

# PHP\_IO

## Files Handling

C. BENSARI

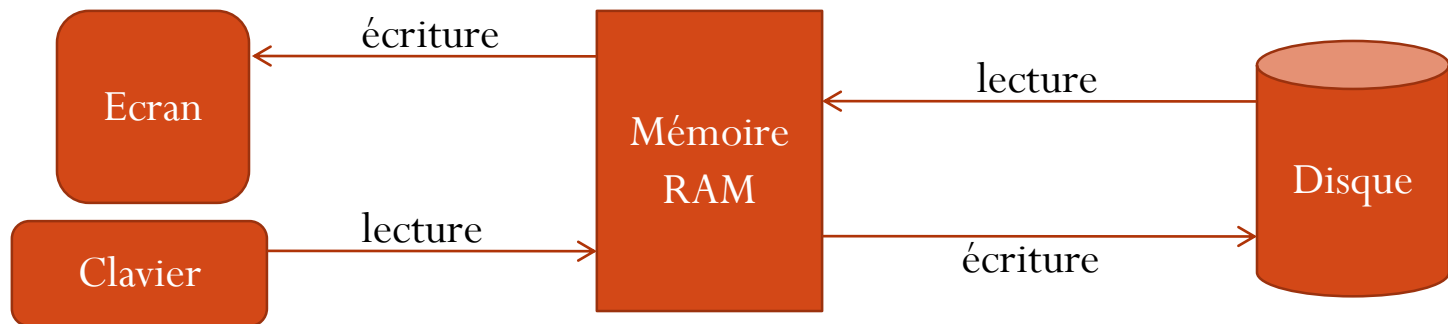
# Les fichiers

- Toutes structures de données vues jusqu'à présent en cours sont stockables sur un disque volatile de la machine (RAM). En mémoire volatile, la durée de vie d'une variable est égale au temps de l'exécution du programme
- Dans une application informatique, les données doivent être mémorisées plus longtemps d'où la nécessité d'utiliser des fichiers
- Un fichier est une structure de données destinée à contenir des informations homogènes. Il permet de stocker les informations sur un support non volatile (disques durs, ..) pour une durée indéterminée

# Les fichiers

## Lire et écrire dans un fichier

- De la même manière que l'échange entre l'utilisateur et le programme (lecture et écriture), il peut y avoir un échange entre la mémoire volatile et les disques :



- Un fichier peut être vu comme un ensemble d'enregistrements organisés d'une manière séquentielle les uns à la suite des autres

# Les fichiers en PHP

- PHP propose de nombreuses fonctions permettant de lire et écrire sur un fichier
  - **fopen**(string \$nom\_fichier, string \$mode) : une fonction permettant d'ouvrir un fichier et retourne un pointeur (handle) vers ce fichier.

Elle prend deux arguments, le premier est le nom (chemin) du fichier, le deuxième est le mode d'ouverture. Il existe plusieurs modes d'ouvertures, on trouve parmi ces modes:

- **r** : ouvre le fichier en lecture seule. Le curseur (pointeur) de lecture du fichier se positionne au début de fichier
- **w** : ouvre le fichier en écriture uniquement. Efface le contenu du fichier ou crée le fichier s'il n'existe pas. Le curseur de lecture se positionne au début du fichier
- **r+** : ouvre le fichier pour lecture et écriture. Le pointeur se positionne au début du fichier
- **x** : crée un nouveau fichier en écriture seule. Envoi false et une erreur si le fichier existe déjà. Place le curseur au début du fichier

# Les fichiers en PHP

Modes	Description
r	<b>Open a file for read only.</b> File pointer starts at the beginning of the file
w	<b>Open a file for write only.</b> Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a	<b>Open a file for write only.</b> The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x	<b>Creates a new file for write only.</b> Returns FALSE and an error if file already exists
r+	<b>Open a file for read/write.</b> File pointer starts at the beginning of the file
w+	<b>Open a file for read/write.</b> Erases the contents of the file or creates a new file if it doesn't exist. File pointer starts at the beginning of the file
a+	<b>Open a file for read/write.</b> The existing data in file is preserved. File pointer starts at the end of the file. Creates a new file if the file doesn't exist
x+	<b>Creates a new file for read/write.</b> Returns FALSE and an error if file already exists

# Les fichiers en PHP

- **fclose** (resource \$handle): permet de **fermer** un fichier pointé par le pointeur (handle) qui a été envoyé par la fonction **fopen**. La fonction envoie **true** si la fermeture a été effectuée avec succès, **false** sinon
- **fread**(resource \$handle, int length) : permet de lire octet par octet le volume (**length**) de données sur le fichier **handle** et le retourner. La fonction s'arrête lorsque length ou la fin du fichier est atteint(e)

```
<?php
// file opening
$myfile = fopen("test.txt", "r") or die("Unable to open file!");
// reading entire file
fread($myfile, filesize("test.txt"));
// close the file
fclose($myfile);
```

```
?>
```

# Les fichiers en PHP

- **fgets(resource \$handle)** : permet de lire une ligne depuis le handle sous forme de **chaîne de caractères**. La fonction passe systématiquement le pointeur à la ligne suivante.
- **Fgetc(resource \$handle)** : lit **caractère** par caractère depuis le pointeur. Après lecture, elle passe au caractère suivant
- **feof(resource \$handle)** : vérifie si le pointeur se trouve en fin de fichier. Si c'est le cas la fonction retourne **true** sinon **false**

```
<?php
$myfile = fopen("test.txt", "r") or die("Unable to open file!");
// Output one line until end-of-file
while(!feof($myfile)) {
    echo fgets($myfile) . "\n";
}
fclose($myfile);
```

```
?>
```

# Les fichiers en PHP

- **fwrite**(resource \$handle, string **\$content**) : permet d'écrire sur un fichier la chaîne de caractère **\$content**

```
<?php
    $myfile = fopen("test.txt", "w") or die("Unable to open file!");
    $fullName = "David DUPOND";
    fwrite($myfile, $fullName);
    $fullName = "Léa DUPOND";
    fwrite($myfile, $fullName);
    fclose($myfile);
?>
```