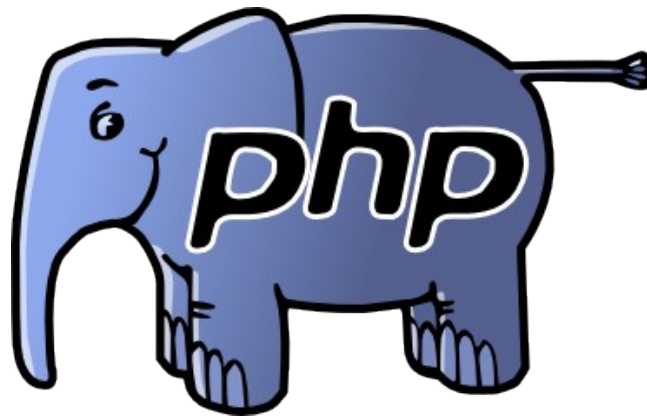


PHP

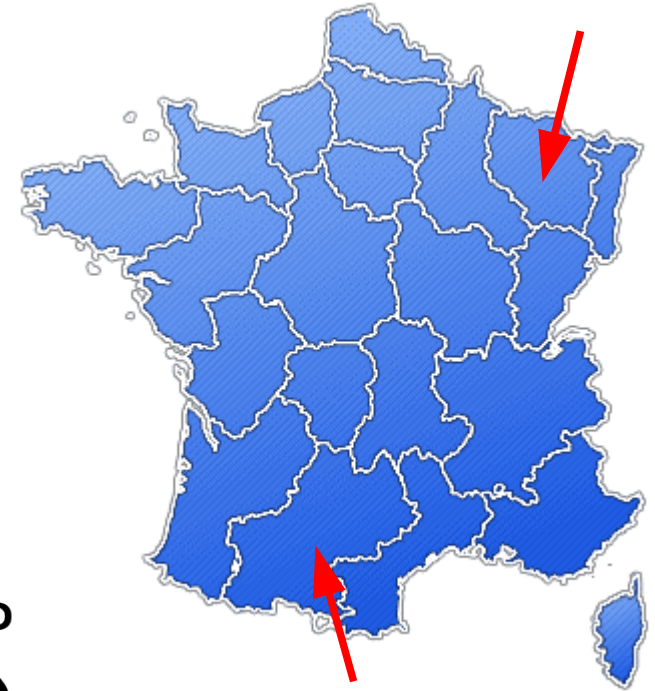
« Un langage pour le Web »



A propos

- Stéphane Gully
- Développeur Web (PHP!) à l'**INIST-CNRS**
- Fédération d'identités
- Framework, Statistiques, Outils collaboratifs ...
- DevelopR6, DevLog, Plume
- Perso :
 - Site : <http://www.zeitoun.net>
 - Twitter : [@kerphi79](#)

Contexte



Objectif

- Vous faire découvrir PHP
 - Histoire de PHP
 - PHP est orienté Web
 - La syntaxe de PHP
 - Richesse fonctionnelle de PHP
 - Points forts, points faibles de PHP
 - Les Framework PHP

Les pères fondateurs



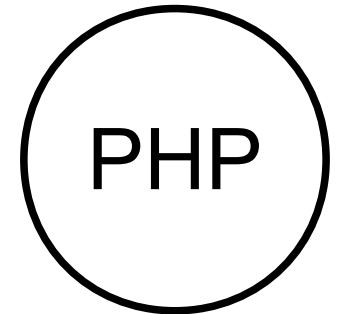
+



+



=

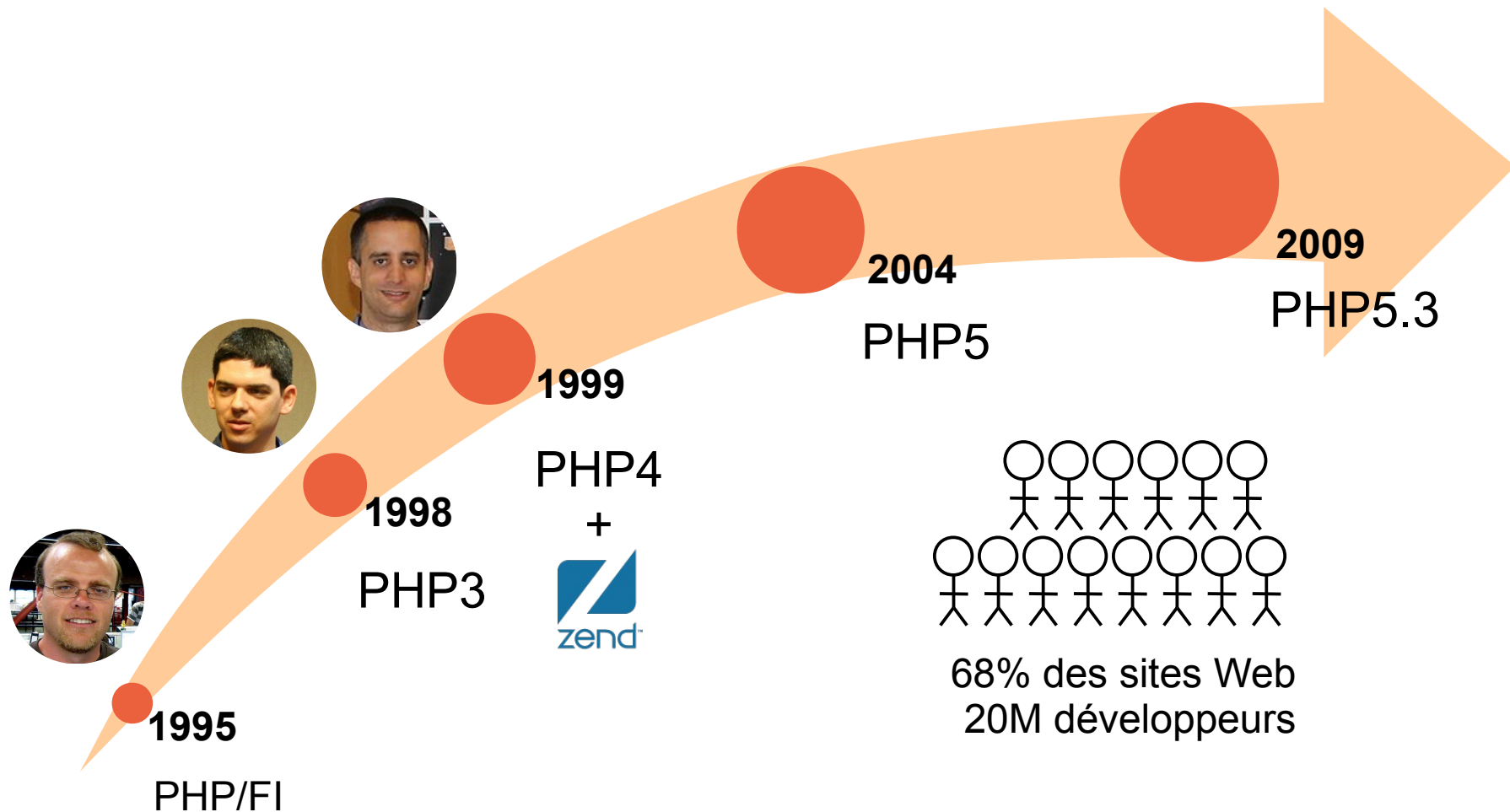


Rasmus Lerdorf

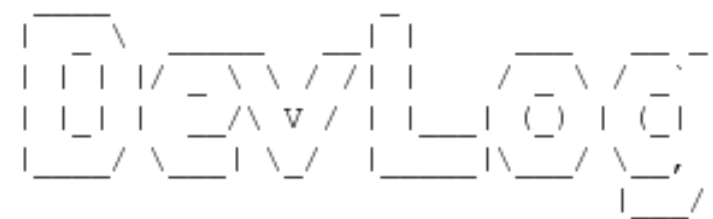
Andi Gutmans

Zeev Suraski

Histoire de PHP

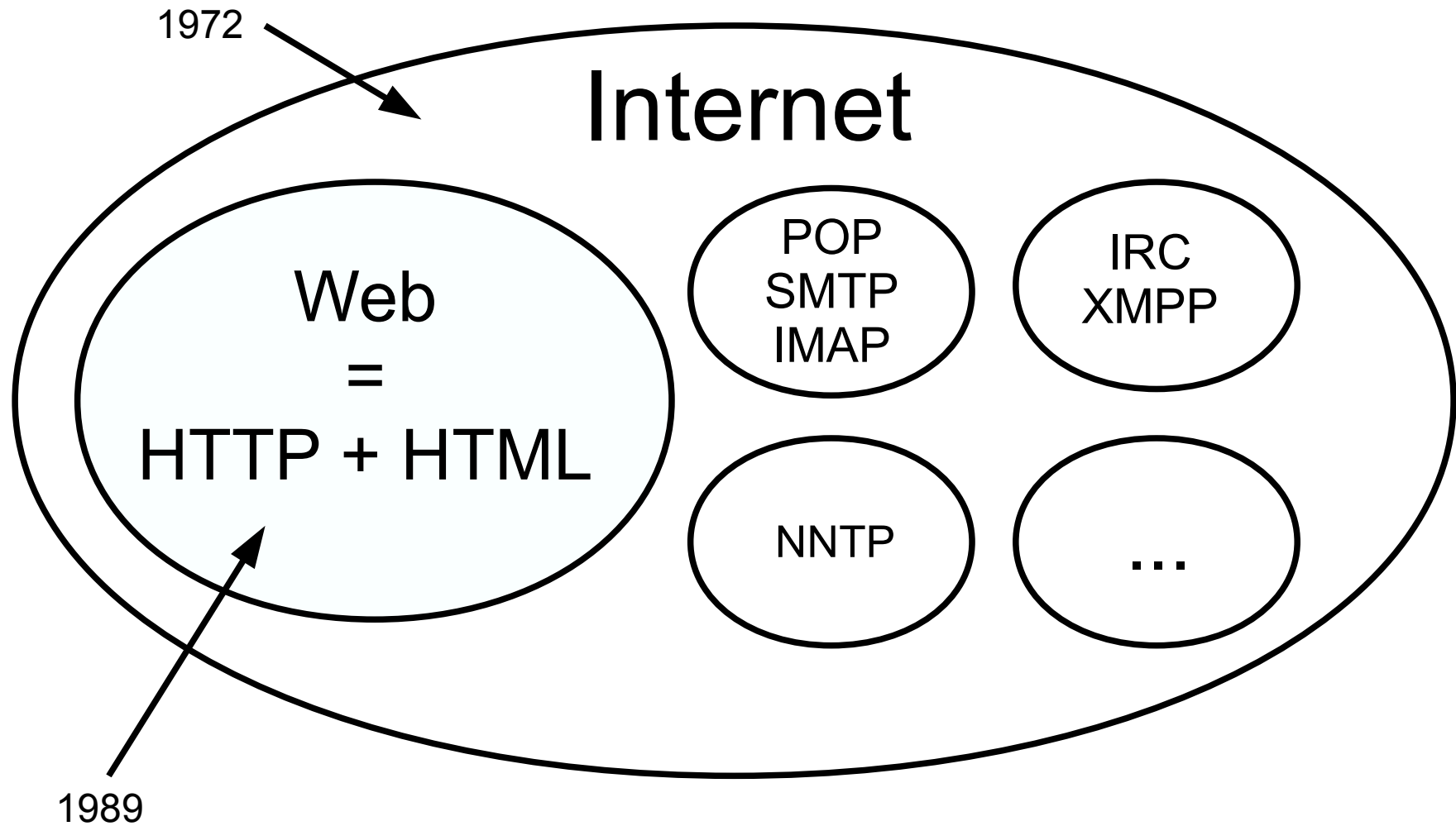


Qui utilise PHP ?

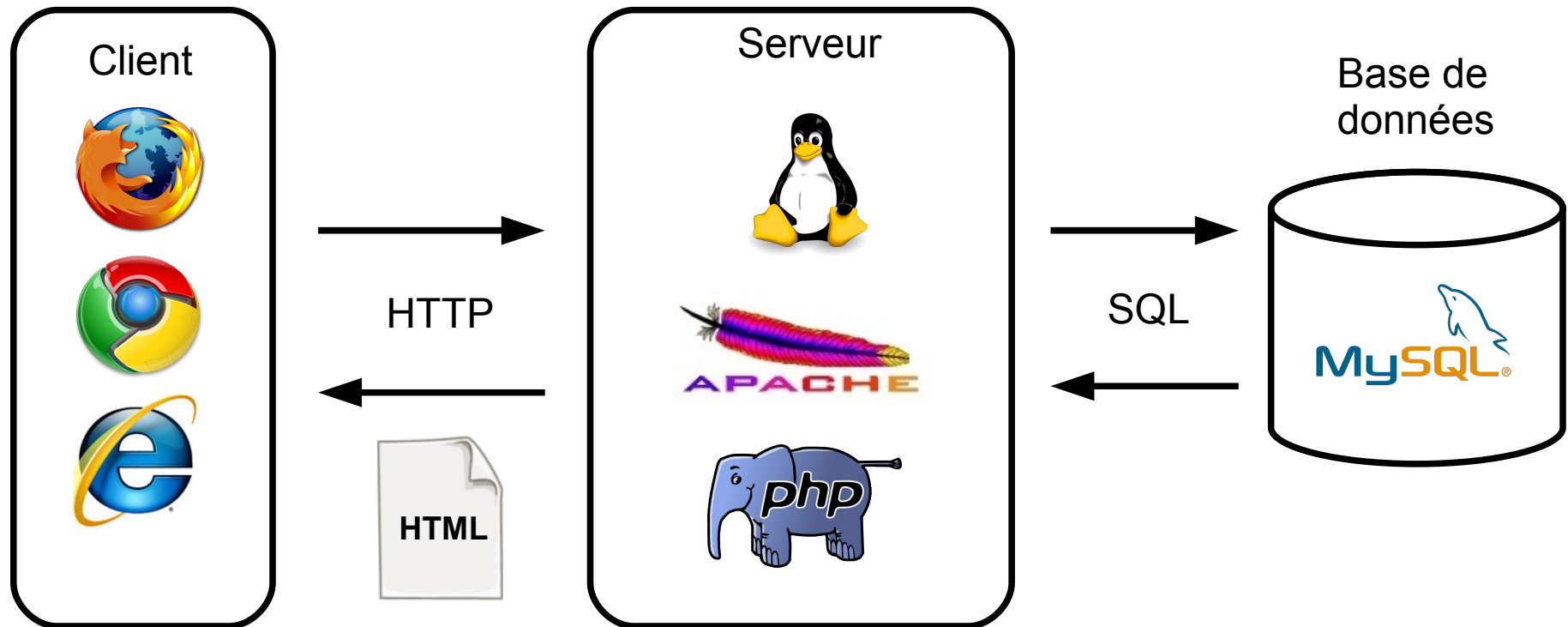


PHP est orienté Web

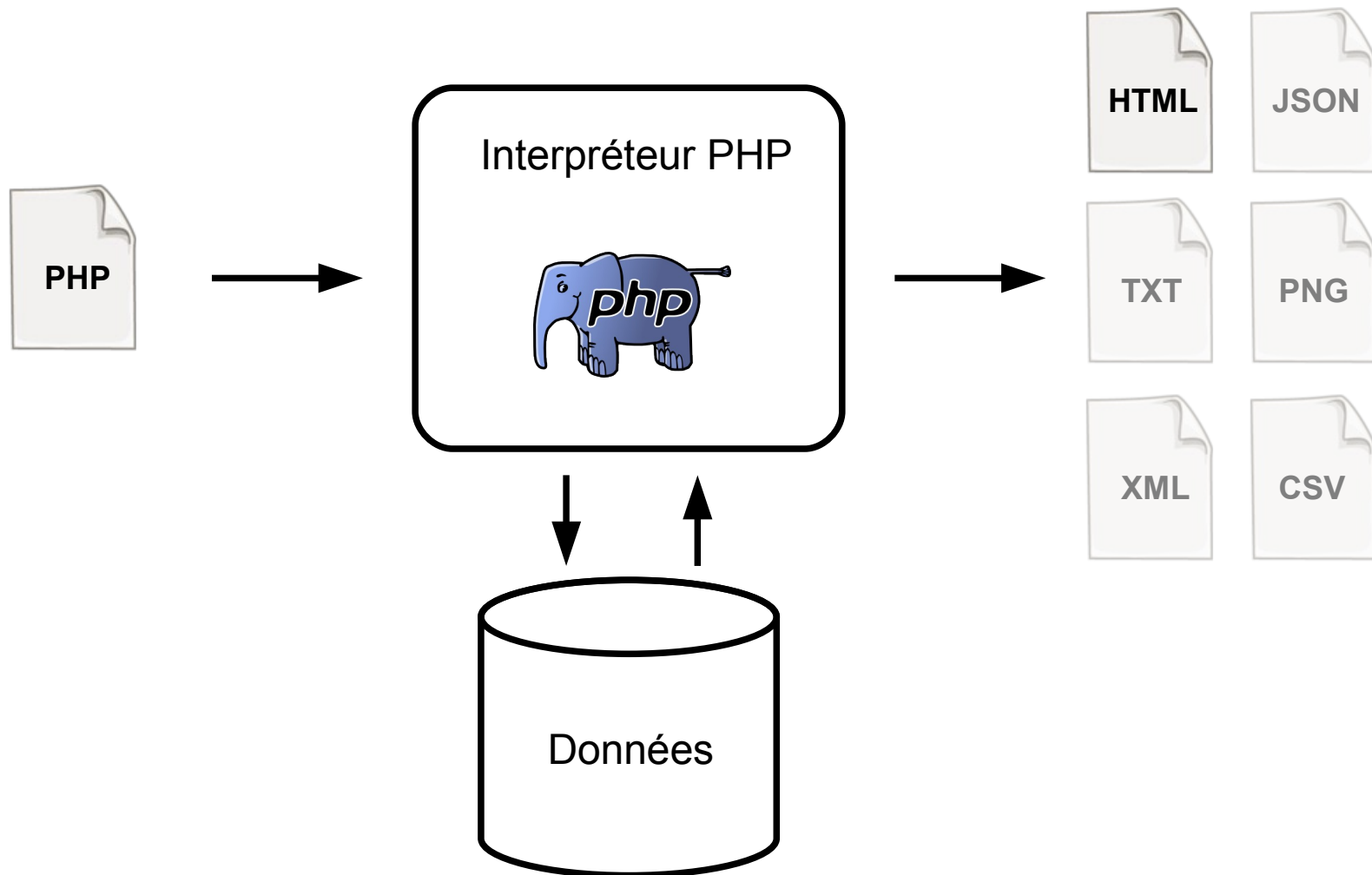
Le Web c'est quoi ?



LAMP



PHP n'est pas compilé



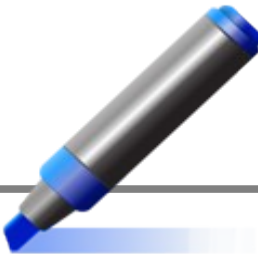
La syntaxe de PHP

Hello World

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

PHP intégré au HTML

```
...  
<p>Ceci sera ignoré par l'interpréteur PHP.</p>  
<?php echo 'Alors que ceci sera analysé par PHP.'; ?>  
<p>Ceci sera également ignoré.</p>  
...
```



PHP est souvent
utilisé comme
langage de template

Les variables

```
$prenom = 'Stéphane';  
$nom    = 'GULLY';  
var_dump($nom, $prenom);
```

```
string 'GULLY' (length=5)  
string 'Stéphane' (length=8)
```

Les tableaux

```
$fruits = array("orange", "prune", "pomme");
$panier = array("capacite" => 5,
               "contenu" => array("pomme", "orange"));

echo $fruits[0];           // orange
echo $fruits[2];          // pomme
echo count($panier["contenu"]); // 2
print_r($panier);
```

```
Array
(
    [capacite] => 5
    [contenu] => Array
        (
            [0] => pomme
            [1] => orange
        )
)
```


Structures de contrôle

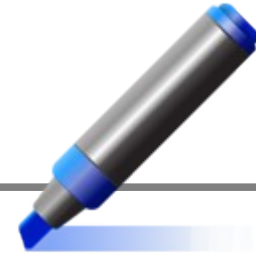
```
$a = rand(1,1000);  
$b = rand(1,1000);  
if ($a > $b) {  
    echo "$a est supérieur à $b";  
} else if ($b > $a) {  
    echo "$b est inférieur à $a";  
} else {  
    echo "$a et $b sont égaux";  
}
```

Les fonctions

```
function carre($n) {  
    return $n * $n;  
}  
  
echo carre(4);
```

Les fonctions anonymes

```
$algo = function($n) {  
    return $n * $n;  
};  
  
echo $algo(4);
```



≥ PHP5.3

Appelés également
« closures »

Pratique pour les
callbacks.

Les références

```
// deux variables partagent le même contenu
$a = 'data';
$b =& $a;
$b = 'data2';
echo $b; // data2
echo $a; // data2

// passage par référence dans une fonction
function mafonction(&$var) {
    $var .= 'xxx';
}
mafonction($a);
echo $b; // data2xxx
echo $a; // data2xxx
```

Les classes et les objets

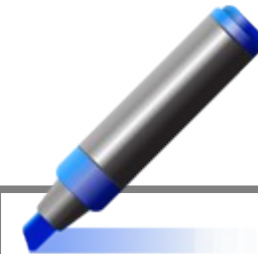
```
class Fruit
{
    protected $type      = null;
    protected $couleur   = null;
    protected $definition = null;

    public function __construct($type, $couleur) {
        $this->type      = $type;
        $this->couleur   = $couleur;
        $this->definition = rechercher_definition($type);
    }

    public function afficher() {
        echo $this->type.' '. $this->couleur.' : '. $this->definition;
    }

    public function __destruct() {
        unset($this->definition);
    }
}

$f = new Fruit('pomme', 'rouge');
$f->afficher(); // pomme rouge
```

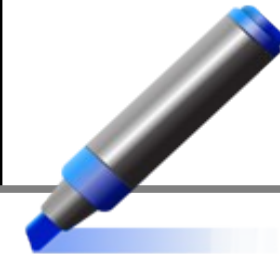


Full POO
depuis PHP5

- ◆ Héritage
- ◆ Interfaces
- ◆ Espaces de noms
- ◆ ...

Les exceptions

```
function inverse($x) {  
    if (!$x) {  
        throw new Exception('Division par zéro.');    }  
    else return 1/$x;  
}  
  
try {  
    echo inverse(5) . "\n";  
    echo inverse(0) . "\n";  
} catch (Exception $e) {  
    echo 'Exception reçue : ', $e->getMessage(), "\n";  
}
```



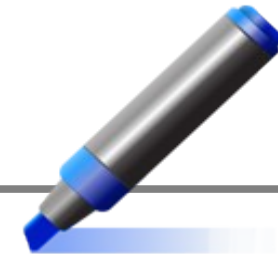
La gestion des erreurs
procédurale est
également possible :

```
trigger_error(...)
```

Sérialisation des données

```
// script 1
include_once 'Fruit.php';
$f = new Fruit('pomme', 'rouge');
$s = serialize($f);
file_put_contents('/tmp/store', $s);

// script 2
include_once 'Fruit.php';
$s = file_get_contents('/tmp/store');
$f = unserialize($s);
$f->afficher();
```



Pratique pour échanger
des données entre
scripts PHP.

Pensez aux méthodes
`__sleep()`
`__wakeup()`

Données venant du navigateur

```
<form method="post" action="script.php">  
  <input name="nom" type="text" value="alain" />  
  ...  
</form>
```

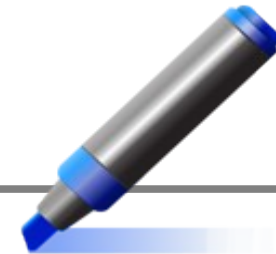
```
echo $_POST['nom']; // alain
```

```
http://monserveur/script.php?nom=alain
```

```
echo $_GET['nom']; // alain
```

```
// script1.php  
setcookie('nom', 'alain');
```

```
// script2.php  
echo $_COOKIE['nom']; // alain
```

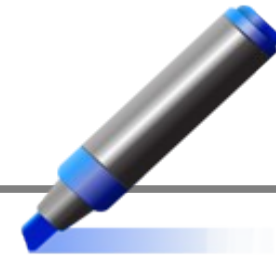


Astuce

\$_REQUEST contient
la fusion de
\$_GET, \$_POST
et \$_COOKIE

Les sessions

```
session_start();  
if (!isset($_SESSION['count'])) {  
    $_SESSION['count'] = 0;  
} else {  
    $_SESSION['count']++;  
}
```



Les données sont stockées coté serveur.

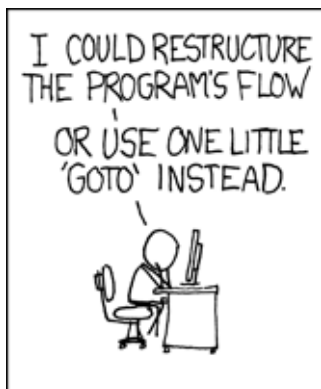
Pour placer des données coté client, utilisez
\$_COOKIE

Le retour du goto !

```
goto a;  
echo 'Foo';  
  
a:  
echo 'Bar';
```

≥ PHP5.3

Attention au
retour du code
Spaghetti !



source image [xkcd](#)

La richesse fonctionnelle de PHP

Traitement du texte

```
$chaine = '11h30';
preg_match('/^([0-9]+)h([0-9]+)$/ ', $chaine, $matches);
print_r($matches);
// Array
// (
//     [0] => 11h30
//     [1] => 11
//     [2] => 30
// )

$result = explode('h', $chaine);
print_r($result);
// Array
// (
//     [0] => 11
//     [1] => 30
// )

echo implode('h', $result);
// 11h30
```

Accès aux bases de données

```
try {
    $dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
    foreach($dbh->query('SELECT * from FOO') as $row) {
        print_r($row);
    }
    $dbh = null;
} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage() . "<br/>";
    die();
}
```



PDO supporte
actuellement :

MySQL
MSSQL
Firebird
Informix
Oracle
ODBC
PostgreSQL
SQLite
4D

Traitements XML

```
$string = <<<XML
<?xml version="1.0" ?>
<langages>
  <php>
    <titre>PHP: Hypertext Preprocessor</titre>
    <auteur>Rasmus Lerdorf</auteur>
  </php>
</langages>
XML;

$xml = simplexml_load_string($string);
echo $xml->php->titre; // PHP: Hypertext Preprocessor
```



Liste des API :

- SimpleXML
- DOM
- XMLReader
- XMLWriter
- Libxml
- Qtdom
- SDO
- WDDX
- XSL

Communication avec JavaScript

```
$msg = array (  
    'date'           => date('Y-m-d H:i:s'),  
    'destination' => '#lacafet',  
    'message'       => 'Salut !',  
);  
echo json_encode($msg);  
  
// {"date":"2010-11-17 16:52:25","destination":"#lacafet","message":"Salut !"}
```

Traitement des images

```
header("Content-type: image/png");  
$string = $_GET['text'];  
$im      = imagecreatefrompng("images/button1.png");  
$orange = imagecolorallocate($im, 220, 210, 60);  
$px      = (imagesx($im) - 7.5 * strlen($string)) / 2;  
imagestring($im, 3, $px, 9, $string, $orange);  
imagepng($im);  
imagedestroy($im);
```



Liste des API :

GD
ImageMagick
Exif
Gmagick
Cairo

Mais c'est pas tout...

- PHP pour l'administrateur système ?
getopt, ssh, ftp, svn, curl, posix, fichiers, ldap ...
- *Services Web*
OAuth, SOAP, XML-RPC
- *Manipulation audio*
OpenAL, ID3, ogg Vorbis
- *Manipulation des formats de compression*
Zip, Zlib, Rar, LZF, Bzip2, Phar
- *Manipulation des dates, heures, fuseaux*
- ...

PHP

points forts

points faibles

Points forts de PHP

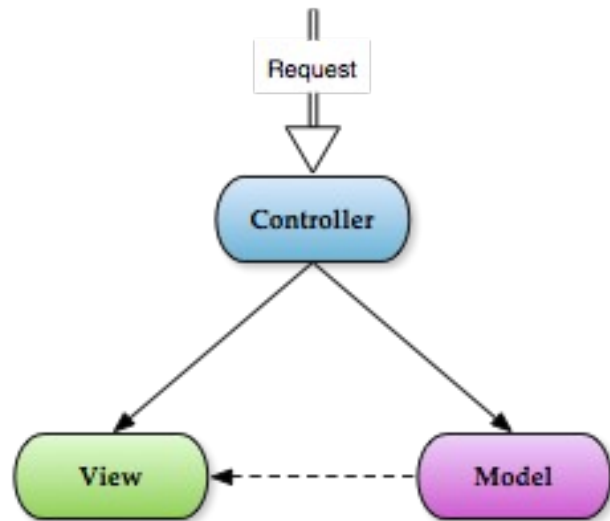
- Énorme communauté
- Excellente documentation
- Très faible courbe d'apprentissage
- Grande richesse fonctionnelle native
- Performances
APC, eAccelerator, Zend Platform
- Simplicité de déploiement
mod_php, wamp, easyphp, [pkgi...](#)
- *Cadres de dev très professionnels*
Zend Framework, Symfony, PEAR/PECL ...

Points faibles de PHP

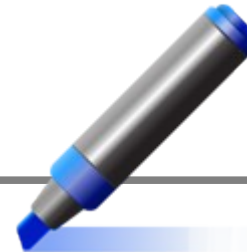
- Socle non homogène parfois incohérent
- Niveau des développeurs hétérogène
- Qualité de code produit hétérogène
- Parfois une mauvaise image
« PHP, un langage de bidouilleur pour faire son petit site oueb dans son garage »
- Peu adapté pour les longs traitements
- Debug natif rudimentaire
echo, var_dump, print_r

Zoom sur les Framework PHP

Les grands Framework PHP



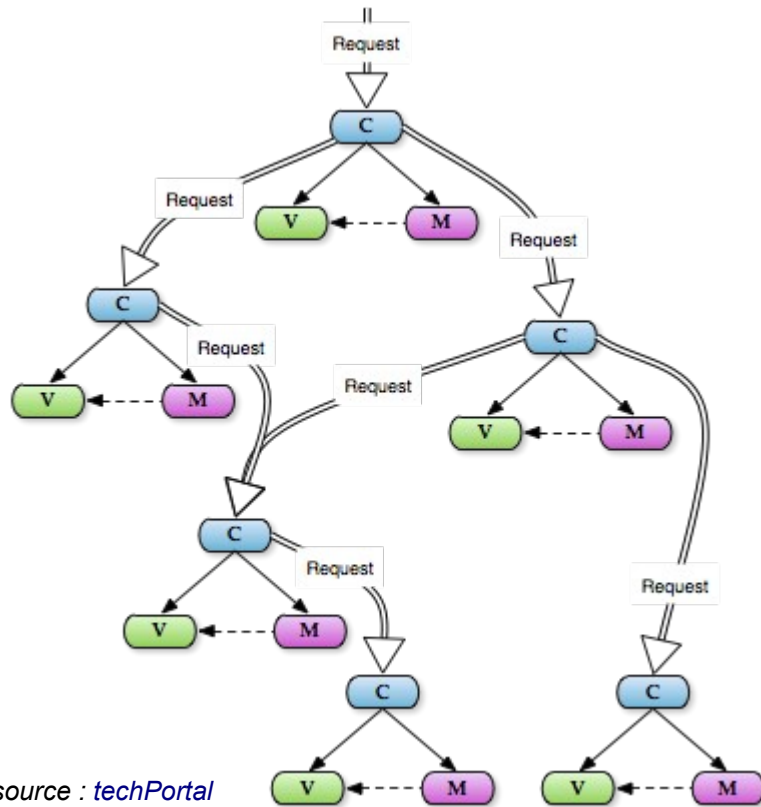
source : [techPortal](#)



Utilisent le patron de conception
« **Model-View-Controller** »
(MVC)



Le Framework Pxxo



Pxxo utilise le patron de conception
« **Hierarchical-Model-View-Controller** »
(HMVC)



Pxxo est un Logiciel Libre

Pxxo est utilisé dans les développements Web de l'[INIST-CNRS](#)

<http://www.pxxo.net>

Exemple d'utilisation de Pxxo

DEVELOP DR6 

Éditer cette page | Déconnexion | Mon compte | Accès réservé | Administration | Rechercher

Vous êtes ici: Le réseau DevelopR6 » Administration DevelopR6 » Administration des groupes

Administration des groupes

Liste des groupes

- admin ✕
- comite-pilotage ✕
- user ✕

Créer le groupe

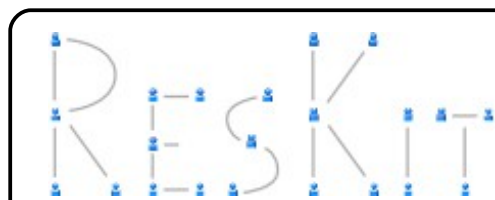
Liste des utilisateurs du groupe admin

- stephane.gully@inist.fr ✕
- nicolas.thouvenin@inist.fr ✕

st

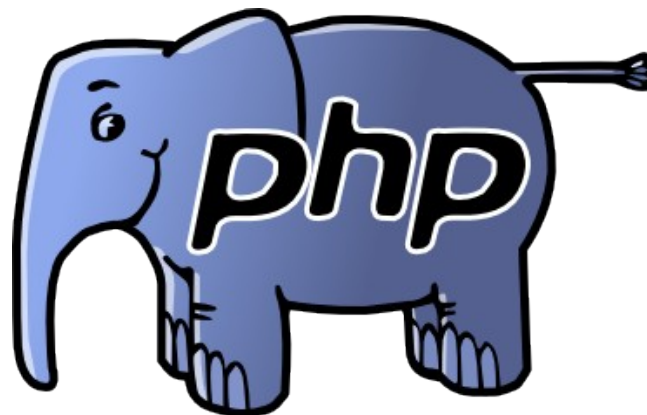
- stefan.gaget@good.ibl.fr
- stephane.gully@inist.fr
- stephane.schneider@inist.fr
- stephane.vialle@lpmi.uhp-nancy.fr

Widgets
Pxxo



ResKit est un Logiciel Libre
ResKit est utilisé par [DevelopR6](#), [DevLog](#), ...
<http://developr6.dr6.cnrs.fr/reskit/>

Dokuwiki



Merci, des questions ?

Stéphane Gully
(stephane.gully@inist.fr)