

Les bases de PHP

22 mars 2016

Table des matières

Objectifs et déroulement	1
Exercice 1 : configuration	1
Exercice 2	2
Exercice 3	5
Exercice 4	7
Exercice 5	11
Exercice 6	12

Objectifs et déroulement

Quelques exercices en PHP de préférence sous Linux.

- Pour ce TP vous travaillerez par groupes de deux (UN groupe de trois est permis si le nombre de personnes dans votre groupe de TP ne permet pas uniquement des groupes de deux) ;
- Vous aurez besoin d'une machine virtuelle Debian par étudiant.
- Dans le cadre de ce TP, vous remettrez une archive tar.gz contenant tous les codes PHP des exercices. Barème approximatif. Questions Q1 à Q20 sur 1 point.

Exercice 1 : configuration

Dans cet exercice vous devez vérifier la configuration des logiciels comme Apache, MySQL, PHP et activer le module userdir qui offre la possibilité au serveur Web de servir des pages situées dans le home directory de l'utilisateur courant. Les pages sont dans /home//public_html et sont accessibles via [http:// @ip_du_serveur/~](http://@ip_du_serveur/~)
...

Afin de gérer la BDD MySQL autrement qu'en ligne de commande l'outil phpmyadmin est tout indiqué. Pour faire installer par l'installateur de paquet Apache, PHP il suffit de lui demander d'installer phpmyadmin, il gère les dépendances. Cependant, il faut lui demander d'installer MySQL car phpmyadmin peut fonctionner sans qu'il y ai de BDD sur la machine. En d'autres termes phpmyadmin installé sur votre machine locale peut administrer éventuellement un serveur MySQL installé sur un autre ordinateur. Voici la trame de ce qu'il faut faire.

```
[root@machina]# apt-get install phpmyadmin mysql-server
...
[user@machina]$ ls -l /etc/apache2/mods-enabled/
```

```
...
lrwxrwxrwx 1 root root 30 nov. 17 10:09 userdir.conf -> ../mods-available/userdir.conf
lrwxrwxrwx 1 root root 30 nov. 17 10:10 userdir.load -> ../mods-available/userdir.load
...
[user@machina]$
```

Il faut vérifier que les liens symboliques sont bien créés dans le répertoire des modules d'Apache modsenabled pointant sur le répertoire des modules eux-mêmes mods-available. Enfin, il faut activer à l'aide de la commande `a2enmod` le module `userdir`.

Exercice 2

Soit le code PHP suivant.

```
<html>
<head>
<title>Loops </title>
</head>
<body>

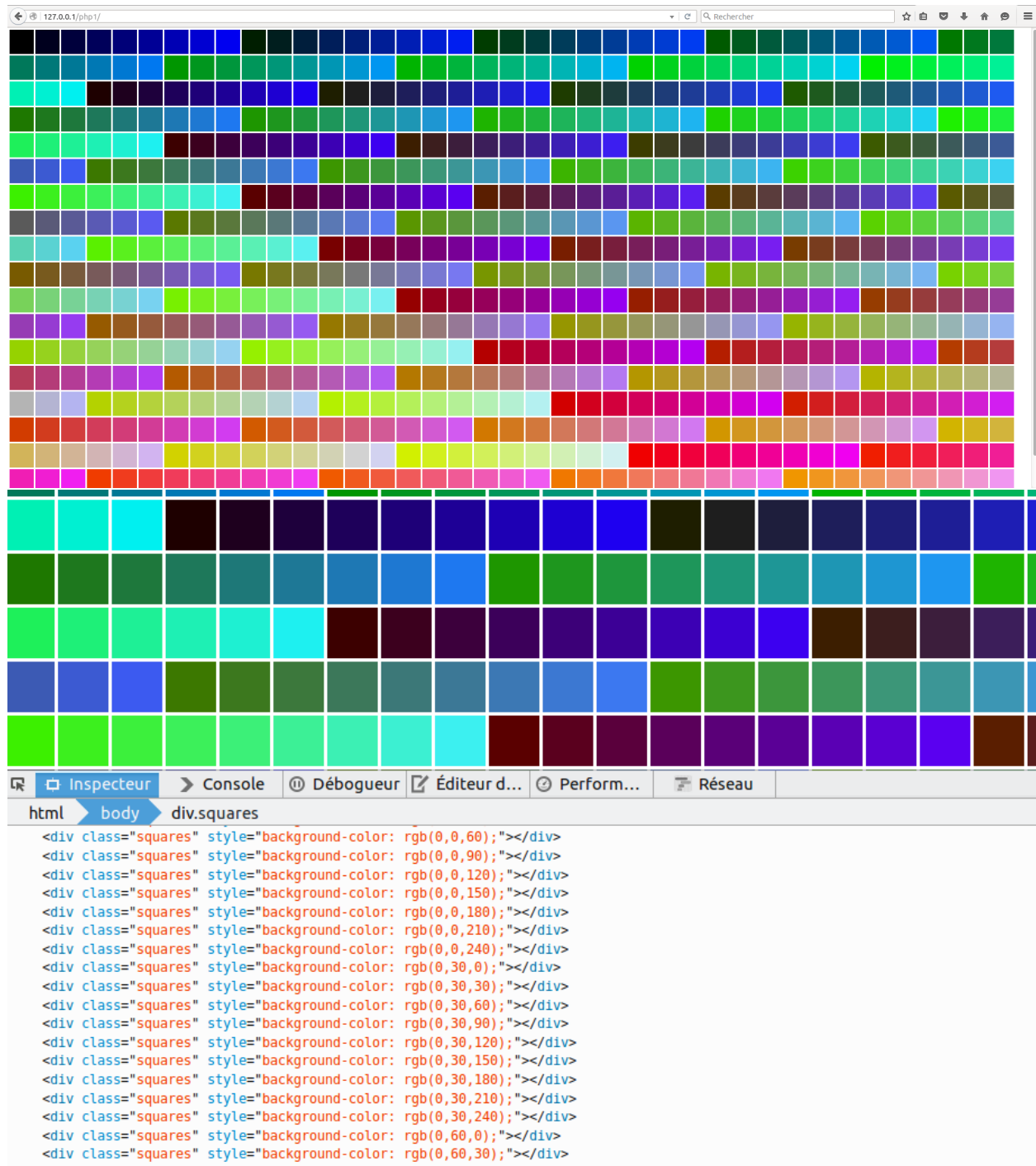
<?php

    for ($intRed=0; $intRed<=255; $intRed=$intRed+30)
    for ($intGreen=0; $intGreen<=255; $intGreen=$intGreen+30)
    for ($intBlue=0; $intBlue<=255; $intBlue=$intBlue+30) {
        $StrColor = "rgb(" . $intRed . "," . $intGreen . "," . $intBlue . ")";
        echo "<span style='color:" . $StrColor . ">" . $StrColor . "</span><br />";
    }
?>

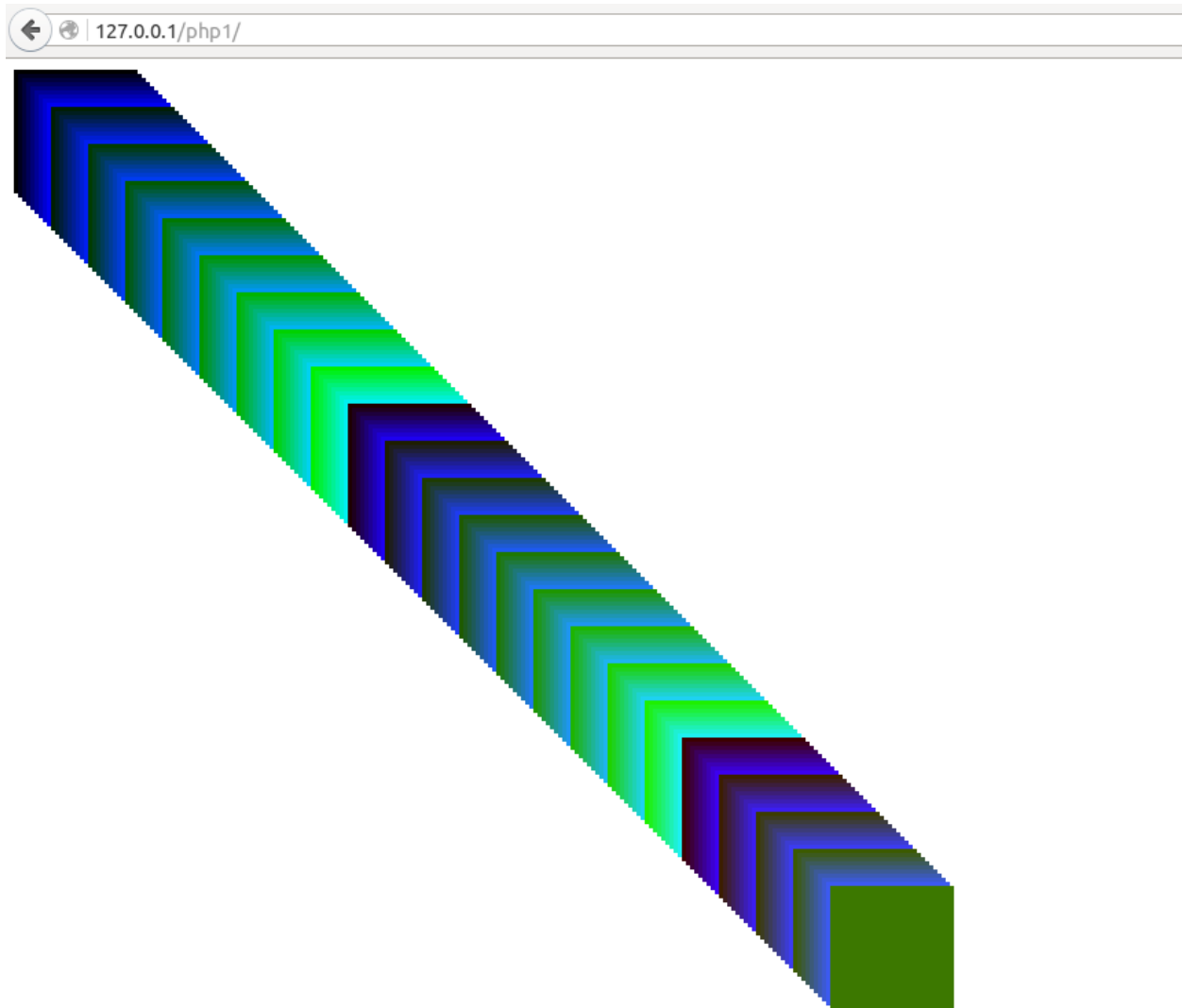
</body>
</html>
```

Modifier le script suivant afin qu'il affiche des petits carrés de couleurs au lieu des `rgb(x,y,z)` ... Faites au moins deux exemples.

Résultat 1 :



Résultat 2 :



Code complet :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Loops</title>
</head>
<body>
<style>
  .squares {

    width: 150px;
    height: 150px;
    /*display: block;*/
    float: left;
    margin: 5px;
```

```

    }

</style>

<?php
    for ($intRed=0; $intRed <= 255 ; $intRed=$intRed+30 )
        for ($intGreen=0; $intGreen <= 255 ; $intGreen=$intGreen+30 )
            for ($intBlue=0; $intBlue <= 255 ; $intBlue=$intBlue+30 )
                {
                    // Affichage par défaut :
                    // $StrColor ="rgb(" . $intRed . "," . $intGreen . "," . $intBlue . ")";
                    //echo "<span style='color:" . $StrColor . "'>" . $StrColor . "</span><br/>" ;

                    // Affichage 1 :
                    echo "<div class='squares' style='background-color: rgb(" . $intRed . ","
                        . $intGreen . "," . $intBlue . ")';></div>";

                    // Affichage 2 :
                    // echo "<span class='squares' style='background-color: rgb(" . $intRed
                    // . "," . $intGreen . "," . $intBlue . ")';></div>";
                }

    ?>

</body>
</html>

```

Exercice 3

Modifier le script précédent afin qu'il prenne un paramètre typeaff passé en GET permettant de choisir l'une ou l'autre des deux solutions proposées. Pour avoir accès à cette page vous tappez l'une des adresses suivantes dans votre navigateur : <http://@IP/~exercice2.php?typeaff=1> ou <http://@IP/~exercice2.php?typeaff=2>
...

Résultat :

Sans typeaff :

127.0.0.1/php1/exo2.php

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

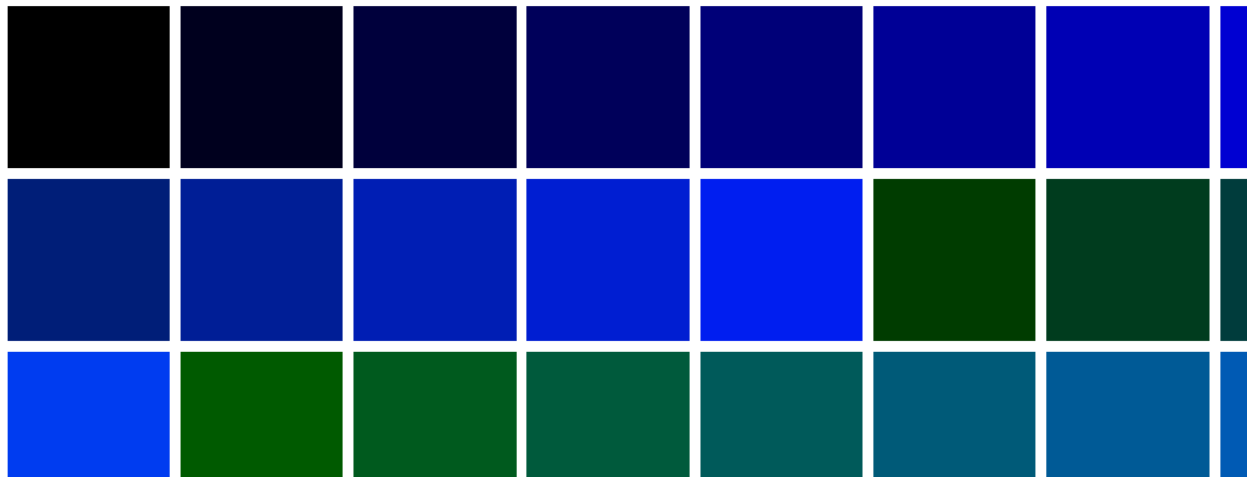
Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

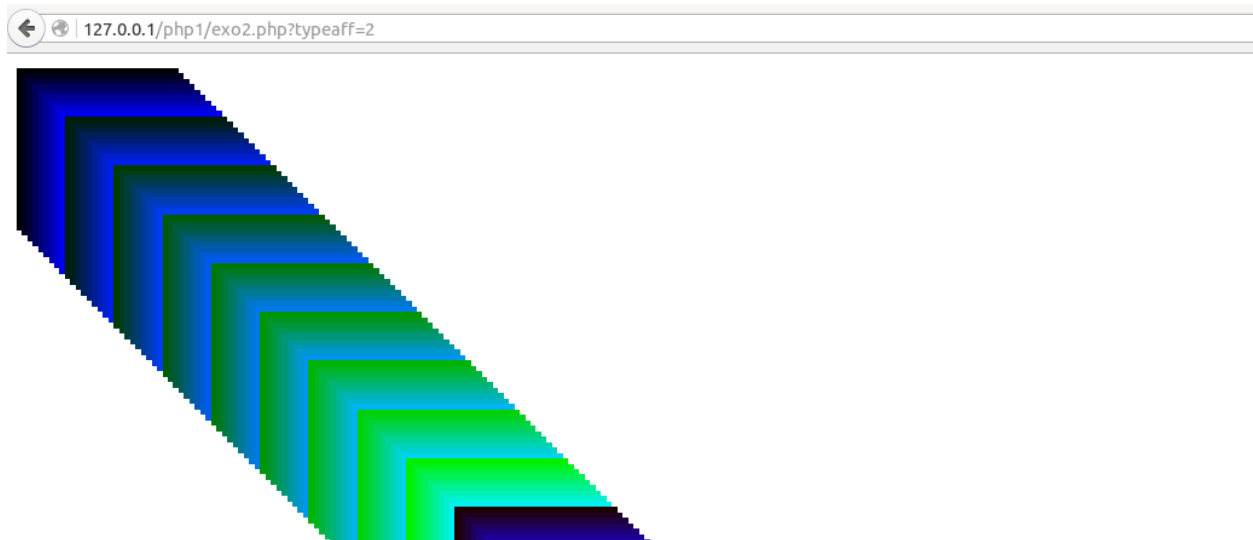
Notice: Undefined index: typeaff in `/var/www/html/php1/exo2.php` on line **31**

typeaff=1 :

127.0.0.1/php1/exo2.php?typeaff=1



typeaff=2 :



Snippet de code :

```
<?php

if ($_GET["typeaff"])
{
    if ($_GET["typeaff"] == 1)
    {
        // Affichage 1 :
        echo
        "<div class='squares' style='background-color: rgb("
        . $intRed . "," . $intGreen . "," . $intBlue . ");'>
        </div>";
    }

    else if ($_GET["typeaff"] == 2)
    {
        // Affichage 2 :
        echo
        "<span class='squares' style='background-color: rgb("
        . $intRed . "," . $intGreen . "," . $intBlue . ");'>
        </div>";
    }
}

?>
```

Exercice 4

La primitive `popen()` est bien utile car elle permet d'exécuter une commande et de récupérer son résultat dans le code qui l'a appelé.

Soit le code C suivant.

```

/* exemple_popen1.c */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#define NBCAR 128

int main (void)
{
    FILE * sortie;
    char ligne [NBCAR];
    char etat [NBCAR];

    if ((sortie = popen("/sbin/ifconfig eth0", "r")) == NULL) {
        fprintf(stderr, "Erreur popen %d \n", errno);
        exit(1);
    }

    while (fgets(ligne, NBCAR, sortie) != NULL) {
        printf("La chaîne : %s", ligne);

        /* %s dans sscanf() correspond à une séquence de
        caractères différents des caractères blancs :
        ici le 1er 'mot' de chaque ligne ! */
        if (sscanf(ligne, "%s", etat) == 1)
            /* printf("=>%s\n", etat); */
            if (strcmp(etat, "UP") == 0) {
                printf("Etat : %s", etat);
                fprintf(stderr, "interface eth0 en marche\n");
                pclose(sortie);
                return 0;
            }
    }

    fprintf(stdout, "interface eth0 inactive\n");
    pclose(sortie);
    return 0;
}

```

La primitive popen() existe aussi en PHP ! Réaliser un script PHP qui fait exactement la même chose que le code C ci-dessus.

Fonction is_eth(\$n) à développer

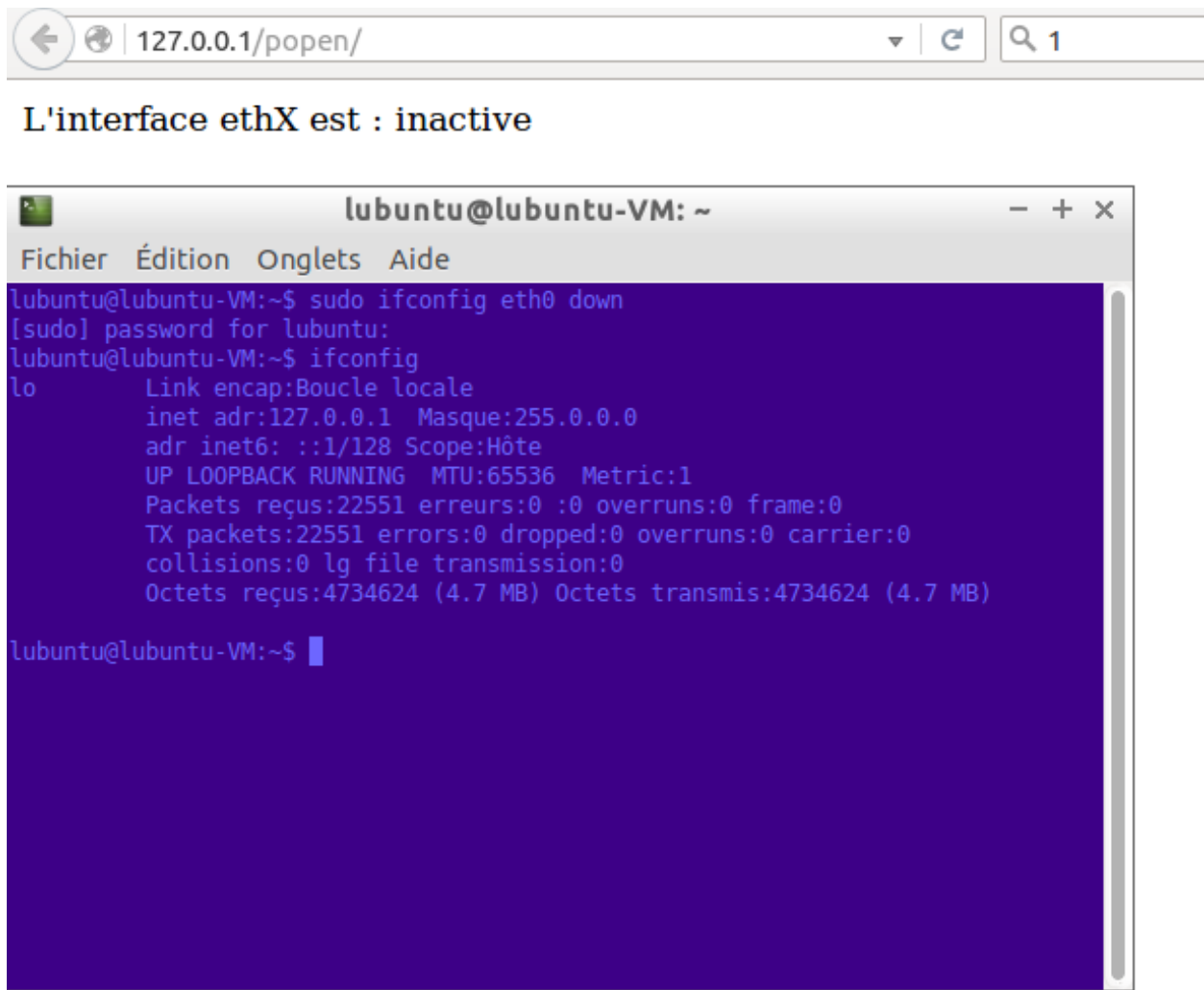
- Le script PHP comportera une fonction is_eth(\$n) qui prendra en paramètre le numéro de l'interface ;
- Cette fonction retournera TRUE ou FALSE suivant que l'interface est active ou non ;
- Vous appellerez la fonction et afficherez : L'interface ethX est : active ou inactive ;

L'opérateur ternaire !

Vous utiliserez avantageusement l'opérateur ternaire ! Pour l'affichage du résultat !

Résultat :

Sans l'interface eth0 :



Avec l'interface eth0 :

eth0 Link encap:Ethernet HWaddr 08:00:27:e9:23:c7

L'interface ethX est : active

```

lubuntu@lubuntu-VM:~$ sudo ifconfig eth0 up
lubuntu@lubuntu-VM:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:e9:23:c7
          inet adr:172.18.243.103  Bcast:172.18.247.255  Masque:255.255.248.0
          adr inet6: fe80::a00:27ff:fee9:23c7/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:354057 erreurs:0 :348 overruns:0 frame:0
          TX packets:47636 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:106370304 (106.3 MB) Octets transmis:3066982 (3.0 MB)

lo        Link encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          Packets reçus:22649 erreurs:0 :0 overruns:0 frame:0
          TX packets:22649 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:4745180 (4.7 MB) Octets transmis:4745180 (4.7 MB)

lubuntu@lubuntu-VM:~$

```

A partir du snippet suivant pris sur internet :

```

<?php
error_reporting(E_ALL);

/* Ajoute une redirection pour que vous puissiez lire stderr. */
$handle = popen('/sbin/ifconfig eth0', 'r');
echo "'$handle'; " . gettype($handle) . "\n";
$read = fread($handle, 2096);

echo "<center>",$read,"</center>";
pclose($handle);

is_eth($read);
?>

```

Code source :

```

<?php
$T="10"; //Temporisation

function is_eth($n)
{
    echo "L'interface ethX est : ";

    // Si $n est vide, afficher inactive sinon afficher active
    echo (empty($n)) ? " inactive\n" : " active\n";

    // if (empty($n))
    // {
    //     echo " inactive\n";
    // }
    // else
    // {
    //     echo " active\n";
    // }
}

/* Ajoute une redirection pour que vous puissiez lire stderr. */
// Rajout du "| grep" pour se focaliser sur une ligne contenant l'interface eth0
$handle = popen('/sbin/ifconfig | grep "eth0"', 'r');
// fread permet de lire le contenu du programme appelé (lu sur 2096 octets)
$read = fread($handle, 2096);
// On affiche "eth0 Link encap:Ethernet HWaddr XX:XX:XX:XX:XX:XX"
echo "<center>",$read,</center>";
// On ferme le processus popen()
pclose($handle);

//On appelle la fonction is_eth avec comme paramètre la ligne contenant eth0
is_eth($read);

//Rafraîchit tous les $T secondes (ici 10)
header("refresh: $T;");
?>

```

Exercice 5

Rajoutez au code PHP de l'exercice précédent la réactualisation de la page toutes les T secondes. En deux étapes

- Dans un premier temps vous coderez en dur dans le code une réactualisation toutes 5 secondes ;

```

<?php
$T="5"; //Temporisation à 5 secondes
...

```

```
//Rafraîchit tous les $T=5 secondes
header("refresh: $T;");
?>
```

Solution : • Puis vous paramétrez cette temporisation via une variable passée en GET. Par exemple <http://@IP/~exercice5.php?T=10> pour une réactualisation toutes les 10 secondes. Cependant, il faut penser au cas où l'argument n'est pas passé en GET et définir pour ce cas particulier une temporisation arbitraire de 5 secondes.

```
<?php
if (isset($_GET['T']) && $_GET['T'] > 2)
{
    $a = $_GET['T'];
}
else
{
    $a="5"; //Temporisation
}

...

header("refresh: $a;");
?>
```

Solution :

Exercice 6

Vous devez enregistrer dans une base de données les états de l'interface ethX. Ainsi que la date où l'état de celle-ci a été consultée.

Plusieurs étapes

- Il faut créer une BDD ;
- Il faut créer une table pour cette BDD ;
- Il vous faut créer une fonction de connexion à la BDD. Comportant des variables comme le login et le passord ainsi que l'adresse IP du serveur ;
- Il vous faudra créer également une fonction de déconnexion à la BDD ;
- Il vous faut une fonction d'insertion des valeurs dans une table ;
- Il vous faut une fonction de consultation des valeurs de la table ...

```
<?php
function connectDB($localhost,$login,$passwd,$dbb )
{
    // Cr  e la connection
    $conn = new mysqli($localhost, $login, $passwd, $dbb);
    // V  rifie la connection
    echo ($conn->connect_error) ? "Erreur de connection: "
        . $conn->connect_error . "\n" : "Connection r  ussie!\n";
    return $conn;
}

function insertDB($link,$ethx,$val,$date)
{
    //Lance requ  te SQL d'insertion dans BDD
    $sqli = "INSERT INTO ethx (etat, date_du_jour) VALUES ('".$ethx."+".$val."'
        , '".$date."')";

    //V  rifie la requ  te SQL
    if ($link->query($sqli) === TRUE) {
        echo "New record created successfully\n";
    } else {
        echo "Error: " . $sqli . "<br>" . $link->error;
    }
}

function affalinDB($link)
{
    //Lance requ  te SQL de s  lection dans BDD
    $sqlr = "SELECT id,etat,date_du_jour FROM ethx";
    $result = $link->query($sqlr);
    //Si au moins une ligne de r  sultats..
    if ($result->num_rows > 0) {
        //Cr  ation d'un tableau qui va permettre d'afficher les r  sultats
        echo "<table><tr><th>ID</th><th>ETH + VALEUR</th><th>DATE</th></tr>";

        while($row = $result->fetch_assoc()) {
            echo "<tr><td>".$row["id"]."</td><td>".$row["etat"]."</td><td>".
                $row["date_du_jour"]."</td></tr>";
        }
    } else {
        echo "0 results";
    }
}
}
```

```
//Ferme la connection
function closeDB($link)
{
    $link->close();
}

$link = connectDB("localhost","root","", "valsystem_db");
insertDB($link,"test","test","test");
affalinDB($link);
closeDB($link);

?>
```

Solution :

Connection réussie! New record created successfully

ID	ETH + VALEUR	DATE
1	test+test	0000-00-00 00:00:00
2	test+test	0000-00-00 00:00:00
3	test+test	0000-00-00 00:00:00
4	test+test	0000-00-00 00:00:00
5	test+test	0000-00-00 00:00:00
6	test+test	0000-00-00 00:00:00
7	test+test	0000-00-00 00:00:00
8	test+test	0000-00-00 00:00:00
9	test+test	0000-00-00 00:00:00
10	test+test	0000-00-00 00:00:00
11	test+test	0000-00-00 00:00:00
12	test+test	0000-00-00 00:00:00
13	test+test	0000-00-00 00:00:00