

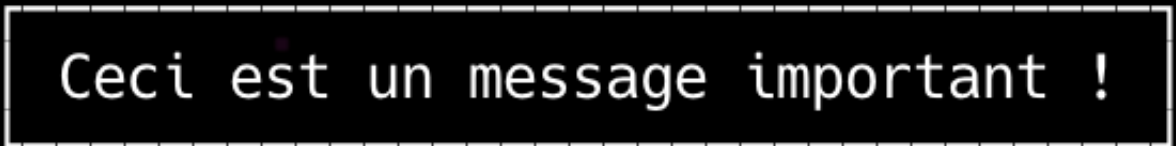
Dojo #1

Les fonctions

1. Créez un fichier `textbox.php`
2. Créez une fonction `displayTextInBox(string $text) : void` dans ce fichier
3. Déclarez une variable `$message` dans le fichier (à l'extérieur de la fonction) et initialisez là avec le texte de votre choix.
4. Toujours dans votre fichier, faites appel à votre fonction en lui passant en paramètre votre variable `$message`.
5. Testez votre code et travaillez-le jusqu'à obtenir le rendu suivant :

Note: vous pouvez utiliser les caractères suivants pour afficher les contours de votre box : `|` `_` `^` `~` `~` `^` `|`

→ `php textbox.php`



Ceci est un message important !

6. Félicitations, il est maintenant temps de refactoriser un peu notre code.

En effet, vous remarquez que le code pour dessiner les lignes du haut et du bas se répète.

Créez une fonction `drawLine(int $length, string $position) : void` qui va se charger d'afficher une ligne. Elle possède 2 paramètres :

- `$length` : la longueur de la ligne
- `$position` : soit `"up"`, soit `"down"` qui détermine la position de la barre pour pouvoir afficher les bons angles de la box en fonction

Faites appel à cette nouvelle fonction dans votre fonction `displayTextInBox`

Les arguments en ligne de commande

Bravo, vous avez un code propre et une jolie boîte pour afficher votre message. C'est bien mais il faut modifier votre script à chaque fois que vous souhaitez modifier votre message... et ça, c'est relou...

En PHP, il est possible de récupérer les arguments de la ligne de commande grâce à la variable `$argv`. Il s'agit d'un tableau (array) contenant autant d'éléments que d'arguments passés à votre script. Une autre variable `$argc` existe aussi et retourne le nombre d'arguments passés en paramètre.

Lorsque vous exécutez votre script de la manière suivante : `php textbox.php arg1 arg2` :

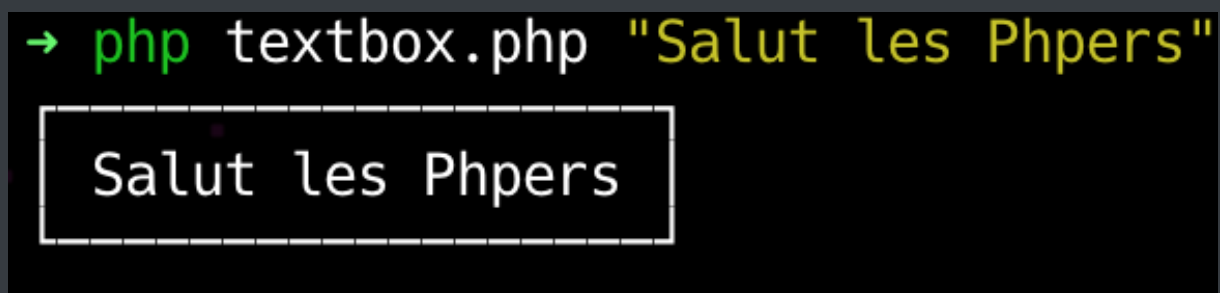
- `$argv[0]` vaut `string(11) "textbox.php"`
- `$argv[1]` vaut `string(4) "arg1"`
- `$argv[2]` vaut `string(4) "arg2"`
- `$argc` vaut `int(3)` // Attention, le nom du script est comptabilisé comme un argument (il s'agit en réalité d'un des arguments de la commande `php`)

Resources:

- <https://www.php.net/manual/en/reserved.variables.argv.php>
- <https://www.php.net/manual/en/reserved.variables argc.php>

1. Plutôt que d'utiliser le texte stocké jusqu'à présent dans votre variable `$message` faites en sorte d'utiliser l'argument passé à votre script en ligne de commande afin d'obtenir le même résultat que ci- dessous.

ex: `php textbox.php "Salut les Phpers"`

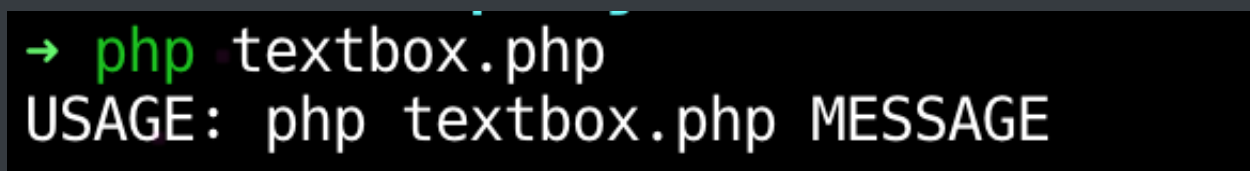


```
→ php textbox.php "Salut les Phpers"
Salut les Phpers
```

La gestion d'erreur

Cool vous pouvez désormais modifier votre message directement en ligne de commande ! Mais que se passe-t-il si vous ne mettez pas de message ? Il faut gérer ce cas d'erreur. C'est la première chose à faire quand on développe une fonctionnalité.

1. Si aucun message n'est passé en argument, faites en sorte d'afficher le message suivant :



```
→ php textbox.php
USAGE: php textbox.php MESSAGE
```

Note: pour sortir d'un programme PHP on utilise la fonction `exit()` : <https://www.php.net/manual/en/function.exit.php>

Jacky tunning

Si vous souhaitez aller plus loin, amusez vous avec les arguments de la ligne de commande et les paramètres de fonction pour pouvoir, par exemple :

- changer le style de la bordure dynamiquement
- ajouter un fond (plutôt que des espaces) dans votre box
- pouvoir afficher plusieurs messages
- ...
- Et pour les plus téméraires, créez un système de vote/sondage, avec la question et une liste de réponses qui s'affichent en dessous dans la box.

On pourrait alors afficher un message à l'utilisateur: "Merci d'entrer votre réponse : "et attendre qu'il entre un chiffre dans le terminal (1, 2 ou 3 en fonction de la réponse) puis afficher un petit message : "Merci pour votre réponse, vous avez voté "xxxxxx" :)

Pour ce faire, jetez un oeil à la fonction `readline` : <https://www.php.net/manual/en/function.readline.php>