



**HAL**  
open science

## Bases de la programmation

Benoît Prieur

► **To cite this version:**

Benoît Prieur. Bases de la programmation. École thématique. IT-Akademy, France. 2020, pp.68.  
hal-02527209v2

**HAL Id: hal-02527209**

**<https://hal.science/hal-02527209v2>**

Submitted on 2 May 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License



# Bases de la programmation

V 0.4 (2 mai 2020) - IT-Akademy (Lyon, mars-avril 2020)

Benoît Prieur - SoartheC - CC-BY-SA 4.0



# Objectifs

- Bases de la programmation
- Algorithmique
- Usage de PHP/MySQL/PhpMyAdmin

# Notion de CLI

- *command line interface*  
(interface en ligne de commande)
  - MS DOS et Power Shell (Windows)
  - BASH : shell Unix du projet GNU (Linux)
  - Terminal (MacOS)

```
PS C:\> Get-Childitem 'MediaCenter\Music' -rec |
>> where < -not $_.PSIsContainer -and $_.Extension -match 'unimp3' > |
>> Measure-Object -property length -sum -min -max -ave
>>
Count          : 1397
Average        : 5491276.09563887
Sum            : 7177097857
Maximum        : 22905267
Minimum        : 3235
Property       : Length

PS C:\> Get-WmiObject CIM_BIOSElement | select biosv*, man*, ser* | Format-List
BIOSVersion    : (IOSCPL - 6040000, Ver 1.00PARTBL)
Manufacturer   : TOSHIBA
SerialNumber    : M021116H

PS C:\> ([wmiSearcher]@'
>> SELECT * FROM CIM_Job
>> WHERE Priority > 1
>> '@).get() | Format-Custom
>>
class ManagementObject#root\cimv2\Win32_PrintJob
{
    Document = Monad Manifesto - Public
    JobId = 6
    JobStatus =
    Owner = User
    Priority = 42
    Size = 1027008
    Name = Epson Stylus COLOR 740 ESC/P 2, 6
}

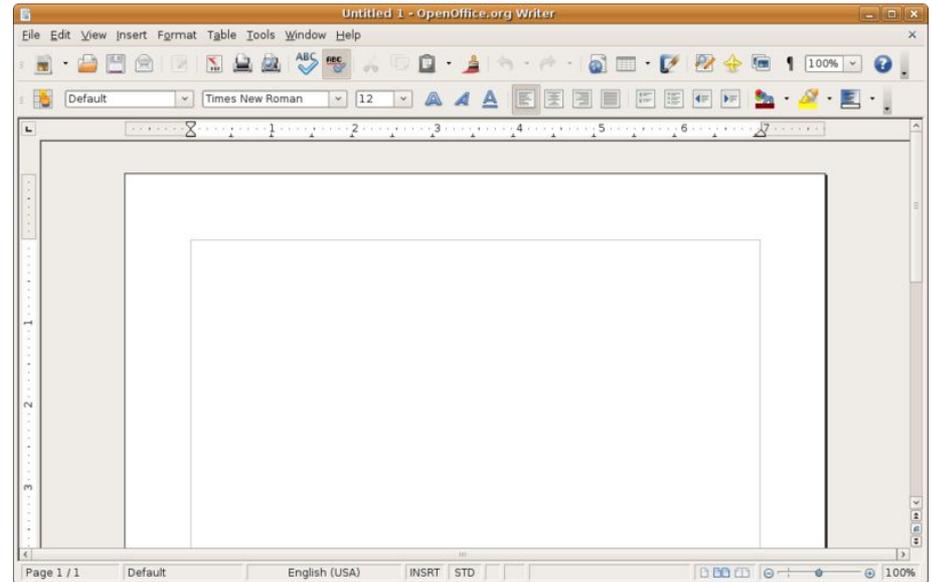
PS C:\> $rssUrl = 'http://blogs.msdn.com/powershell/rss.aspx'
PS C:\> $blog = [xml](New-object System.Net.WebClient).DownloadString($rssUrl)
PS C:\> $blog.rss.channel.item | select title -first 3

title
----
MMS: What's Coming In PowerShell 02
PowerShell Presence at MMS
MMS Talk: System Center Foundation Technologies

PS C:\> $host.version.ToString().Insert(0, 'Windows PowerShell: ')
Windows PowerShell: 1.0.0.0
PS C:\>
```

# Applications fenêtrées

- Applications “Desktop”

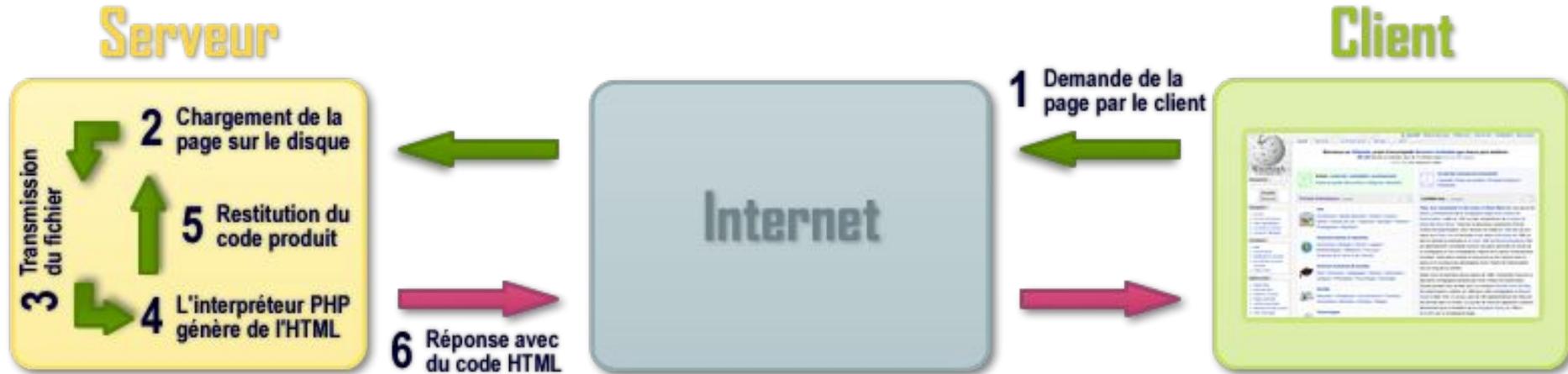




# Applications web (1)

- Client-serveur
- Technologie web (HTML, CSS, javascript)
- Serveur web (serveur de pages web)

## Applications web (2)





# Applications hybrides

- Utilisation du navigateur web
- Ou du moins des technologies web en local
- L'exemple du développement mobile (cross-platform software)



# Applications mobiles natives

- Native, hybride ou web
- Native
  - iOS (Objective-C, Swift)
  - Android (Java)
- Hybride/web
  - Ionic
  - Cordova
- Autre exemple : Xamarin



# Applications embarquées (1)

- Informatique embarquée
- Objets connectés

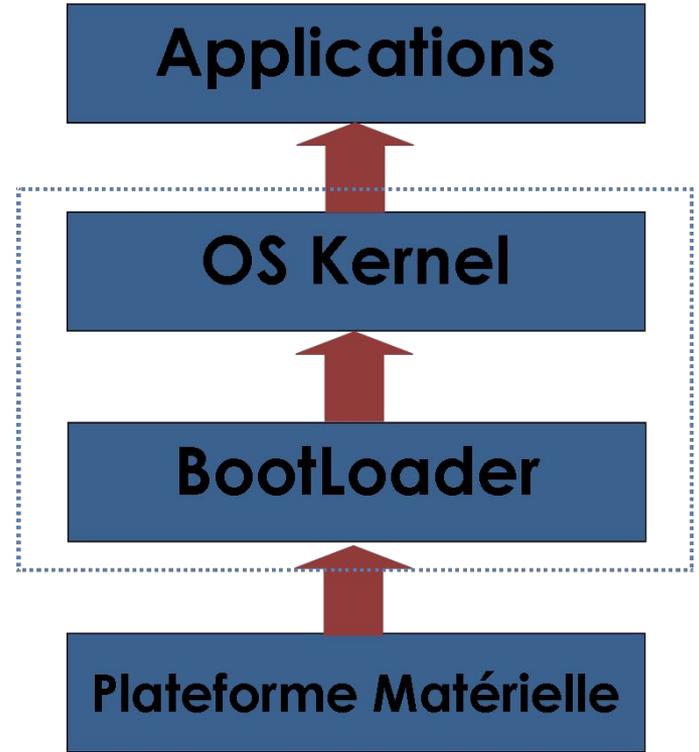


## Applications embarquées (2)

- Notions de driver (pilote)
- Notions de firmware (micrologiciel)
- firmware des ordinateurs (BIOS ou EFI) exécuté par le CPU



## Applications embarquées (3)





# Langages de scripts (interprétés)

- Interprété par un interpréteur en temps réel
- Langage semi-interprété : préalablement traduit en bytecode (préservation des performances)
  - Exemples de .Net ou Java
- Séquence d'interprétation :
  - Lire et analyser une instruction
  - Si l'instruction est correcte, l'exécuter
  - Passer à l'instruction suivante



# Langages compilés

- “Traduction” en langage machine, par la compilation (compilateur)
- Par opposition au script (exécution pas-à-pas)
- Quelques exemples :
  - C/C++
  - Fortran
  - Rust, etc.



# Langages de programmation de bas niveau

- Proximité au matériel
- Langage machine
- Peu d'abstraction par rapport au processeur
  - Adressage et allocation sur le tas (heap)
  - Usage de la pile (stack)

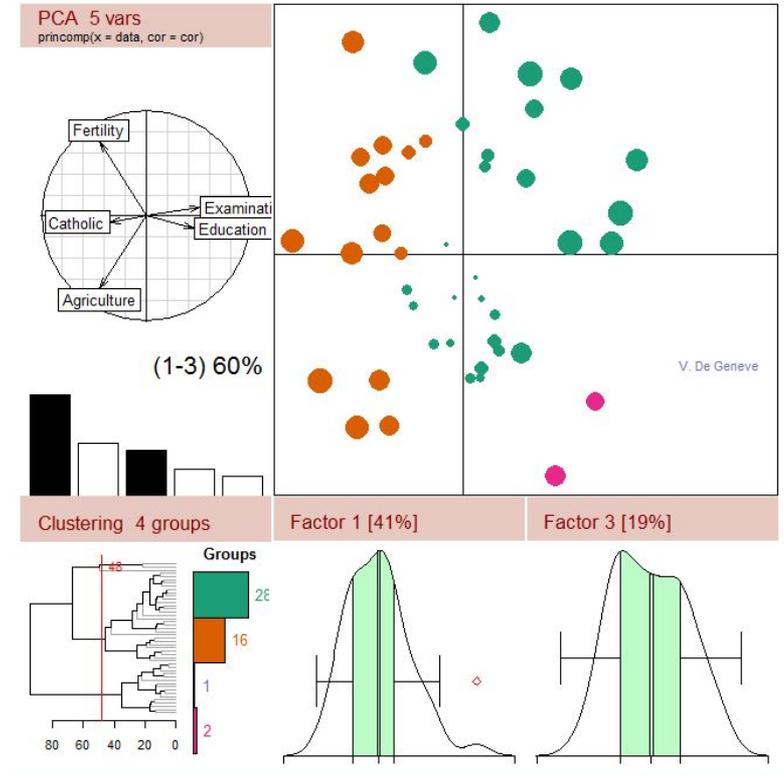


# Langages de programmation de haut niveau

- De l'abstraction
- Perte de performance très relatif (optimisation)
- La question des langages C/C++
- Notion de *garbage collector*

# Langages spécialisés

- Le langage R
- Spécialisé en statistique et en sciences des données (data science)



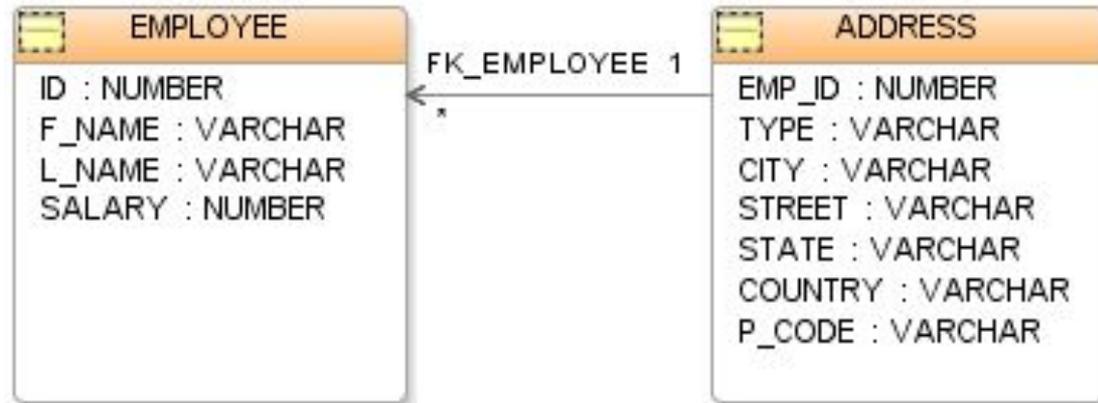


## Langage R, exemple de code

```
> x <- 1:6 # Create vector.  
> y <- x^2 # Create vector by formula.  
> print(y) # Print the vector's contents.  
[1] 1 4 9 16 25 36  
  
> mean(y) # Arithmetic mean of vector.  
[1] 15.16667
```

# Langage SQL

- Langage de requêtage des bases de données relationnelles
- Exemples de requêtes avec l'exemple ci-joint.



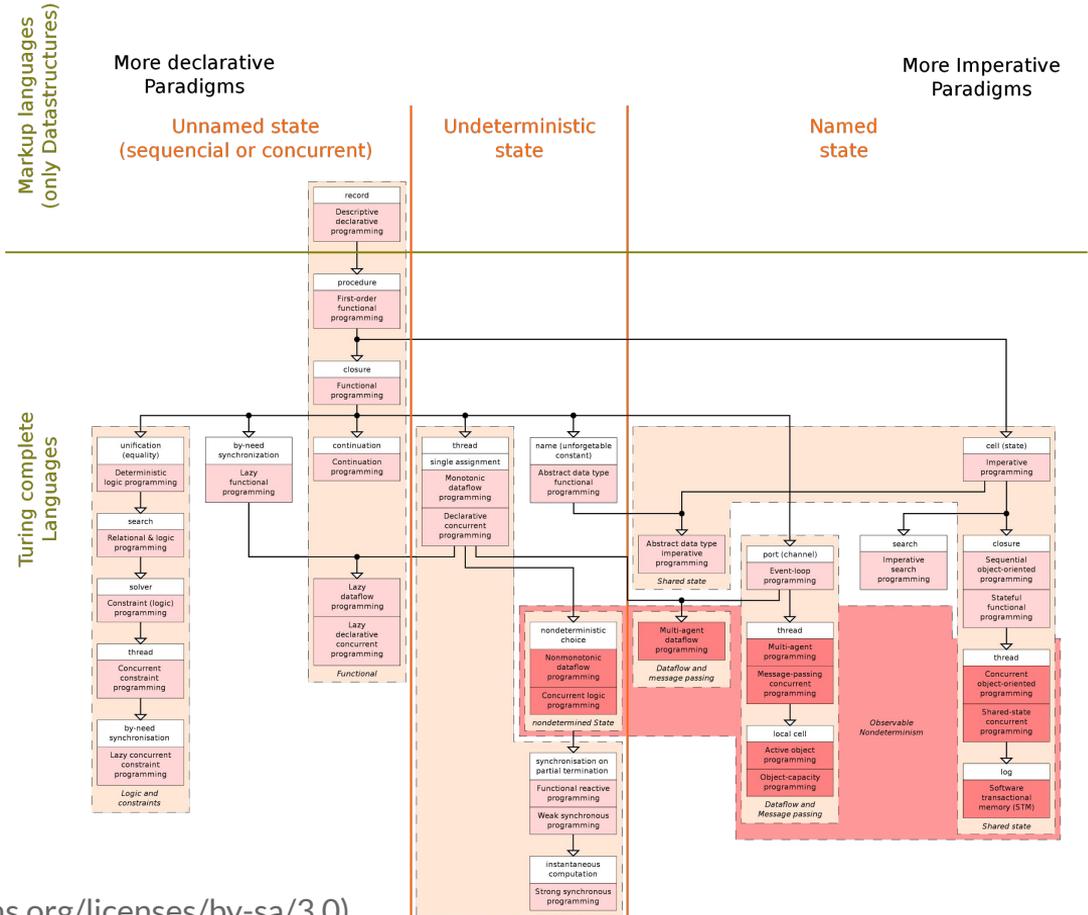


# Paradigmes de programmation

- Procédural
- Programmation orientée objet (POO)
- Fonctionnel

Lien direct :

[https://upload.wikimedia.org/wikipedia/commons/f/f7/Programming\\_paradigms.svg](https://upload.wikimedia.org/wikipedia/commons/f/f7/Programming_paradigms.svg)





# Programmation procédurale

- Procédure, routine, sous-routine ou fonction
- Parfois complétée par la POO



# Programmation orientée objet (1)

- Objet (attributs et méthodes)
- Héritage et polymorphisme
- Notion de classe et d'instance de classe (métaphore du moule)



## Programmation orientée objet (2)

- Exemple de la classe Button
- Mention de UML

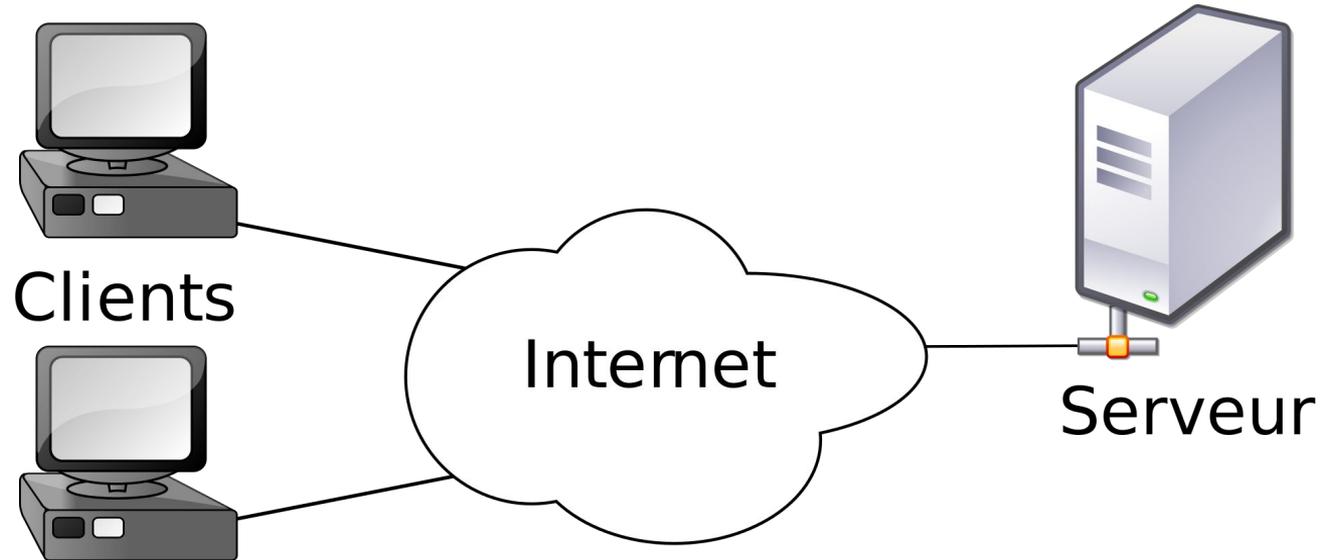
<b>Button</b>
- xsize - ysize - label_text - interested_listeners - xposition - yposition
+ draw() + press() + register_callback() + unregister_callback()



# Programmation fonctionnelle

- Calcul comme évaluation de fonctions mathématiques
- OCaml, LISP, SCALA

# Approche client-serveur



Crédit : Gnome-fs-server.svg: David VignoniGnome-fs-client.svg: David Vignoni - derivative work: Calimo (d) / LGPL (<http://www.gnu.org/licenses/lgpl.html>)



# Langages *client-side* et *server-side*

- Activité collective
- Lieu d'exécution des langages :
  - HTML, PHP, Javascript, CSS, Python
  - Autres langages



# Installation de l'environnement de développement (1)

- IDE : Visual Studio Code
- Installation :
  - Apache (serveur web)
  - PHP (environnement)
  - MySQL (base de données) et PhpMyAdmin (client web)



## Installation de l'environnement de développement (2)

- Vérifier la bonne installation de MySQL en accédant aux bases de données avec PhpMyAdmin
- Vérifier la bonne installation de PHP avec `<?php phpinfo(); ?>`
- Vérifier la bonne installation de PHP CLI avec un script à exécuter en ligne de commande



## Installation de l'environnement de développement (3)

- Tester PHP CLI

```
<?php
```

```
echo "Hello! What is your name (enter below):\n";
```

```
$strName = fread(STDIN, 80);
```

```
echo 'Hello ', $strName, "\n";
```

```
?>
```



# Notion de tableau (1)

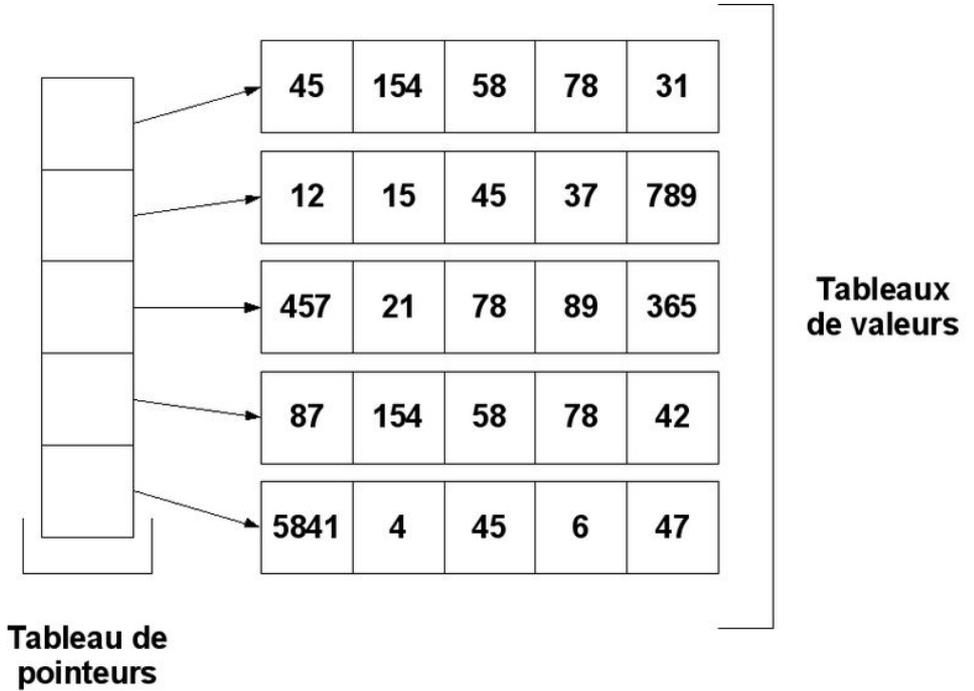
- Structure de données
- Tableau à une dimension
- Tableau à plusieurs dimensions
- Index commence à zéro en général



## Notion de tableau (2)

<b>Valeur</b>	<b>45</b>	<b>154</b>	<b>58</b>	<b>78</b>	<b>31</b>	<b>5</b>	<b>74</b>
<b>Index</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>

# Notion de tableau (3)





## Notion de pseudo-code (1)

- Langage de Description d'Algorithmes (LDA)
- Langage (presque) naturel

**Fonction factorielle (n)**

**r = 1**

**Pour i de 1 jusqu'à n avec un pas de 1**

**r = r\*i**

**Fin pour**

**Retourner r**

**Fin Fonction**



## Notion de pseudo-code (2)

1. Prendre en paramètre deux longueurs d'un triangle rectangle. Calculer la longueur de l'hypoténuse. Fonction **Racine(a)**.
2. Retourner le nombre d'occurrences d'une lettre dans une chaîne de caractères pour une lettre donné.
3. Renvoyer vrai quand le nombre passé en paramètre est pair. Renvoyer faux sinon.
  - a. Fonction **Modulo(a, b)**.



## Notion de pseudo-code (3)

On a un tableau en entrée. Il inclut les notes d'une classe dans une matière donnée. De 0 à 20. Par exemple [ 4 15 18 9 13 12 ]

- Note présente ou non. **Chercher(tableau)** ; renvoie un booléen.
- **Somme(tableau)**
- **Moyenne(tableau)**
- **Min(tableau)**
- **Max(tableau)**



# Notion de variable (1)

- Un nom et une valeur
- Les types :
  - Chaîne de caractères (string)
  - Nombres entiers (int)
  - Nombre décimaux (float)
  - Booléens (bool)
  - “ rien “ (NULL)



## Notion de variable (2)

```
<?php
```

```
$annee_courante = 2020;
```

```
?>
```



## Notion de variable (3)

- Notion de concaténation

```
<?php
```

```
$annee_courante = 2020;
```

```
echo "Nous sommes en ";
```

```
echo $annee_courante;
```

```
?>
```

```
<?php
```

```
$annee_courante = 2020;
```

```
echo "Nous sommes en $annee_courante";
```

```
?>
```



## Calculs arithmétiques (1)

- +
- -
- \*
- /

```
<?php
```

```
$nombre = 2 + 4;
```

```
$nombre = 3 - 2;
```

```
$nombre = 3 * 6;
```

```
$nombre = 10 / 5;
```

```
$nombre = 3 * 3 + 3;
```

```
$nombre = (3 + 3) * 2;
```

```
?>
```



## Calculs arithmétiques (2)

- $-a$  : négation
- $a \% b$  : modulo
- $a^{**} b$  : exponentielle



## Conditions (1)

- `==` : est égal à
- `>` : est supérieur à
- `<` : est inférieur à
- `>=` : est supérieur ou égal à
- `<=` : est inférieur ou égal à
- `!=` : est différent de



## Conditions (2)

```
<?php
$nombre = 8;

if ($nombre < 10)
{
    echo "En-dessous de la moyenne";
}
else
{
    echo "Au-dessus de la moyenne";
}

?>
```



## Conditions multiples (1)

- AND (&&)
- OR (||)



## Conditions multiples (2)

- AND (&&)
- OR (||)

```
if ($club == "OL" OR $club == "LOU")  
{  
    echo "Le club est basé à Lyon";  
}
```



## switch...case...default

```
<?php
$temp = 25;

switch($temp) // commentaire
{
    case 0:
        echo "Il gèle";
        break;

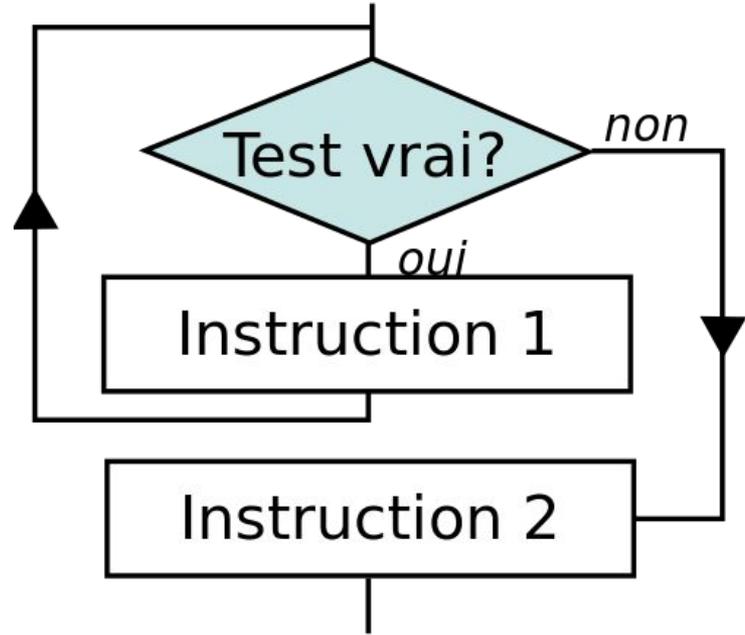
    case 5:
        echo "Il fait froid";
        break;

    case 20:
        echo "Il fait chaud";
        break;

    default:
        echo "Coucou";
}
?>
```

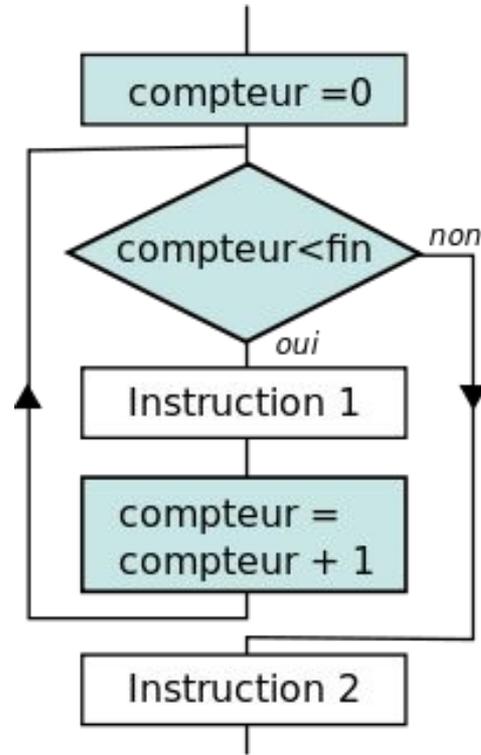
## Boucle while (tant que)

```
<?php
while ($continuer == true)
{
    // instructions
}
?>
```



## Boucle for

```
<?php
for ($nombre = 1; $nombre <= 100; $nombre++)
{
    echo $nombre;
}
?>
```





# Tableaux en PHP (1)

```
<?php
```

```
$prenoms = array ('Jean-Philippe', 'Régis', 'Sophiane', 'Nicolas');
```

```
?>
```



## Tableaux en PHP (2)

```
<?php
```

```
$prenoms[0] = 'Jean-Philippe';
```

```
$prenoms[1] = 'Régis';
```

```
$prenoms[2] = 'Sophiane';
```

```
$prenoms[3] = 'Nicolas';
```

```
?>
```



## Tableaux en PHP (3)

```
<?php
```

```
echo $prenoms[1];
```

```
?>
```



## Tableaux associatifs en PHP (1)

```
<?php
$fiche = array (
    'prenom' => 'Benoît',
    'nom' => 'Prieur',
    'ville' => 'Saint-Maurice-de-Beynost');
?>
```

```
<?php
$fiche['prenom'] = 'Benoît';
$fiche['nom'] = 'Prieur';
$fiche['ville'] =
'Saint-Maurice-de-Beynost';
?>
```



## Tableaux associatifs en PHP (2)

```
<?php
```

```
echo $fiche['ville'];
```

```
?>
```



# Parcours de tableaux en PHP (1)

- Boucle for
- boucle foreach
- print\_r



## Parcours de tableaux en PHP (2)

```
for ($numero = 0; $numero < 5; $numero++)  
{  
    echo $prenoms[$numero];  
}
```

```
foreach($fiche as $element)  
{  
    echo $element;  
}
```



## Parcours de tableaux en PHP (3)

```
<?php foreach($fiche as $cle => $element) ?>
```

```
foreach($fiche as $cle => $element)
{
    echo '[' . $cle . '] vaut ' . $element;
}
?>
```



# print\_r pour le débogage en PHP

- Affiche l'ensemble du contenu du tableau

```
print_r($fiche);
```



# Les fonctions en PHP (1)

- Fonctions prêtes à l'emploi
- Fonctions définies



## Les fonctions en PHP (2)

```
<?php
$phrase = 'Bonjour IT-Akademy';
$longueur = strlen($phrase);

echo 'La phrase a ' . $longueur . ' caractères';

$phrase2 = str_replace('Bonjour', 'Coucou', $phrase);

?>
```



## Les fonctions en PHP (3)

- strlen
- str\_replace
- str\_shuffle
- strtolower
- date
- etc.



## Les fonctions en PHP (4)

```
<?php  
function Bonjour($nom)  
{  
    echo 'Bonjour ' $nom';  
}
```

```
Bonjour('Benoît');  
Bonjour('Toto');  
?>
```



## Exercice en PHP CLI (1)

On a un tableau en entrée. Il inclut les notes d'une classe dans une matière donnée. De 0 à 20. Demander à l'utilisateur le nombre de notes. Puis faire entrer les notes.

- Moyenne(tableau)
- Min(tableau)
- Max(tableau)



## Exercices en PHP CLI (2)

1. Prendre en paramètre deux longueurs d'un triangle rectangle. Calculer la longueur de l'hypoténuse (voir fonction **sqrt**)
2. Retourner le nombre d'occurrences d'une lettre dans une chaîne de caractères pour une lettre donné.
3. Renvoyer vrai quand le nombre passé en paramètre est pair. Renvoyer faux sinon (voir **%**)



## Exercice en PHP Web (1)

- Créer une page PHP
- Afficher la date du jour dans une balise H1
- Afficher une liste à puces (tags li, ul) contenant une liste de prénoms (“codés en dur”). Utiliser une boucle.
- Afficher un tableau des capitales du monde. Utiliser une boucle.



# La notion d'include (1)

- Inclusion de page en PHP

```
<?php include("menus.php"); ?>
```



## La notion d'include (2)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mon site</title>
  </head>
  <body>

    <?php include("menus.php"); ?>

    <?php include("pied_de_page.php"); ?>

  </body>
</html>
```



# La notion de require

```
<?php
```

```
require('somefile.php');
```

```
?>
```



## Exercice en PHP Web (2)

*(facultatif)*

- Afficher une combobox listant quelques pays du monde.
- Quand on choisit un pays, on obtient la capitale.
- Les pays-capitales sont issus d'une table MySQL PAYS.



# Les 7 travaux-pratiques

- Envoi du README.txt (Github) par messagerie instantanée
- Réaliser le tout et le livrer dans un repository Github dédié