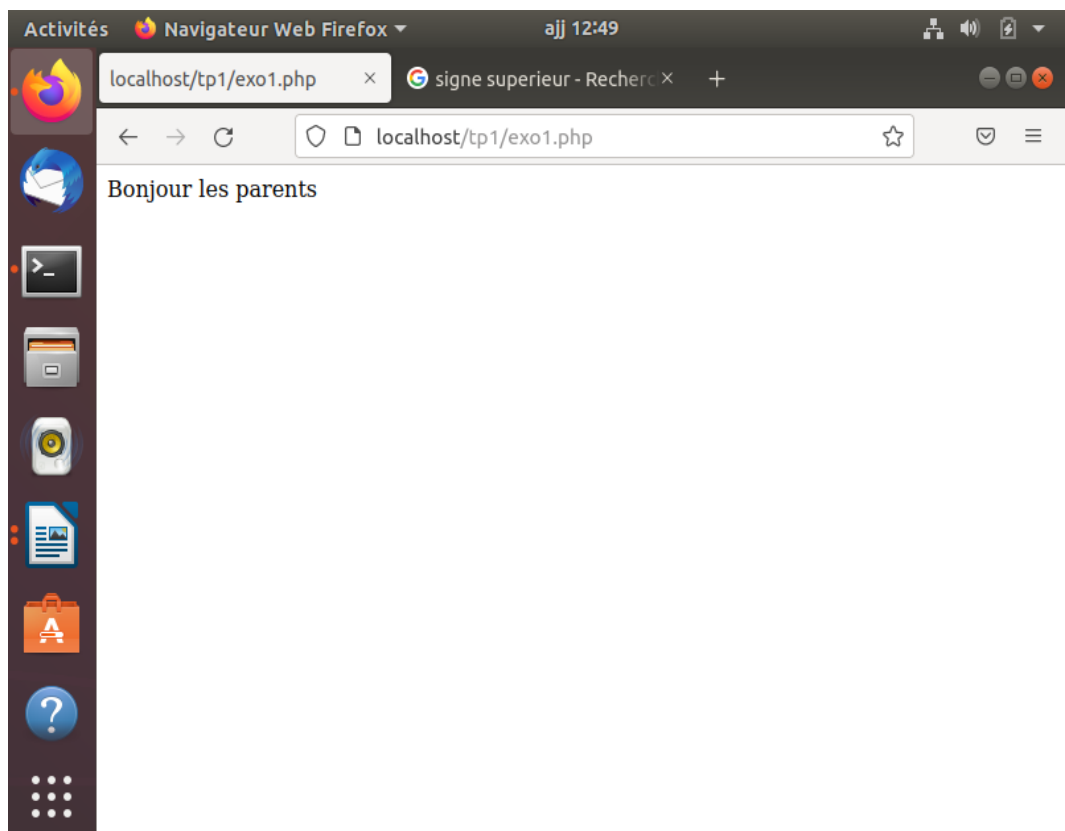


Figure 2.2 exo1.php

Installons libapache2-mod-php

root@membre:/var/www/html/tp1# **apt install libapache2-mod-php**



Grace à l'installation du module libapache2-mod-php , mon serveur web est capable d'exécuter du code php

Passons à la création du formulaire html

Pour cela, copions le fichier premier.html en formulaire.html

root@membre:/var/www/html/tp1# cp premier.html formulaire.html

Un formulaire sert à saisir des données pour les envoyer à un autre programme pour traitement
Il y a plusieurs méthodes d'envoi des données :

- la methode get
- la methode post

En tant que être humain, je comprends que dans la déclaration d'un formulaire html, on doit au minimum préciser :

la méthode et le programme qui doit traiter les données
on utilise pour cela les attributs method et action
method= 'post' action= 'exo2.php'

```
<?php
$prenom=$_POST['prenom'];
$nom=$_POST['nom'];

echo "Bonjour $prenom $nom ";
?>
```

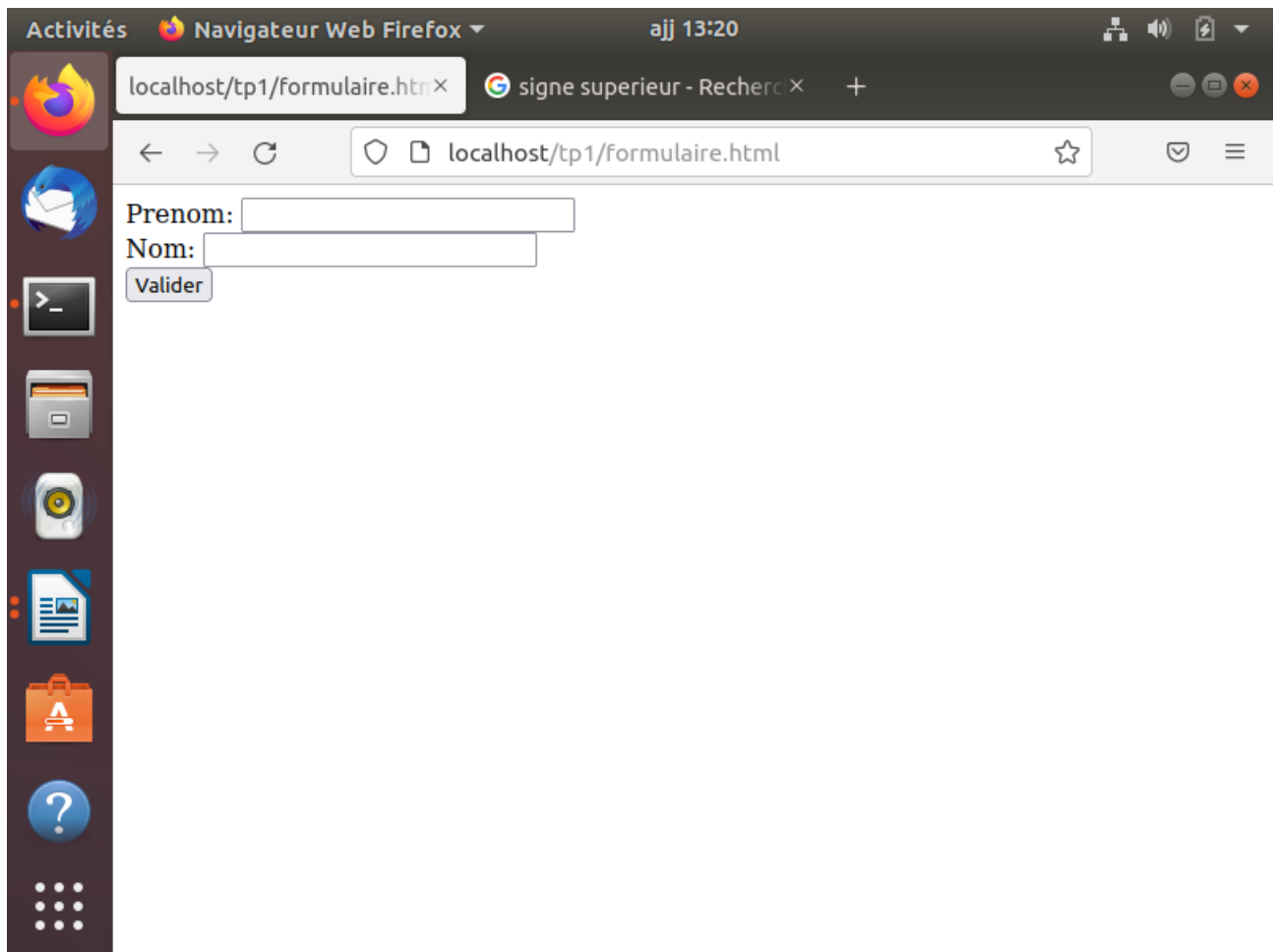


Figure 2.3 Formulaire.html

```
root@membrane:/var/www/html/tp1# cp exo1.php exo2.php
```

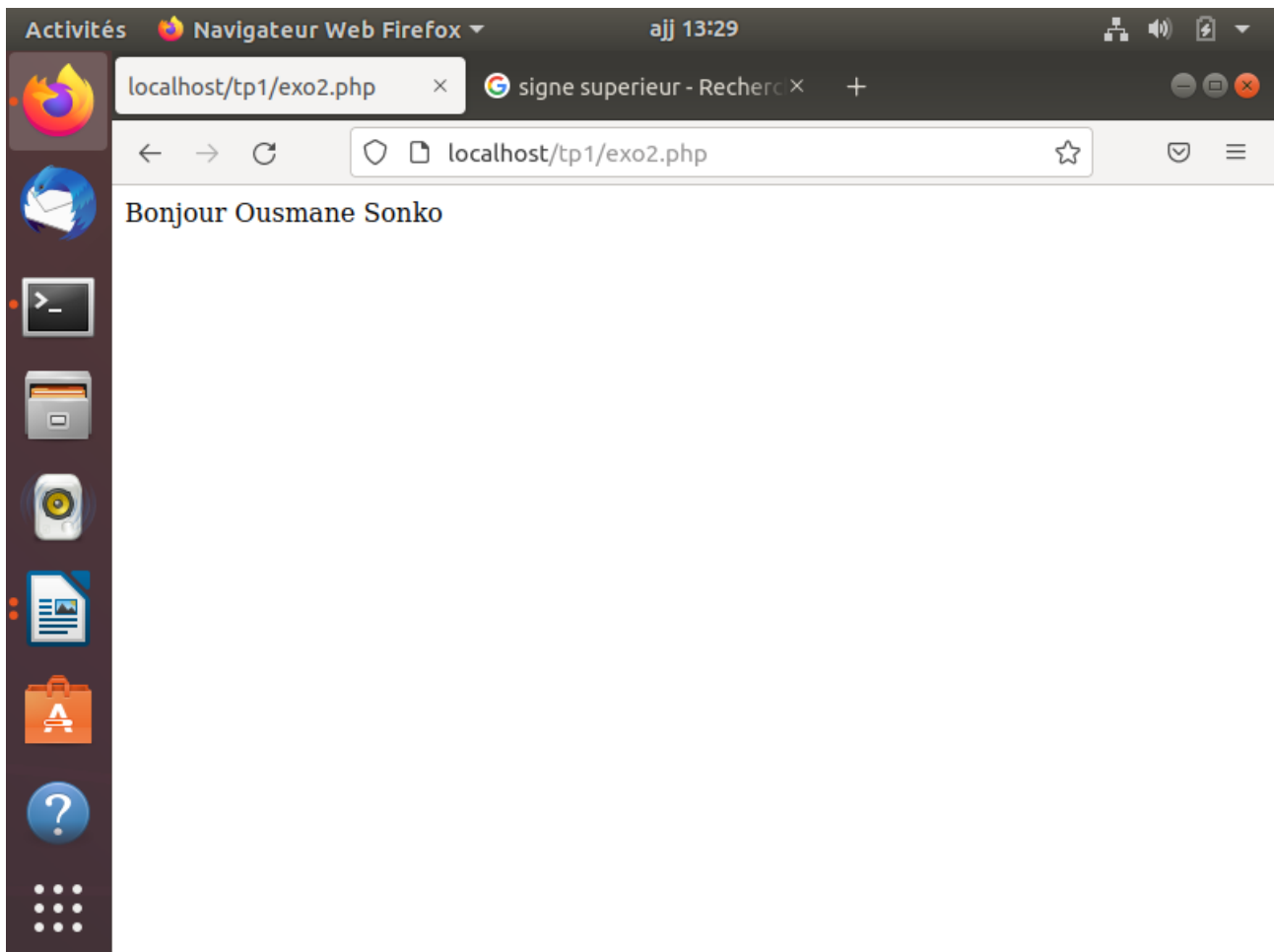


Figure 2.4 Exo2.php

Conclusion :

On a appris aujourd'hui à :

- 1- faire des opérations avancées sur une base de données
- 2- gérer les utilisateurs et leurs privilèges
- 3- créer des formulaires html
- 4- créer des programmes php
- 5- récupérer les données envoyées par un formulaire dans un programme en php grâce aux supervariables globales POST['nomvariable']

1- instructionnelle condition en php

```
if (condition) { instructions}
else {instructions}
root@membre:~# cd /var/www/html/
root@membre:/var/www/html# mkdir tp2
root@membre:/var/www/html#
```

exo1if.php

on fixe la variable \$nom à bouki ; c'est a dire on affecte (=) à nom la valeur bouki

if si le nom est babadi , on affiche vous n'etes pas le bienvenu
sinon on affiche bonjour patron

```
<?php
$nom="bouki";
if ($nom=="babadi") {
    echo "vous n'etes pas la bienvenue";
} else {
    echo "Bonjour patron";
}
?>
```

programme exofor.php

on cree un tableau indexé tab qu'on remplit par les indices de 0 à 9
on parcourt le tableau et on affiche les contenus des cases

```
$tab=array() ;
for($i=0;$i <10 ; $i++)
{ $tab[$i]=$i ; }
```

```
for($i=0;$i <10 ; $i++)
{ echo « $tab[$i] » ; }
```

contenu réel de exofor.php

Algo :

debut

initailiser le tableau tab

pour i allant de 0 à 9 affecter à tab[i] la valeur i

pour i allant de 0 à 9 afficher tab[i] et revenir à la ligne
fin

```
<?php
$tab=array();

for($i=0; $i<10; $i++) {
    $tab[$i]=$i;
}

for($i=0; $i<10; $i++) {
    echo "$tab[$i] <br>";
}

?>
```

NB :

pour dire pour i allant de 0 à n **for(\$i=0; \$i<n; \$i++)**

On peut aussi initialiser un tableau par

\$tab=[]

l'instruction **print_r()** pour affichage brute

```
<?php
$tab=[];

for($i=10; $i<21; $i++) {
    $tab[$i]=$i;
}
print_r($tab);

?>
```

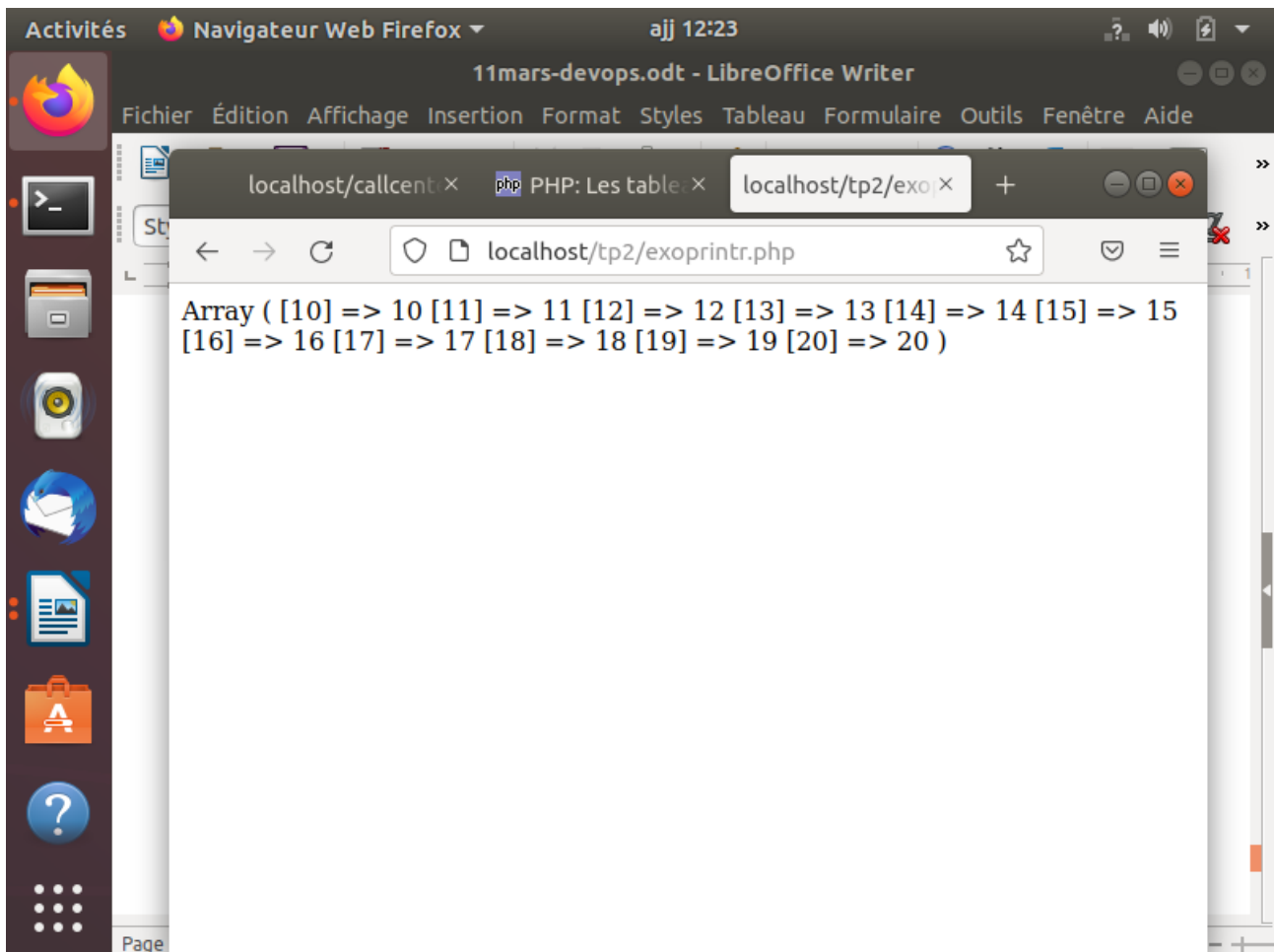



Figure 2.5 exoprint.php

`print_r()` a permis de comprendre la structure generale d'un tableau en php

Un tableau peut être créé en utilisant la structure de langage `array()`. Il prend un nombre illimité de paramètres, chacun séparé par une virgule, sous la forme d'une paire `key => value`.

```
array(
    key  => value,
    key2 => value2,
    key3 => value3,
    ...
)
```

while

créer le tableau `tab` et le remplir par les indices de 0 à 9
on utilise `while` pour afficher les éléments du tableau

`i=0`

tant que `i` est inférieur à 10 alors

debut
affiche tab[i]
et incremente i
fin

```
<?php
$tab=[];

for($i=0; $i<=9; $i++) {
    $tab[$i]=$i;
}
$i=0;
while($i<10)
{
    echo "$tab[$i] <br>";
    $i++;
}

?>
```

Tableau associatif et foreach

On crée le tableau associatif age ayant comme clé le nom d'un user et comme valeur son age

On cree un autre tableau associatif contenant l'email des memes users

On affiche un tableau réel contenant nom et age

macky	macky@gouv.sn	62
sonko	sonko@gmail.com	45
seck	seck@ythies.sn	63

NB un tableau html se cree en utilisant la balise table

tr pour ligne

td pour colonne

```

<html>
<head>
</head>
<body>
<table border="1px">
<tr><td>Macky</td><td>macky@gouv.sn</td><td>62</td></tr>
<tr><td>Sonko</td><td>sonko@gmail.com</td><td>45</td></tr>
<tr><td>Seck</td><td>seck@thies.sn</td><td>63</td></tr>
</table>
</body>
</html>

```

on cree un tableau à l'aide de php et on apprend à utiliser foreach

```

<?php
$ages = ['Macky' => 62, 'Sonko' => 45, 'Seck' => 63];
$mails['Macky'] = 'macky@gouv.sn';
$mails['Sonko'] = 'sonko@gmail.com';
$mails['Seck'] = 'seck@thies.sn';
$chaine="<html><head></head><body><table border='1px'>";

foreach ($ages as $nom => $age) {
$chaine=$chaine."<tr><td>$nom</td><td>$mails[$nom]</td><td>$age</td></tr>";
}
$chaine=$chaine."</table></body></html>";

echo "$chaine";

?>

```

fichier log est tres important pour un développeur

```

root@membre:/var/www/html/tp2# tail -f /var/log/apache2/error.log
[Fri Mar 18 10:18:15.531270 2022] [mpm_prefork:notice] [pid 953] AH00163: Apache/2.4.29
(Ubuntu) configured -- resuming normal operations
[Fri Mar 18 10:18:15.531287 2022] [core:notice] [pid 953] AH00094: Command line:
'/usr/sbin/apache2'
[Fri Mar 18 13:37:44.840340 2022] [php7:emerg] [pid 2553] [client 127.0.0.1:56822] PHP Parse
error: syntax error, unexpected " (T_ENCAPSED_AND_WHITESPACE), expecting '-' or identifier
(T_STRING) or variable (T_VARIABLE) or number (T_NUM_STRING) in
/var/www/html/tp2/tab.php on line 13

```

Cours du Vendredi 25 mars 2022

Objectif :

- 1- Ecrire un programme en php permettant de manipuler une base de données mysql
- 2- Faire le Couplage Téléphonie Informatique (CTI) permettant à un agent qui travaille dans un centre d'appels de renseigner des infos par rapport à un client qui appelle :

les champs à renseigner :

prenom	nom	telephone	adresse	email	service
--------	-----	-----------	---------	-------	---------

Il y a 6 besoins **fonctionnels** en termes de champs par enregistrement mais nous allons ajouter le champ id comme clé primaire de la table **client** de manière à identifier de manière unique chaque potentiel client appelé prospect

Les thématiques abordés :

- 1- Comment tester l'envoi des données à un programme par la methode GET
- 2- la balise HTML **textarea**
- 3- **Quelle démarche adopter pour la manipulation d'une base de données à travers un programme PHP ?**

a- se connecter au serveur de bases de données en fournissant :

- l'adresse IP du serveur de bases de données (par défaut, c'est **localhost**)
- **le nom de l'utilisateur avec lequel on se connecte**
- **le mot de passe de l'utilisateur**
- **le nom de la base à utiliser**

Nous allons pour cela créer la fonction php

```
$conn=mysqli_connect('localhost ','bouki','passer','callcenter');
```

b- preparer la requete dans une variable

```
$req= 'select * from client ' ;
```

c- executer la requete grace à la fonction

```
$result= mysqli_query($conn,$req)
```

d- NB : le seul cas où on a besoin d'exploiter les résultats avec soit :

- `mysqli_fetch_assoc($result)` pour parcourir les résultats ligne par ligne
- `$stab=mysqli_fetch_all($result)` pour récupérer tous les résultats de la requête dans un tableau

Organisation

Tâches à faire :

- 1- installer php-mysql
- 2- Se connecter au serveur de BD en ligne de commande à mysql pour créer la base callcenter

- 3- Se connecter à la base et on crée la tables client avec les 7 champs dont le dernier service de type text
- 4- On crée le compte bouki avec le mot de passe passer123
- 5- Donner tous les privilèges sur la base callcenter1 à bouki2 et les appliquer

6- programmer

Algo call.php

- a- Récupérer le téléphone de l'utilisateur par la méthode GET
- b- se connecter à la base de données callcenter1
- c- Préparer dans la variable req

\$req='select * from client where telephone=\$telephone'

- d- Executer la requete

\$result= mysqli_query(\$conn,\$req)

si le numero de telephone existe dans la table client, on cree le formulaire dans une variable php et on l'affiche ; on utilisera l'attribut value de la balise input pour afficher les anciennes valeurs pour tester l'existence du numero de telephone, je ferai appel à mysqli_num_rows(\$result) ; s'il est strictement positif, alors le numero existe sinon le numero n'existe pas

sinon on cree un formulaire vide et on l'affiche

NB : je vais concevoir 2 programmes

1- create.php

et insere l'enregistrement dans la table

Algo de create.php

- a- se connecter à la base de données
- b- preparer la requete d'insertion
- c- executer la requete d'insertion

2- update.php qui va recuperer l'adresse, le service et le telephone et fait la mise à jour des infos du client dont le numero de telephone a été fourni

Algo update

- a- se connecter à la base de données
- b- preparer la requete de mise à jour
- c- executer la requete de mise à jour

Mise en œuvre

1- écrire create.php
 2- écrire update.php
 3- écrire call.php
 mysql> create database callcenter1;
 Query OK, 1 row affected (0,00 sec)
 mysql> use callcenter1
 Database changed
 mysql> create table client(id int primary key auto_increment, prenom varchar(30), nom varchar(30),adresse varchar(30),telephone varchar(30), email varchar(30), service text);
 Query OK, 0 rows affected (0,14 sec)

mysql> create user bouki2@localhost identified by "passer";
 Query OK, 0 rows affected (0,00 sec)
 mysql> grant all privileges on callcenter1.* to bouki2;
 ERROR 1133 (42000): Can't find any matching row in the user table
 mysql> grant all privileges on callcenter1.* to bouki2@localhost;
 Query OK, 0 rows affected (0,00 sec)

mysql> flush privileges;
 Query OK, 0 rows affected (0,00 sec)

Voici le code de

create.php

```
<?php
$prenom=$_REQUEST['prenom'];
$nom=$_REQUEST['nom'];
$adresse=$_REQUEST['adresse'];
$email=$_REQUEST['email'];
$telephone=$_REQUEST['telephone'];
$service=$_REQUEST['service'];
$conn=mysqli_connect('localhost','bouki2','passer','callcenter1');
$req="insert into client(prenom,nom,adresse,email,telephone,service)
values('$prenom','$nom','$adresse','$email','$telephone','$service)";
$result=mysqli_query($conn,$req);
echo " creation reussie";
?>
```

code de update.php

```
<?php
$adresse=$_REQUEST['adresse'];
$telephone=$_REQUEST['telephone'];
$service=$_REQUEST['service'];
$conn=mysqli_connect('localhost','bouki2','passer','callcenter1');
$req="update client set adresse='$adresse', service='$service' where telephone='$telephone' ";
$result=mysqli_query($conn,$req);
```

```

    echo " Mise a jour reussie";
?>

```

Code de call.php

```

<?php
$telephone=$_REQUEST['telephone'];
$conn=mysqli_connect('localhost','bouki2','passer','callcenter1');
$req="select * from client where telephone=$telephone";
$result=mysqli_query($conn,$req);
if ( mysqli_num_rows($result) > 0)
{
    while ($row=mysqli_fetch_assoc( $result))

    {
        $prenom=$row['prenom'];$non=$row['nom'];$telephone=$row['telephone'];$email=$row['email'];
        $service=$row['service'];$adresse=$row['adresse'];

        $chaine="<form method='post' action='update.php'>
        Prenom : <input type='text' name='prenom' value=$prenom > <br>
        Nom : <input type='text' name='nom' value= $non> <br>
        Telephone :<input type='text' name='telephone' value= $telephone> <br>
        Email : <input type='text' name='email' value= $email > <br>
        Adresse : <input type='text' name='adresse' value= $adresse> <br>
        Service : <textarea type='text' name='service' cols='40' rows='10'> $service </textarea><br>
        <input type='submit' value='Valider'>";
        echo $chaine;

    }
}
else
{
    $chaine="<form method='post' action='create.php'>
    Prenom : <input type='text' name='prenom'> <br>
    Nom : <input type='text' name='nom'> <br>
    Telephone :<input type='text' name='telephone'> <br>
    Email : <input type='text' name='email'> <br>
    Adresse : <input type='text' name='adresse'> <br>
    Service : <textarea type='text' name='service' cols='40' rows='10'> </textarea><br>
    <input type='submit' value='Valider'>";
    echo $chaine;
}

?>

```

Cours du 1 avril 2022

Rappels sur la manipulation de bases de données avec php

1- Préréquis

les paquets à installer

- apache2
- libapache2-mod-php
- php
- mysql-server
- mysql-client
- php-mysql

Demarche de programmation php/mysql

- 1- On crée une base de données et les tables qu'il faut
- 2- On crée un utilisateur sur le serveur de base de données et on lui donne tous les droits
- 3- Quelque soit le type de requete, la demarche est **pratiquement** la même :
 - se connecter à la base de données
 - preparer la requete dans une variable
 - executer la requete

si la requete est de type **select** , il faut recuperer les resultats et les traiter

Les fonctions php de manipulation de bases de données :

- mysqli_connect(servername,username,password dbname)
- mysqli_query(conn,req)
- mysqli_fetch_assoc(result)
- mysqli_fetch_all(result)

En php :

- Entrees/sorties :

On a vu 2 méthodes pour faire des entrées en php :

Soit on passe par un formulaire

soit on passe par un navigateur `http://servername/uriappli?para1=valeur¶2=valeur2` ***lorsque la methode d'envoi est get***

Pour recuperer les parametres envoyés dans un programme php, on utilise les variables superglobales : `$_GET['para']`, `$_POST['para']`, `$_REQUEST['para']`

- instructions conditionnelles if else
- boucles while et foreach
- tableaux
- fonctions

Devoir 1

On vous demande de mettre en place une application de gestion de compte bancaire des présidents. Cette application doit permettre à travers des formulaires de :

- créer un compte à un président avec un solde de départ
- d'afficher dans un tableau tous les comptes
- de saisir le numero de compte d'un president et le versement et on fait la mise à jour
- de supprimer un compte d'un president en donnant son numcompte, son prenom et son nom

NB : le schema de la table client(id,prenom,nom,numcompte,code,solde)

Méthode de travail

- On écrit la partie backend de notre application en php
- On écrit la partie frontend de notre application soit en utilisant des formulaires ou des tableaux html pour l'affichage

Définition de fonction en php

Pour créer une fonction, on utilise l'instruction «function» suivi du nom que l'on souhaite donner à cette fonction, suivi de 2 parenthèses.

Les instructions à exécuter lors de l'appel de cette fonction seront placée entre deux accolades.

Cas 1 : fonction sans parametre

```
<?php
function maFonctionA()
{
    // instructions ici
}
?>
```

Cas2 : fonction avec parametre

```
<?php
function maFonctionC($a, $b, $c)
{
    // instructions ici
}
?>
```

Appel de fonction

```
<?php  
maFonctionC(12, 25, 1);  
?>
```

Conception du programme

Je prévois 4 fonctions dans la partie backend

0- database() ; se connecter à la base bicis

1- create(prenom,nom,numcompte,code,solde) ; qui insere dans la table client

2- read() ; renvoie tous les comptes des presidents sous forme de tableau

3- versement(numcompte,prenom,nom,montant) ; ajouter montant au solde du client

4- supprimer(numcompte,prenom,nom)

Remarque :

Une fonction peut retourner une valeur grace à l'instruction **return variable**

On se connecte au serveur de BD pour créer la base bicis, créer un compte un utilisateur et lui donner tous les droits

```
mysql> create database bicis;
```

```
Query OK, 1 row affected (0,00 sec)
```

```
mysql> use bicis;
```

```
Database changed
```

```
mysql> create table client(id int primary key auto_increment, prenom varchar(30), nom  
varchar(30),numcompte varchar(30),code varchar(4),solde varchar(30));
```

```
Query OK, 0 rows affected (0,01 sec)
```

```
mysql> create user babadi identified by 'passer123';
```

```
Query OK, 0 rows affected (0,08 sec)
```

```
mysql> grant all privileges on bicis.* to babadi;
```

```
Query OK, 0 rows affected (0,00 sec)
```

```
mysql> flush privileges;
```

```
Query OK, 0 rows affected (0,00 sec)
```

NB : notre programme est bicis.php

On crée une fonction database() de connexion à la base bicis car toute action sur la base, on doit s'y connecter

```

function database()

{

    $conn= mysqli_connect("localhost","babadi","passer123","bicis");
    return $conn;
}

function create($prenom,$nom,$numcompte,$code,$solde)
{
    $conn=database();
    $req="insert                into                client(prenom,nom,numcompte,code,solde)
values('$prenom','$nom','$numcompte','$code','$solde')";
    $result=mysqli_query($conn,$req);
    mysqli_close($conn);
    echo 'Nouveau compte cree avec succes';

}

function lire()
{
    $conn=database();
    $req="select * from client";
    $result=mysqli_query($conn,$req);
    $tab=mysqli_fetch_all($result);
    mysqli_close($conn);
    return $tab;

}

function versement($prenom,$nom,$numcompte,$montant)
{
    $conn=database();
    $req1="select solde from  client where numcompte='$numcompte' and prenom='$prenom' and
nom='$nom'";
    $result1=mysqli_query($conn,$req1);
    if (mysqli_num_rows($result1)>0)
    { $tab1=mysqli_fetch_all($result1);$nouveausolde=$tab1[0][0]+$montant;
    $req2="update client set solde='$nouveausolde' where numcompte='$numcompte'";
    $result2=mysqli_query($conn,$req2);
    $message="versement effectué: votre nouveau solde est de $nouveausolde"; }
    else { $message="données erronés"; }
    mysqli_close($conn);
    return $message;
}

```

```

function retrait($prenom,$nom,$numcompte,$montant)
{
    $conn=database();
    $req1="select solde from client where numcompte='$numcompte' and prenom='$prenom' and nom='$nom'";
    $result1=mysqli_query($conn,$req1);
    if (mysqli_num_rows($result1)>0) { $tab1=mysqli_fetch_all($result1); if($tab1[0][0]>$montant)
    { $nouveausolde=$tab1[0]$
    $req2="update client set solde='$nouveausolde' where numcompte='$numcompte'";
    $result2=mysqli_query($conn,$req2);
    $message="retrait effectué avec succes: votre nouveau solde est de $nouveausolde"; }
    else { $message ="votre solde est insuffisant";}

        }

    else { $message="données erronés"; }
    mysqli_close($conn);
    return $message;
}

```

```

function supprimer($prenom,$nom,$numcompte)
{
    $conn=database();
    $req="delete from client where numcompte='$numcompte' and prenom='$prenom' and nom='$nom'";
    $result=mysqli_query($conn,$req);
    if (mysqli_affected_rows($conn)>0 ) {
    echo "suppression reussie";
    }
    else echo "informations erronées";
}

```

Pour appeler il suffit d'écrire son nom et ses paramètres

Cours samedi 02 avril 2022

Objectif :

Améliorer les fonctions écrites la veille en intégrant des contrôles

Par exemple, un utilisateur ne doit pas retirer plus que son solde disponible.

Un autre exemple : il faut veiller à ce que les numcompte existent et correspondent bien au prenom et au nom du client avant de faire des actions

1- comment tester si une requete mysql a bien marché ou non

cas de la requete insert, on utilise : mysqli_num_rows(\$result1)

exemple

```
function versement($prenom,$nom,$numcompte,$montant)
{
    $conn=database();
    $req1="select solde from client where numcompte='$numcompte' and prenom='$prenom' and nom='$nom'";
    $result1=mysqli_query($conn,$req1);
    if (mysqli_num_rows($result1)>0)
    { $tab1=mysqli_fetch_all($result1);$nouveausolde=$tab1[0][0]+$montant;
      $req2="update client set solde='$nouveausolde' where numcompte='$numcompte'";
      $result2=mysqli_query($conn,$req2);
      $message="versement effectué: votre nouveau solde est de $nouveausolde"; }
    else { $message="données erronées"; }
    mysqli_close($conn);
    echo $message;
}
```

function retrait(\$prenom,\$nom,\$numcompte,\$montant)

```
{
    $conn=database();
    $req1="select solde from client where numcompte='$numcompte' and prenom='$prenom' and nom='$nom'";
    $result1=mysqli_query($conn,$req1);
    if (mysqli_num_rows($result1)>0) { $tab1=mysqli_fetch_all($result1); if($tab1[0][0]>$montant)
    { $nouveausolde=$tab1[0][0]-$montant;
      $req2="update client set solde='$nouveausolde' where numcompte='$numcompte'";
      $result2=mysqli_query($conn,$req2);
      $message="retrait effectué avec succes: votre nouveau solde est de $nouveausolde"; }
      else { $message ="votre solde est insuffisant";}
    }

    else { $message="données erronées"; }
    mysqli_close($conn);
    echo $message;
}
```

function create(\$prenom,\$nom,\$numcompte,\$code,\$solde)

```
{
    $conn=database();
    $req1="insert into client (prenom,nom,numcompte,code,solde) values ($prenom,$nom,$numcompte,$code,$solde)";
    $result=mysqli_query($conn,$req1);
    mysqli_close($conn);
}
```

```
echo 'Nouveau compte cree avec succes';
```

```
}
```

```
function lire()
```

```
{  
    $conn=database();  
    $req="select * from client";  
    $result=mysqli_query($conn,$req);  
    $tab=mysqli_fetch_all($result);  
    mysqli_close($conn);  
    return $tab;  
}
```

Test de bonne exécution des requêtes insert, update, delete

On utilise

function retrait(\$prenom,\$nom,\$numcompte,\$montant)

```
{  
    $conn=database();  
    $req1="select solde from client where numcompte='$numcompte' and prenom='$prenom' and  
nom='$nom'";  
    $result1=mysqli_query($conn,$req1);  
    if (mysqli_num_rows($result1)>0) { $tab1=mysqli_fetch_all($result1); if($tab1[0][0]>$montant)  
{ $nouveausolde=$tab1[0][0]-$montant;  
    $req2="update client set solde='$nouveausolde' where numcompte='$numcompte'";  
    $result2=mysqli_query($conn,$req2);  
    $message="retrait effectué avec succes: votre nouveau solde est de $nouveausolde"; }  
    else { $message ="votre solde est insuffisant";}  
    }  
}
```

```
else { $message="données erronés"; }
```

```
mysqli_close($conn);
```

```
echo $message;
```

```
}
```

function supprimer(\$prenom,\$nom,\$numcompte)

```
{  
    $conn=database();  
    $req="delete from client where numcompte='$numcompte' and prenom='$prenom' and  
nom='$nom'";  
    $result=mysqli_query($conn,$req);  
    if (mysqli_affected_rows($conn)>0 ) {  
    echo "suppression reussie";  
    }
```

```

}
else echo "informations erronées";
}

```

Comment appeler des fonctions dans un programme principal

Pour pouvoir appeler nos fonctions qui sont dans bicis.php, nous utilisons une astuce, en lui envoyant le paramètre go et selon (**switch case**) la valeur, on appelle la bonne fonction

Il faut remarquer, pour retourner le tableau, nous étions obligé d'**installer php-json** pour disposer de la fonction json_encode() pour convertir le tableau en json .

Côté frontend dans formulaireirelire.php, on utilise la fonction **json_decode(\$tab,true)**

```

$go=$_REQUEST['go'];
switch ($go) {
    case "1":
        echo json_encode(lire());
        break;
    case "2":
        $prenom=$_REQUEST['prenom'];
        $nom=$_REQUEST['nom'];
        $numcompte=$_REQUEST['numcompte'];
        $code=$_REQUEST['code'];
        $solde=$_REQUEST['solde'];
        create($prenom,$nom,$numcompte,$code,$solde);
        break;
    case "3":
        $prenom=$_REQUEST['prenom'];
        $nom=$_REQUEST['nom'];
        $numcompte=$_REQUEST['numcompte'];
        $montant=$_REQUEST['montant'];
        versement($prenom,$nom,$numcompte,$montant);
        break;
    case "4":
        $prenom=$_REQUEST['prenom'];
        $nom=$_REQUEST['nom'];
        $numcompte=$_REQUEST['numcompte'];
        $montant=$_REQUEST['montant'];
        retrait($prenom,$nom,$numcompte,$montant);
        break;
    case "5":
        $prenom=$_REQUEST['prenom'];
        $nom=$_REQUEST['nom'];
        $numcompte=$_REQUEST['numcompte'];
        supprimer($prenom,$nom,$numcompte);
        break;}

```

Nous avons terminé la partie backend de notre application bicis.php

Maintenant passons à la partie frontend de notre application qui est un ensemble de 5 programmes

application		
formulairelire.php	Pour afficher tous les comptes dans un tableau	On fait appel à la fonction curl qui est un client HTTP avancé
formulairecreate.php	Pour créer un nouveau compte en donnant : prenom,nom,numcompte,code,solde	On fait appel à javascript (ajax)
formulaireversement.php	Pour permettre de faire un versement en donnant prenom,nom,numcompte,montant	On fait appel à javascript (ajax)
formulaire retrait.php	Pour permettre de faire un retrait en donnant prenom,nom,numcompte,montant	On fait appel à javascript (ajax)
formulairesupprimer.php	Pour supprimer un compte en donnant prenom,nom,numcompte	On fait appel à javascript (ajax)

On donne la méthodologie d'utilisation de curl dans un programme php
on installe paquet php-curl

Utilisation de la curl dans php

algo d'utilisation de curl

- a- on initialise a connexion avec curl_init()
- b- on envoie les options avec curl_setopt()
- c-on execute la requete http avec la curl_exec()
- d- on ferme la connexion http avec curl_connect()

mise en œuvre : formulaireirelire.php

```
<?php
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'http://localhost/tppro/bicis.php?go=1');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
echo "Tableau des comptes";
$tab = curl_exec($ch);
$tab=json_decode($tab,true);
curl_close($ch);
```



```

$chaine="<table border='1px'><tr><td>Prenom</td>      <td>Nom</td><td>Numcompte</td>
<td>Code</td><td>Solde</td></tr>";
foreach($stab as $stab1){
$prenom=$stab1[1];$nom=$stab1[2];$numcompte=$stab1[3];$code=$stab1[4];$solde=$stab1[5];
$chaine=$chaine."<tr><td>$prenom</td><td>$nom</td><td>$numcompte</td><td>$code</td><t
d>$solde</td></tr>";
}
$chaine=$chaine."</table>";
echo "$chaine";
?>

```

Initiation à javascript (jquery, ajax)

Il faut telecharger le code jquery et le mettre dans le jquery.js dans le même dossier de votre programme

1-Pour faire du jQuery il faut commencer par déclarer les scripts JQuery dans votre page. On fait cela au début dans le Head :

```

<HTML lang="fr">
  <HEAD>

    <meta charset="utf-8" />
    <script src="jquery.js" type="text/javascript"></script>

```

2-Ensuite il faudra déclarer un formulaire.

Le formulaire se nomme ici FormMonFormulaire01

```

<BODY>

  <div id="bloc_page">

    <FORM METHOD="POST" id="FormMonFormulaire01">
    Prenom: <input type="text" name="prenom" id="prenom" size=10 value=""><br>
    Nom: <input type="text" name="nom" id="nom" size=10 value=""><br>
    Numcompte: <input type="text" name="numcompte" id="numcompte" size=10 value=""$
    Code: <input type="text" name="code" id="code" size=10 value=""><br>
    Solde: <input type="text" name="solde" id="solde" size=10 value=""><br>
    <input id="funcId" name="go" type="hidden" value="2"><br>
    </FORM>

    <INPUT id="butFonctionAjax" type="BUTTON" value="Lancer fonction Ajax"><br>

```

```

<div STYLE="margin-left:auto; margin-right:auto; width:400px; position:relativ$
<span id="status"></span><br>
</div> <!-- div bloc_page -->
</BODY>

```

3- On déclarera une balise DIV qui contiendra ce qui est retournée par la page appelée en ajax

```

<div STYLE="margin-left:auto; margin-right:auto; width:400px; position:relative; font-size:10pt;
font-family:verdana; border: 2px black solid;" id="divAffichageResultat"></div><br>
<span id="status"></span><br>
</div> <!-- div bloc_page -->

```

4- Au début de notre page on déclarera notre code JQuery. Dans se code on fera référence à l'**ID** du bouton et on s'intéressera à son événement **click**. Quand on cliquera sur le bouton on déclenchera une fonction

```

<SCRIPT TYPE="text/javascript">
    $(function() {

        $("#butFonctionAjax").click(function() {

            var form = $('#FormMonFormulaire01');
            var str = form.serialize();

            $.ajax( {
                type: "POST",
                url: 'bicis.php',
                data: str,
                success: function( response ) {
                    $('#divAffichageResultat').html( response ); //Affichage de l'url
cible, ici AjaxTemplate02.php, dans une DIV
                    $('#status').text('Post');
                    //console.log( response );
                },
                error: function( response ) {
                    $('#status').text('Erreur pour poster le formulaire : '+
response.status + " " + response.statusText);
                    //console.log( response );
                }
            } );
        });

    });
</SCRIPT>

```

NB : il faut le script dans la balise head

En gras ci-dessus, la recuperation du formulaire , conversion des données en chaine de caractères et precision de la fonction à appeler avec les données necessaires

Sécurité et ajax

Objectif

code de :essaiajax1.php

```
<!DOCTYPE html>
<HTML lang="fr">
  <HEAD>

    <meta charset="utf-8" />
    <script src="jquery.js" type="text/javascript"></script>

    <title>Titre du site</title>

    <SCRIPT TYPE="text/javascript">
      $(function() {

        $("#butFonctionAjax").click(function() {

          var form = $('#FormMonFormulaire01');
          var str = form.serialize();

          $.ajax( {
            type: "POST",
            url: 'AjaxTemplate021.php',
            data: str,
            success: function( response ) {
              $('#divAffichageResultat').html( response ); //Affichage de l'url
              $('#status').text('Post');
              //console.log( response );
            },
            error: function( response ) {
              $('#status').text('Erreur pour poster le formulaire : '+
response.status + " " + response.statusText);
              //console.log( response );
            }
          });
        });
      });
    </SCRIPT>
  </HEAD>
</HTML>
```

```
});
</SCRIPT>
```

```
</HEAD>
```

```
<BODY> <div id="bloc_page">
```

```
<FORM METHOD="POST" id="FormMonFormulaire01">
<input type="text" name="TxtChampInput01" id="TxtChampInput01" size=10 value=""><br>
<input type="text" name="TxtChampInput02" id="TxtChampInput02" size=10 value="" ><br>
</FORM>
```

```
<INPUT id="butFonctionAjax" type="BUTTON" value="Lancer fonction Ajax"><br>
<div STYLE="margin-left:auto; margin-right:auto; width:400px; position:relative; font-size:10pt;
font-family:verdana; border: 2px black solid;" id="divAffichageResultat"></div><b$
<span id="status"></span><br>
</div> <!-- div bloc_page -->
</BODY>
</HTML>
```

Code de AjaxTemplate021.php

```
<?php

$varChamp01 = $_POST['TxtChampInput01']; //Récupération de la valeur Champ01
$varChamp02 = $_POST['TxtChampInput02']; //Récupération de la valeur Champ02

$MaDateMiseEnLigne = date("Y-m-d h:i:s");

$chaine="<table border='1px'><tr><td>Date<td><td>$MaDateMiseEnLigne<td></tr>
        <tr><td>Valeur champ 01<td><td>$varChamp01<td></tr>
        <tr><td>Valeur champ 02<td><td>$varChamp02<td></tr></table>";
echo "$chaine";

?>
```

Methode 2 ajax

contrôle à faire du coté utilisateur

Code de AjaxTemplate20.php

```
<!DOCTYPE html>
<HTML lang="fr">
<HEAD>
```

```

<meta charset="utf-8" />
<script src="jquery.js" type="text/javascript"></script>

<title>Titre du site</title>

<SCRIPT TYPE="text/javascript">
    $(function() {

        $("#butFonctionAjax").click(function() {
            $('#status').text('Lancement de la fonction Ajax');

            var varValeurChampInput01 = $("#TxtChampInput01").val();
            var varValeurChampInput02 = $("#TxtChampInput02").val();

            if (varValeurChampInput01.length > 0){
                $('#status').text('ArticleId : ' + varValeurChampInput01);

                $('#divAffichageResultat').load('AjaxTemplate021.php',
{   TxtChampInput01:varValeurChampInput01,   TxtChampInput02:varValeurChampInput02   },
function( response, status, xhr ) {
                    if ( status == "error" ) {
                        var msg = "Sorry but there was an error: ";
                        //alert(msg);
                        $( "#divAffichageResultat" ).html( msg + xhr.status + " " +
xhr.statusText );
                    }
                    else{ //success espéré
                        $('#status').text('C est Ok : ' + status);
                    }
                });
            }
            else{
                $('#status').text('Pas de valeur indiquée dans le champ de saisi 01');
            }

        });

    });
</SCRIPT>

</HEAD>

<BODY>

```

```

        <div id="bloc_page">
<?php

    //Tiré de http://www.siteduzero.com/informatique/tutoriels/les-magic-quotes-ou-guillemets-
    magiques/desactiver-les-magic-quotes
    //Cette option permet de retirer les magic quotes sur un serveur où c'est activé et où vous
    n'avez pas la main. C'est importante lorsque l'on poste récupère des valeurs de champs Input et
    textarea faute de quoi, par exemple, "C'es$
    function stripslashes_r($var) // Fonction qui supprime l'effet des magic quotes
    {
        if(is_array($var)) // Si la variable passée en argument est un array, on appelle la fonction
        stripslashes_r dessus
        {
            return array_map('stripslashes_r', $var);
        }
        else // Sinon, un simple stripslashes suffit
        {
            return stripslashes($var);
        }
    }

    if(get_magic_quotes_gpc()) // Si les magic quotes sont activés, on les désactive avec notre
    super fonction ! ;)
    {
        $_GET = stripslashes_r($_GET);
        $_POST = stripslashes_r($_POST);
        $_COOKIE = stripslashes_r($_COOKIE);
    }

    echo "<input type=\"text\" name=\"TxtChampInput01\" id=\"TxtChampInput01\" size=10
    value=\"\" /><br />\n";
    echo "<input type=\"text\" name=\"TxtChampInput02\" id=\"TxtChampInput02\" size=10
    value=\"\" /><br />\n";

    echo "<INPUT id=\"butFonctionAjax\" type=\"BUTTON\" value=\"Lancer fonction
    Ajax\"><br />\n";

    echo "<div STYLE=\"margin-left:auto; margin-right:auto; width:400px; position:relative; font-
    size:10pt; font-family:verdana; border: 2px black solid;\" id=\"divAffichageResultat\"></div>\n<br
    />";
    echo "<span id=\"status\"></span><br />\n";

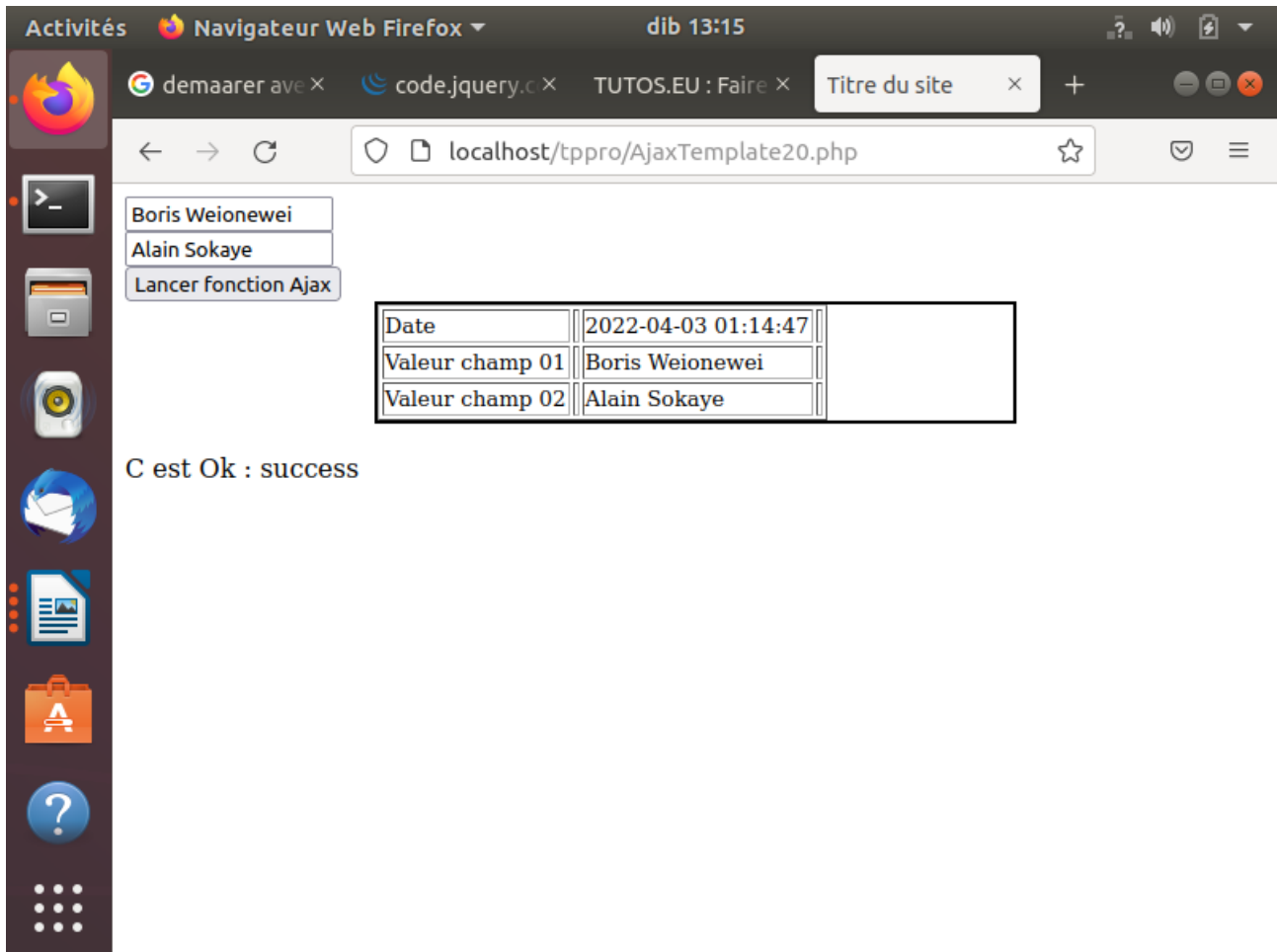
?>

</div> <!-- div bloc_page -->

```

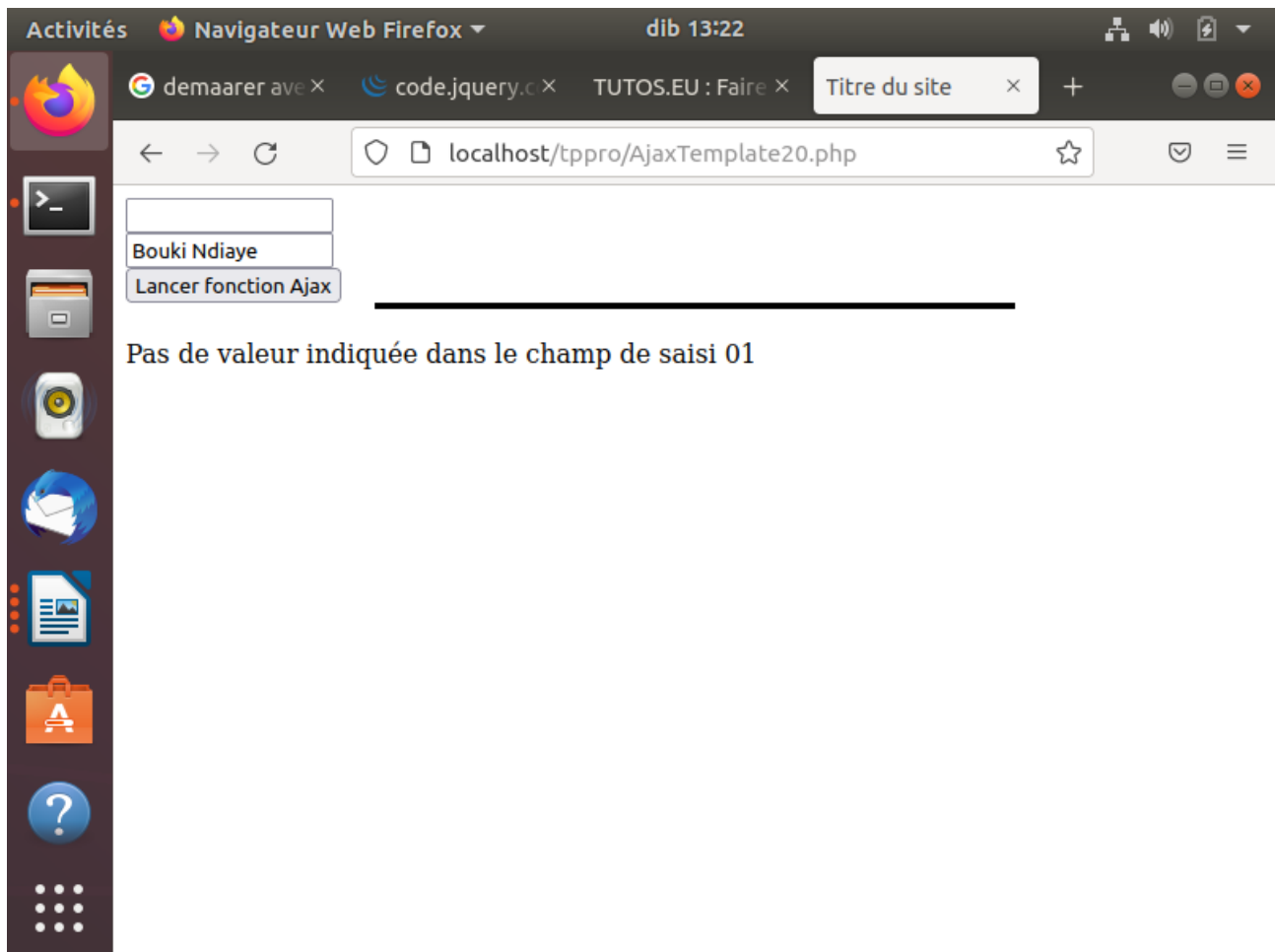
</BODY>
</HTML>

Resultat 1



fig

Cas de non saisi du champ 1



fig

Bilan général

Compétences à développer pour être développeur frontend

- Langages de programmation : html, css, php, javascript

Bibliothèque : jquery

Fonction php	Paquet nécessaire
curl_init(), curl_setopt(),curl_exec(),curl_close()	php-curl
json_encode(\$tab), json_decode(\$tab,true)	php-json
explode(separateur,chaine)	

Compétences à développer pour être développeur backend

- une bonne connaissance de mysql
- savoir activer le journal des événements de mysql à travers les paramètres `general_log_file` et `general_log`
- savoir utiliser la commande de consultation du journal de mysql `tail -f //var//log/mysql/mysql.log`
- savoir consulter le journal des erreurs de apache2 `tail -f /var/log/apache2/error.log`

Les instructions conditionnelles php

- `if () {} else {}`
- `switch ($var) {`
 `case « valeur1 » : instruction1;`
 `break ;`
 `case « valeur2 » : instruction2 ;`
 `break ;`
 `}`

Les boucles

- `foreach($tab as $ligne) { instruction :}`
- `while (condition) {instruction ;}`

Conclusion

- Le métier de **DevOps** nécessite aussi bien des compétences en développement qu'en opération (système (**Linux,Windows**) et réseaux (**équipements et logiciels**))
- Il faut mettre les deux ans à venir pour développer de compétences en outils et technologies de **frontend** à savoir **html/css, javascript** pour faire des contrôles et des actions coté navigateurs avant d'envoyer des données au serveur
- Les 2 ans doivent être mis à profit pour maîtriser des outils et technologies coté **backend** : **bases de données, php,python,nodejs,java**