

Hello World!

Je suis Mehdi « Arcanite » OUESLATI, et vous assistez à ma toute première GConfs. Comme c'est un moment important pour moi, et que la soirée des 1000 abonnés est importante pour l'asso, j'ai décidé de marquer le coup et de réaliser l'impossible : Je compte vous démontrer en 10 points que PHP est le meilleur langage de programmation du monde. Autant dire que je ne suis pas dans la merde.

Enfin bref. Tout ça est trivial, et je ferai mieux de me mettre au boulot. Pour vous démontrer que PHP est le meilleur langage du monde, je compte commencer par un axiome simple. Un axiome si évident que même le plus noob des sups l'accepterait sans sourciller. Et cet axiome est que : *\*pause\**  
**CAML, c'est de la merde.**

**Caml est merdique.** Tout le monde le sait, et tout le monde déteste ce langage sorti des tréfonds de l'enfer. Mais soyons scientifiques deux secondes et demandons-nous : Pourquoi est-ce que CAML est si détestable et détesté ? La réponse, simple, réside en ce côté strict qu'a CAML. Car CAML est strict. CAML est si strict que même l'opérateur « + » est strict. En CAML, vous ne pouvez même pas additionner un int et un float. Vous êtes obligés d'en convertir un, et si vous voulez additionner deux floats, vous devez utiliser un opérateur « +. » bizarre.

PHP, lui, est son opposé parfait. PHP, lui, s'en fout de ce que vous additionnez. Vous pouvez non seulement additionner deux floats ou un int et un float, mais même un float et un string ! Oui oui, un string. D'ailleurs quand vous additionnez un float et un string, PHP s'en fout tellement qu'il vous retourne un int. Sans rire, c'est pas beau ça ?

D'ailleurs, tant que j'y pense : PHP s'en fout tellement que vous pouvez lui donner n'importe quel type, il vous le comptera toujours comme un cast en int. « 123 » == « 000123fOo » output TRUE. Magique, hein ? Comment ça non ? Comment ça c'est une faille de sécurité ? Oui, bon... Ok, ça veut dire que si on compare un hash hexadécimal avec un nombre, on peut avoir un true. Mais bon ! Comme vous le dira n'importe quel SRS : « La sécurité, c'est surcoté ! »

Bref, notons ensemble : « Point positif numéro 1 de PHP : PHP s'en fout. »

Comme point positif numéro 2, j'aimerais considérer le fait que PHP a confiance en lui, car PHP a confiance en lui. PHP est un rebelle, un libre penseur, un révolutionnaire. PHP en a rien à foutre des conventions, il prend ses propres choix et les assume. Vous ne me croyez pas ? Voici un exemple :

```
<?php
$foo = (
    ( FALSE ) ? 'A' :
    ( TRUE ) ? 'B' :
    ( FALSE ) ? 'C' :
    ( FALSE ) ? 'D' : 'E' );

echo $foo;
?>
```

Allez y, je vous donne une minute pour réfléchir. Vraiment. OK. Levez la main : Qui pense que c'est la réponse 'A' ? Personne, normal. Qui pense que c'est la 'B' ? Ok... La 'C' ? Nice. Et qui dit que c'est la 'D' ? Et qui dit que c'est la réponse E ? D'accord.

Bah, désolé presque tout le monde, mais la bonne réponse est comme souvent la réponse 'D'. Vous ne me croyez pas ? Jugez par vous-même : « exécute le code »

Si vous voulez l'explication, en fait elle est simple : en PHP, l'opérateur ternaire est transitif par la gauche. Oui c'est bizarre. Oui, PHP est probablement le seul langage à avoir cet opérateur chelou. Mais comme je vous l'ai dit, PHP s'assume complètement. La page de documentation sur l'opérateur ternaire PHP est super bien faite et ce comportement semble souhaité.

D'ailleurs, quand je dis que PHP s'assume, je veux non seulement dire que PHP assume ses choix d'implémentation, mais aussi qu'il assume ses blagues. Le mot PHP lui-même n'a pas de sens. C'est un acronyme récursif pour « Hypertext Preprocessor », et les dévs ont donné ce nom justement pour le lolz. La blague étant que le nom est aussi cohérent que le langage. Vous voyez : en règle générale, quand vous codez dans un langage dit conventionnel, vous passez des heures à déprimer devant votre clavier jusqu'à ce que ce maudit `Runtime_Error` qui vous donne envie de mourir ? Bah PHP s'est dit qu'il allait vous donner envie de mourir... de rire ! L'interpréteur PHP contient des easter eggs. Par exemple, si vous mettez « :: » n'importe où dans votre code, vous aurez une erreur... En hébreux ! Oui, oui ! En hébreux ! L'erreur dit exactement le problème, qui est « double colon operator ». Et du coup en cas de typo, si vous ne parlez pas hébreux, vous êtes obligés de Google. Notons donc : « Point positif numéro 3 : PHP a le sens de l'humour »

En vrai, j'ai même envie de dire que PHP vous apprend les langues étrangères. Quoi que. Je fais ce que je veux en fait. Du coup, wai, je le mets. Plaiguez vous à GConfs pour les deadlines. « Point positif numéro 4 : PHP aime les langues étrangère »

D'ailleurs, non seulement PHP aime les langues étrangères, mais PHP dépend de langages étrangers. En fait, l'abstraction de PHP est buggée. Dans les langages de scripts normaux, les fonctions natives sont indépendantes des fonctions des langages sur lesquelles les scripts fonctionnent. Bien entendu, vous l'aurez compris, ce n'est pas le cas en PHP. Par exemple, la fonction « `money_format()` » ne marche pas sur Windows. Pourquoi ? Parceque PHP appelle la fonction C « `strfmon` » qui n'est pas disponible nativement sous Windows. Evidemment, je comprends que certains d'entre vous puissent penser que c'est une mauvaise nouvelle, mais c'est pas si mal comme fonctionnalité ! En vrai, c'est même génial ! Pour la première fois, vous pouvez dire « Mais ça marchait sur ma machine » sans perdre de crédibilité ». Notons donc : « Point positif numéro 5 : PHP vous protège en vous donnant accès aux excuses de noob ! »

Aussi, tant que j'y pense, je parlais tout à l'heure de debugging, mais ce n'est pas le seul soucis que vous pouvez rencontrer quand vous programmez. Non, vous avez choisi d'être un programmeur parceque vous êtes avant tout un artiste ! Vous êtes le poète du 21<sup>ème</sup> ! Votre code, c'est une œuvre d'art ! Et justement, PHP vous encourage votre créativité ! Et, pour ce faire, PHP a implémenter le concept de synonymie. Les codes suivants par exemple, font tous la même chose :

```
<?php
    echo "Hello World <br>";
?>

<?php
    echo("Hello World <br>");
?>

<?="Hello World <br>"?>
<?php
    print("Hello World <br>");
?>
<?php
    @System.@out.print("Hello World <br>");
?>

<?php
```

```
?> Hello World <br> <?php
```

```
?>
```

Rajoutons donc à notre liste : « Point positif numéro 6 : PHP vous encourage à être créatifs. »

D'ailleurs, à part la synonymie, PHP a plein d'autres trucs bizarres. Par exemple, si vous vous demandez ce que c'est que le « @ » juste ici, eh bien... C'est un opérateur unique à PHP qui permet de réduire les warnings au silence. Magie-Magie.

Pareil, en PHP on peut incrémenter des strings ! Non, vraiment. C'est pas une blague.

```
<?php
```

```
$foo = 'a';  
$foo++;  
echo $foo;  
echo '<br>';
```

```
$foo = 'a';  
$foo--;  
echo $foo;  
echo '<br>';
```

```
$foo = 'z';  
$foo++;  
echo $foo;  
echo '<br>';
```

```
$foo = 'ab123';  
$foo++;  
echo $foo;  
echo '<br>';
```

```
?>
```

J'attirerais particulièrement votre attention sur le fait que « z »++ égale « aa ». Je sais vraiment pas comment ça peut faire du sens, mais ça marche. Pour citer Sunbro : « C'est des incantations vaudous en PHP »

D'ailleurs c'est pire que ce que vous pensez, puisque si vous prenez ce bug et que vous l'utilisez avec celui qui fait que les strings sont toujours évaluées en int, ou encore un autre sur lequel je ne vais pas m'étendre qui fait que les class en PHP sont pratiquement des strings, ça vous fait des trucs qui sont carrément dangereux !

Mais bon, autant c'est dangereux, autant c'est trop putain de cool, alors ajoutons le à notre liste : « Point positif numéro 7: PHP est unique »

*\*pause\**

A ce stade, j'aimerais faire un petit aparté pour vous expliquer pourquoi PHP est à ce point unique. PHP est à la base construit sur une bibliothèque C servant à parser du HTML . Cette bibliothèque était appelée CGI pour Common Gateway Interface. A la base, Ramus Lerdorf, son créateur, voulait juste se coder un blog, mais le projet lui plaisait, et il a rajouter des fonctionnalités de scripts perls, le fait de pouvoir insérer du code PHP dans du HTML, et des interactions avec des bases de données. Il a ensuite mis le tout sur un usenet-group, et les gens ont commencé à s'amuser avec. Ça explique beaucoup pourquoi PHP est si... bizarre. C'est un langage fait par le web pour le web, et sa construction est totalement anarchique. C'est un peu le Twitch Plays Pokémon des langages.

Fun fact : PHP a été créé en 1995, mais n'a eu de standard officiels qu'en 2014, et ces standards sont plus des standards de facto qu'autre chose...

Vous savez, on pourrait faire une conférence entière sur l'histoire du PHP, du coup, je pense arrêter de m'étendre et fermer la parenthèse historique ici.

\*pause\*

Bon.

Fermons la parenthèse et reprenons.

Maintenant qu'on s'est bien marrés sur toutes ces singularités bizarres de PHP, je vais commencer à faire mon taff et à parler des bons cotés de PHP.

Déjà, je vais commencer par une de mes fonctionnalités préférées du langage : la meta-programmation.

En levant la main : Qui d'entre vous a déjà codé en PHP ? Ok, pas mal. Et qui d'entre vous connaît « \$\$ » ?

Ok.

Pour les autres, je vous présente une fonctionnalité trop cool en PHP... Roulement de tambour... le \$\$. Le \$\$ permet de faire des variables variables. Comme je me doute que ça ne vous aide pas plus que ça, voici un exemple simple :

```
<?php
    $a = "hi";
    $b = "a";
    echo $$b;
?>
```

L'output est « hi ».

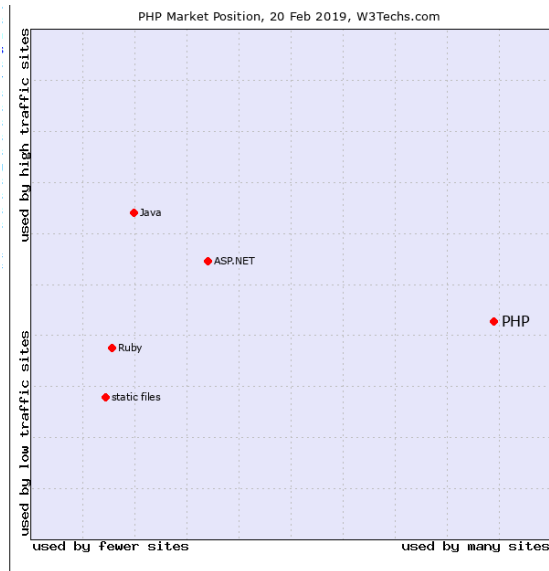
Du coup vous pouvez faire du scripting au sens le plus littéral du terme.

Pour les néophytes dans la salle, un script est supposé être un code servant à automatiser une tâche ; et, dans ce cas précis, on peut même scripter l'action de scripter. Scripto-ception. Rien que pour ça, PHP mérite un point de plus. « Numéro 8 : PHP permet de méta-programmer ».

Alors, souvent avec les variables variables, on se demande à quoi ça sert. Voici un exemple simple d'utilisation :

```
<?php
    $foo->$bar = 'stuff';
    $foo->$bar();
    $bar();
?>
```

Aussi, PHP a un atout majeur dans sa manche, qui est sa communauté immense, et qu'il représentait en 2016 82% de la part de marché des sites webs.



Evidemment, je pourrais sortir l'argument classique du PHP-fanboy lambda. A savoir que PHP est utilisé par Facebook et Wikipedia, mais Mr.Bashar avait utilisé un argument similaire pour justifier que l'AFIT soit en CAML, et je refuse d'être associé à CAML. Trauma de Sup, vous comprenez ?

Par contre, ce que je peux vous dire est que les défauts de PHP sont grandement corrigés à coup de frameworks comme Laravel ou Symfony. Qu'il y a des codes open sources de tout ce que vous voulez en PHP, et ce autant pour le développement WEB que pour le général purpose (même si, soyons honnête, utiliser PHP pour autre chose que du WEB, c'est sale.)

Vous n'aurez jamais de souci pour trouver de l'aide ou de la doc pour PHP, et il y aura toujours des tutos et des frameworks pour faire le boulot à votre place.

Notons donc : « PHP a une super bonne communauté ». Le corollaire est que PHP est pour les assistés, mais ne soyons pas mauvaise langue.

Un véritable point positif de PHP, par contre, est que c'est un excellent langage pour débiter. PHP est extrêmement facile à utiliser, notamment grâce à ses MAGIC\_CONSTANT, ou encore ses super globales comme `$_SESSION`, `$_POST`, `$_GET` et `$_SERVER` qui facilitent énormément la gestion de la back-end. Evidemment, la quasi-totalité des gens dans cette salle est capable de parser une requête http, mais le fait que ça soit possible ne veut pas dire que c'est souhaitable.

Evidemment, encore, on est des informaticiens, et on aime avoir le contrôle absolu, surtout quand on s'occupe de tâches liées au WEB, mais entre récupérer les données d'un user en écrivant appelant juste une super globale `$_POST`, et avoir à se taper tout le travail depuis le « `Socket socket = new Socket()` ; »...

Surtout qu'il faut bien comprendre que PHP est un excellent moyen de s'introduire à l'informatique. Un gamin de 13 ans qui débute sera beaucoup plus curieux et intéressé s'il peut coder des sites webs en récupérant juste un super global array `$_POST`, plutôt que s'il devait passer deux heures en JAVA à essayer de comprendre comment print un triangle sur console.

En bref : « Point positif numéro 10 : PHP est noob-friendly ».

En résumé,

PHP s'en fout

PHP a confiance en lui

PHP a le sens de l'humour

PHP aime les langues étrangère

PHP vous protège en vous donnant accès aux excuses de noob !

PHP vous encourage à être créatifs.

PHP est unique

PHP permet de méta-programmer

PHP a une super bonne communauté

PHP est noob-friendly

Du coup, PHP est objectivement un langage de merde. Mais il est utile, il remplit bien son rôle de langage utile pour des scripts rapides, et même si aujourd'hui tout le monde est complètement raide dingue des node.js event-driven functional trendy framework whatever.js, mais il faut bien comprendre que PHP a sa part de marché, et qu'elle ne semble pas baisser. Qu'on le veuille ou pas, les technologies inférieures, pourvues qu'elle puisse produire quelque chose de concret rapidement et simplement, seront toujours d'actualité, et PHP, en plus de remplir cette fonction, est un langage fait par le web, pour le web. Son côté dafuq vient justement de ce fait.

Mesdames, mesdemoiselles, messieurs, c'est sur ces belles paroles que je finis ma première conf. Si vous avez des questions, je vous en prie, posez les moi. Sinon, vous pouvez me contacter à [mehdi.oueslati@epita.fr](mailto:mehdi.oueslati@epita.fr), ou consulter mon linkedin ou mon github vide aux liens que vous voyez sur ce powerpoint.

A vous !

Qui a des questions ?