

Vade-mecum output buffers

V1.0.0



Cette œuvre est mise à disposition selon les termes de la [licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Partage à l'Identique 3.0 non transposé](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Document en ligne : www.mickael-martin-nevot.com

1 Tampons de sortie

Il est possible de contrôler l'affichage en PHP grâce aux tampons de sortie (*output buffers*).

2 Démarrage du tampon

La fonction `ob_start()` enclenche la temporisation de sortie :

```
ob_start();  
echo 'J'écris dans le tampon !';
```

3 Vidage du tampon

La fonction `ob_end_flush()` vide le tampon, le retourne en tant que chaîne de caractère et stoppe la temporisation de sortie :

```
ob_start();  
echo 'J'écris dans le tampon !';  
ob_end_flush();
```

4 Récupération des données du tampon

4.1 Récupération avec affichage

La fonction `ob_get_contents()` retourne le contenu du tampon de sortie :

```
ob_start();  
echo 'J'écris dans le tampon !';  
$str = ob_get_contents();  
ob_end_flush();  
var_dump($str);
```

4.2 Récupération sans affichage

La fonction `ob_get_clean()` évite l'affichage, lit le contenu du tampon de sortie puis le vide :

```
ob_start();  
echo 'J'écris dans le tampon !';  
$str = ob_get_clean();
```

5 Empiler les tampons

5.1 Comportement de la pile

Les tampons sont empilables : il est possible d'en démarrer plusieurs à la suite :

```
ob_start();
echo 'Je suis dans le premier !';
ob_start();
echo 'Je suis dans le second !';
ob_end_clean();
```

Dans cet exemple, seul le premier tampon s'affiche. PHP gère une pile de tampons, ainsi le premier est démarré et empilé, suivi du second. L'appel de la fonction `ob_get_clean()` vide donc le second. Le premier n'est pas affecté car il est en-dessous dans la pile.

5.2 Vidage de tampons empilés

Dans l'exemple suivant, le second tampon est vidé dans le premier avec l'appel de la fonction `ob_end_flush()`. Le premier est ensuite de nouveau utilisé :

```
ob_start();
echo 'Je suis dans le premier !';
ob_start();
echo 'Je suis dans le second !';
ob_end_flush();
echo 'Je suis encore dans le premier !';
```

Voyons un nouvel exemple :

```
echo '1';           // « 1 » est affiché.
ob_start();        // Le premier tampon est démarré.
echo '2';          // 2 est écrit dans le premier tampon.
ob_start();        // Le second tampon est démarré.
echo '3';          // 3 est écrit dans le second tampon.
$one = ob_get_clean(); // $one = 3, et le second tampon est détruit.
echo '4';          // 4 est écrit dans le premier tampon.
$two = ob_get_clean(); // $two = 24, et le premier tampon est détruit.
echo '5';          // « 5 » est affiché.
echo $one and $two; // « 3 and 24 » est affiché.
```