

Langage PHP

Numérique & Science Informatique



Langage PHP

Numérique & Science Informatique



« I did not develop the PHP we know today. Dozens, if not hundreds of people, developed PHP. I was simply the first developer. »

Rasmus Lerdorf

Langage PHP

Numérique & Science Informatique



Coût en ressources

	Energy		Time		Mb
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Go	2.83	(c) Ada	1.47
(v) Lisp	2.27	(c) Pascal	3.02	(c) Rust	1.54
(c) Ocaml	2.40	(c) Ocaml	3.09	(v) Lisp	1.92
(c) Fortran	2.52	(v) C#	3.14	(c) Haskell	2.45
(c) Swift	2.79	(v) Lisp	3.40	(i) PHP	2.57
(c) Haskell	3.10	(c) Haskell	3.55	(c) Swift	2.71
(v) C#	3.14	(c) Swift	4.20	(i) Python	2.80
(c) Go	3.23	(c) Fortran	4.20	(c) Ocaml	2.82
(i) JavaScript	4.45	(i) JavaScript	6.52	(v) C#	2.85
(i) PHP	29.30	(i) PHP	27.64	(i) Ruby	3.97
(v) Erlang	42.23	(v) Erlang	36.71	(i) JavaScript	4.59
(i) Lua	45.98	(i) Ruby	59.34	(v) Java	6.01
(i) Ruby	69.91	(i) Perl	65.79	(i) Perl	6.62
(i) Python	75.88	(i) Python	71.90	(i) Lua	6.72
(i) Perl	79.58	(i) Lua	82.91	(v) Erlang	7.20

Langage PHP

Numérique & Science Informatique



Langage PHP*

- Langage de programmation libre
- Interprété, impératif, procédural, orienté objet
- Langage de script côté serveur
- Créé en 1994 par Rasmus Lerdorf
- Version 8 en mars 2021



*Personnal Home Page -> PHP: Hypertext Preprocessor

Langage PHP

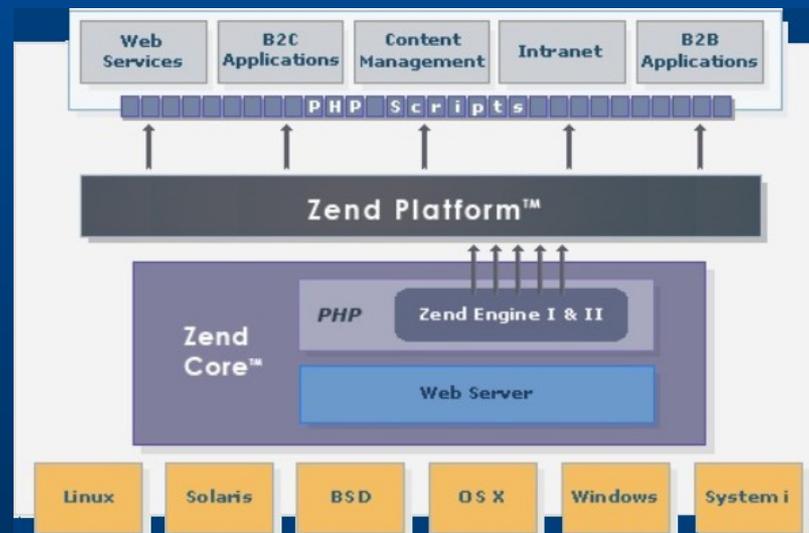
Numérique & Science Informatique



Accélération

Un langage interprété doit être compilé à la volée

- compilation en interne du code PHP en bytecode
- exécution par une machine virtuelle



Langage PHP

Numérique & Science Informatique



Déclaration de type en PHP

Typage dynamique et faible*

Scalaires :

`int`, `integer` : entier

`float`, `double` : réel

`string` : texte

`bool`, `boolean` : booléen

Composés :

`array` : tableau

`object` : instance

`callable` : fonction de rappel

`iterable` : itérateur

Langage PHP

Numérique & Science Informatique



Les tableaux

■ Tableau simple :

```
$a = [1, 2, 3, 4];  
$b = array(1, 2, 3, 4);
```

■ Tableau associatif :

```
$map = array(  
    'version' => 4,  
    'OS'      => 'Linux',  
    'lang'    => 'english',  
    'tags'    => true,  
    'version' => 5, // la valeur 4 sera écrasée par la valeur 5  
    3.14     // clé = 0  
);
```

Langage PHP

Numérique & Science Informatique



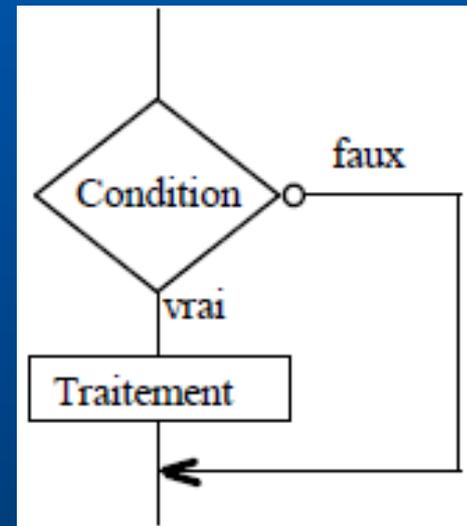
L'instruction conditionnelle

si <expression logique (vraie)> **alors**
 Traitement_vrai

```
if ( condition ) {  
    Traitement();  
}
```

La condition est de type booléen

Les instructions à exécuter sont indiquées par les accolades



Langage PHP

Numérique & Science Informatique



Exemple d'instructions conditionnelles

Écrire un programme qui indique si une année est bissextile ou non.

Une année est bissextile si elle est :

- divisible par 400
- ou divisible par 4 mais pas par 100

Langage PHP

Numérique & Science Informatique



Implémentation en PHP

```
function is_bissextile(int $year) : bool
{
    /** détermine si une année est bissextile
    @param      $year -- année ≥ 1582 (calendrier grégorien)
    @return     true si année bissextile, false sinon
    */
    if ( (is_int($year) == false) || ($year < 1582) )
        return false;

    if ( $year % 400 == 0 )
        return true;
    else
        if ( ($year % 4 == 0) && ($year % 100 != 0) )
            return true;

    return false;
}
```

Langage PHP

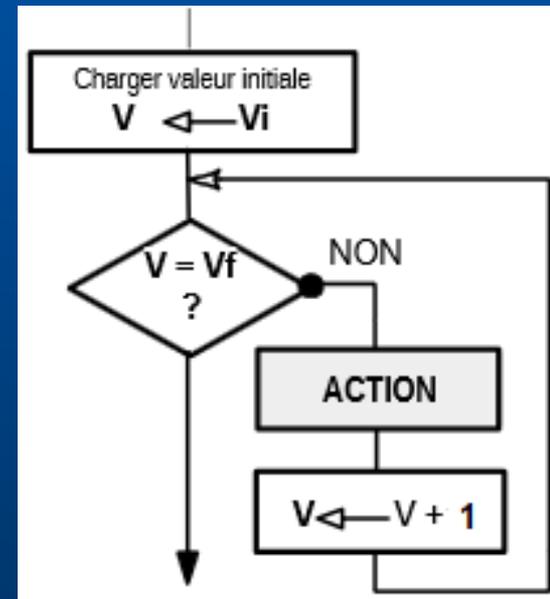
Numérique & Science Informatique



Boucle FOR : nombre d'itérations **connu**

pour <variable> **de** <valeur_initiale> **à** <valeur_finale> **faire**
Action(s)

```
for ($i = $Vi; $i <= $Vf; $i++) {  
    Action();  
}
```



Le corps de boucle est indiqué par les accolades

Langage PHP

Numérique & Science Informatique



Les parcours de tableaux : boucle bornée

```
function sum(array $array) : int
{
    $total = 0;
    foreach ( $array as $value )
        if ( is_int($value) )
            $total += $value ;
    return $total ;
}

// calcul la somme des 10 premiers termes
// d'une suite arithmétique de raison 1
echo sum(range(1, 10));
```

Langage PHP

Numérique & Science Informatique



Parcours de tableaux associatifs

```
function vowel_count(array $array) : int
{
    $total = 0;
    foreach ( $array as $key => $value )
        if ( in_array($key, array('a', 'e', 'i', 'o', 'u', 'y')) )
            $total += $value ;
    return $total ;
}

// calcul la somme des voyelles
echo vowel_count(
    array('a' => 3, 'b' =>2, 'c' => 7, 'd' => 0, 'e' => 2, 'f' => 1)
);
```

Langage PHP

Numérique & Science Informatique



Les itérateurs

```
function gen(int $start, int $end, int $step=1) : iterable
{
    for ( $i = $start ; $i < $end ; $i += $step )
        yield $i;
}
```

```
// calcul la somme des 10 premiers termes
```

```
// d'une suite arithmétique de raison 1
```

```
$total = 0;
```

```
foreach ( gen(1, 11) as $value )
```

```
    $total += $value ;
```

```
echo $total;
```

Langage PHP

Numérique & Science Informatique

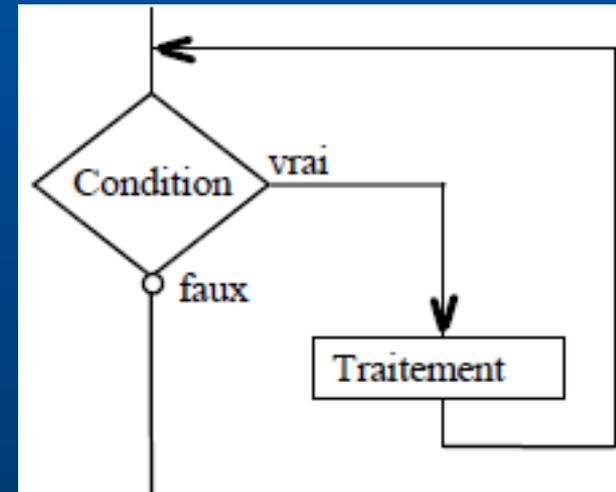


Boucle WHILE : nombre d'itérations **inconnu**

tanque <expression logique (vraie)> **faire**
Traitement

```
while ( condition ) {  
    Traitement();  
}
```

La condition est de type booléen



Les instructions à exécuter sont indiquées par les accolades

Langage PHP

Numérique & Science Informatique



Passage par référence

Une variable passée par référence à une fonction peut être modifiée

```
function swap(int& $a, int& $b) : void
{
    $tmp = $a;
    $a = $b;
    $b = $tmp;
}
```

```
$a = 3; $b = 82;
swap($a, $b);
```

```
echo $a. " " . $b;
```

Langage PHP

Numérique & Science Informatique



Closure

fonction définie dans le corps d'une autre fonction
utilise des paramètres ou des variables locales de cette dernière

```
function test() : int
{
    // Notre closure
    $double = function(int $a) : int {
        return $a * 2;
    };

    return $double(4);
}

echo test();
```

Langage PHP

Numérique & Science Informatique



Manipulation de fichiers

4 primitives de base :

- Ouvrir
- Écrire
- Lire
- Fermer



Langage PHP

Numérique & Science Informatique



Ouvrir un fichier

`fopen(string $fileName, string $mode) : resource | bool`

Modes d'ouverture :

- r : lecture
- w : écriture (écrase)
- w+ : écriture et lecture (écrase)
- a : ajout

Retour :

- pointeur sur une structure de fichier
- false si erreur

Langage PHP

Numérique & Science Informatique



Écrire dans un fichier

`fwrite(resource $handle , string $string , int $length = ?) : int|false`

Paramètres :

- `handle` : pointeur de système de fichiers
- `string` : la chaîne à écrire
- `length` : nombre d'octets à écrire

Retour :

- Le nombre d'octets écrits
- `false` si erreur

Langage PHP

Numérique & Science Informatique



Lire un fichier

`fread(resource $handle , int $length) : string|false`

Paramètres :

- `handle` : pointeur de système de fichiers
- `length` : nombre d'octets à lire

Retour :

- La chaîne lue
- `false` si erreur

Langage PHP

Numérique & Science Informatique



Fermer un fichier

`fclose(resource $handle) : bool`

Paramètres :

- `handle` : pointeur de système de fichiers

Retour :

- `true` en cas de succès ou `false` si erreur

Le nombre de fichiers ouverts en simultané est **limité** par l'OS

Langage PHP

Numérique & Science Informatique



Les exceptions

PHP a une gestion des exceptions similaire à ce qu'offrent les autres langages de programmation.

```
try {  
    if ( rand(0, 1) == 1)  
        throw new Exception("This is an exception");  
}  
catch(Exception $e) {  
    echo $e->getMessage() . "<br/>";  
}  
finally {  
    echo "Process complete";  
}
```

Langage PHP

Numérique & Science Informatique



Les sessions

Une session en PHP repose sur un identifiant
(une suite de chiffres et de lettres)

Les variables de session servent à stocker des informations liées à l'internaute sur tout le site.

Attention :

- Ne pas faire apparaître l'identifiant dans la barre URL
- Fermer le navigateur

Langage PHP

Numérique & Science Informatique



Création des variables de session

```
<?php
// on teste si les variables sont définies
if ( @$_POST['login'] && @$_POST['passwd'] ) {
    // tout est ok, on peut démarrer une session
    session_start();

    // on enregistre les paramètres du visiteur
    $_SESSION['login']    = $_POST['login'];
    $_SESSION['passwd'] = $_POST['passwd'];

    // on redirige notre visiteur vers une autre page
    header ('location: index.php');
}
?>
```

Langage PHP

Numérique & Science Informatique



Vérification des variables de session

```
<?php
    // démarrage session (indispensable dans toutes les pages)
    session_start();

    // on teste si les variables sont définies
    if ( @$_SESSION['login'] && @$_SESSION['passwd'] ) {
        // on exécute ci-dessous le code ad-hoc
    }
    else
        // on redirige notre visiteur vers la page d'identification
        header ('location: login.html');
?>
```

Langage PHP

Numérique & Science Informatique



Destruction des variables de session

```
<?php
    // démarrage session
    session_start();

    // destruction des variables de session
    session_unset();

    // destruction de la session
    session_destroy();

    // redirection du visiteur vers la page d'identification
    header ('location: login.html');
?>
```

Langage PHP

Numérique & Science Informatique



Liste des sessions sur le serveur

```
<?php
// modifie le chemin de sauvegarde des sessions (en tout début)
session_save_path('./data');

// corps du programme avec session_start() ...

$sessionNames = Array();
foreach ( scandir(session_save_path()) as $file )
    if ( $file != "." && $file != ".." ) {
        // décode les données encodées de session
        session_decode(
            file_get_contents(session_save_path(). "/" . $file));
        array_push($sessionNames, $_SESSION);
        session_unset();
    }
?>
```

Langage PHP

Numérique & Science Informatique



Les classes

```
class human {  
    public      const AGE = 20;  
    protected  string $gender = "M";  
    private    string $name  = "";  
  
    public function __construct(string $name)  
    {  
        $this->name = $name;  
    }  
    public function get_name() : string  
    {  
        return $this->name;  
    }  
}
```

```
$h = new human("John");  
echo $h->get_name() . " " . human::AGE;
```

Langage PHP

Numérique & Science Informatique



Constructeur et destructeur

```
class myClass
{
    public function __construct()
    {
        //Constructeur : myClass::__construct()
        echo 'Constructeur : ' . __METHOD__ . '()' . PHP_EOL;
    }
    public function __destruct()
    {
        //Destructeur : myClass::__destruct()
        echo 'Destructeur : ' . __CLASS__ . '::' . __FUNCTION__ . '()' . PHP_EOL;
    }
}

new myClass();
```

Langage PHP

Numérique & Science Informatique



Constantes

PHP dispose de nombreuses constantes prédéfinies :

- `__FILE__` The full path and filename of the file
- `__DIR__` The directory of the file
- `__FUNCTION__` The function name
- `__CLASS__` The class name
- `__METHOD__` The class method name
- `__LINE__` The current line number of the file
- `__NAMESPACE__` The name of the current namespace
- ...

Langage PHP

Numérique & Science Informatique



Exemple : test unitaire débrayable

Test le fichier courant et le fichier source

```
if ( strtolower(pathinfo(__FILE__)[ 'filename' ]) ==
    strtolower(pathinfo($_SERVER[ 'REQUEST_URI' ])[ 'filename' ]) ) {

    // do unit tests

}
```

`$_SERVER` — tableau de variables de serveur et d'exécution

Langage PHP

Numérique & Science Informatique



L'héritage

```
class woman extends human {
    public function __construct(string $name)
    {
        parent::__construct($name);
        $this->gender = "F";
    }
    public function __toString() : string
    {
        return $this->get_name();
    }
}
```

```
$w = new woman("Jane");
echo $w. " " . human::AGE;
```

Langage PHP

Numérique & Science Informatique



Interfaces

spécifie les méthodes qu'une classe doit implémenter

```
interface iTemplate          // Déclaration de l'interface 'iTemplate'
{
    public function setVariable(string $name, int $var) : void;
}

class Template implements iTemplate
{
    private $vars = array();

    // Implémentation de l'interface
    public function setVariable(string $name, int $var) : void
    {
        $this->vars[$name] = $var;
    }
}
```

Langage PHP

Numérique & Science Informatique



Connexion PHP / SGBDR

L'accès à une base de données s'effectue par l'intermédiaire de pilotes.

- PHP Data Objects (PDO)
Créé en 2005, version 5.1 de PHP, licence PHP
- MySQLi (« i » pour « improved »)
Créé en 1994, version 8 en 2017, licence GNU/GPL

Langage PHP

Numérique & Science Informatique



Connexion avec PDO

```
try {
    $pdo = new PDO(
        "mysql:host=$servname;dbname=$database",
        $user,
        $passwd);

    if ( $stmt = $pdo->query("SELECT * FROM users" ) )
        if ( $users = $stmt->fetchAll() )
            foreach ( $users as $user )
                print($user['id'] ." -> ". $user['nom'] ."<br/>");
    }
catch(PDOException $e) {
    print("erreur : $e->getMessage()");
}
```

Langage PHP

Numérique & Science Informatique



Connexion avec mysqli

```
$mysqli = new mysqli($servername, $user, $passwd, $database) ;
```

```
if ( $mysqli )  
    if ( $result = $mysqli->query("SELECT * FROM users") )  
        while ( $user = $result->fetch_row() )  
            print($user[0] ." -> ". $user[1] ."<br/>");
```

Langage PHP

Numérique & Science Informatique

