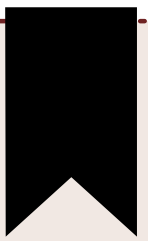


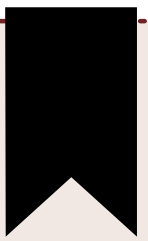
# Algorithmique en PHP



## Exercices PHP



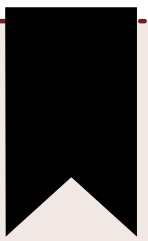
# But du jeu



- Traduire les algorithmes vus en cours
- Les faire tourner
- Éventuellement les modifier



# Pour commencer

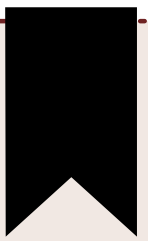


- Hello World !
  - Ouvrez un éditeur de texte
  - Insérez le code suivant :

```
<?php  
echo "Hello World!\n";  
?>
```
  - Sauvegardez le fichier dans votre dossier préféré sous le nom `helloWorld.php`
  - Remarques :
    - Un script PHP doit toujours commencer par `<?php` et finir par `?>`
    - Un retour à la ligne est codé par `"\n"`
    - Une instruction finit toujours par `;` (ou par un bloc `{ ... }`)



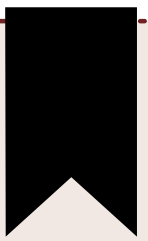
# Exécuter le script



- Ouvrez un *terminal*
- Déplacez-vous dans le dossier contenant votre script  
(les commandes unix `cd`, `pwd`, `ls`, la touche *<Tabulation>*, et les flèches Haut et Bas vous seront utiles)
- Pour l'exécutez, tapez la commande :  
`php -f helloWorld.php`



# Deuxième script



- Sauvegardez le code suivant dans le fichier date.php

```
<?php
```

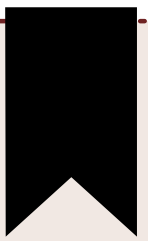
```
print "Il est ";    // écrit la chaîne entre guillemets  
echo date("H:i:s"); /* écrit l'heure courante donnée par  
la fonction prédéfinie date() dans le format spécifié */  
echo ", nous livrons de 18:00 à 22:00 !\n";
```

```
?>
```

- Exécutez-le !



# Lire les arguments (les entrées)



- Pour passer des arguments à un script (les fameuses entrées), on se sert du tableau d'arguments `$argv` de PHP
- Tapez le script `addition.php` suivant :

```
<?php
```

```
$a = $argv[1];
```

```
$b = $argv[2];
```

```
echo $a + $b . "\n";
```

```
?>
```

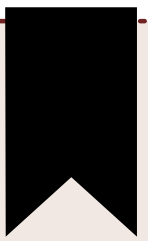
- Pour l'exécuter, tapez dans le terminal la commande suivante :

```
php -f addition.php 5 7
```

- Remarque :
  - Toutes les variables en PHP commencent par le caractère `$`
  - L'affectation utilise l'opérateur `=`
  - L'égalité l'opérateur `==`



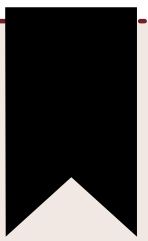
# Mini commande



- A la place d'interpréter un script (se trouvant dans un fichier), on peut vouloir exécuter une petite suite d'instructions directement
- `php -r 'echo $argv[1]+$argv[2]."\n";' 5 7`



# Enjeu de la programmation : la lisibilité



- Small-scale programming : “communicate with a computer”
- Large-scale programming : “communicate with other developers”
- Lisibilité :
  - 1 - utiliser des noms de variables *parlants* : un programme de thermodynamique sera plus clair si les variables s'appellent *enthalpie*, *energie\_libre* et *entropie* que *e1*, *e2*, *e3*.
  - 2 - Découper vos scripts en parties distinctes





# Découpage d'un script - 1

- Un script doit toujours comporter 3 parties :
  - 1 - Acquisition des paramètres du script et des données :
    - lecture de la ligne de commande (éventuellement test du bon nombre et de la validité des arguments)
    - affectation des variables

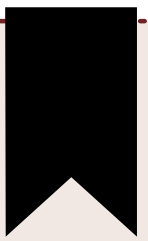
<?php

// acquisition des données

\$a = \$argv[1];

\$b = \$argv[2];

# Découpage d'un script - 2 & 3



- 2 - Algorithme
  - Calcul (que l'on peut découper en fonctions)
  - Stockage du résultat dans une structure appropriée (en mémoire)
  
- 3 - Présentation des résultats
  - Affichage du résultat (echo)
  - ou sauvegarde dans des fichiers

