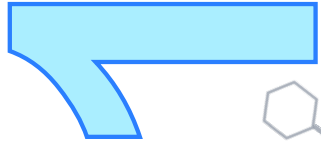
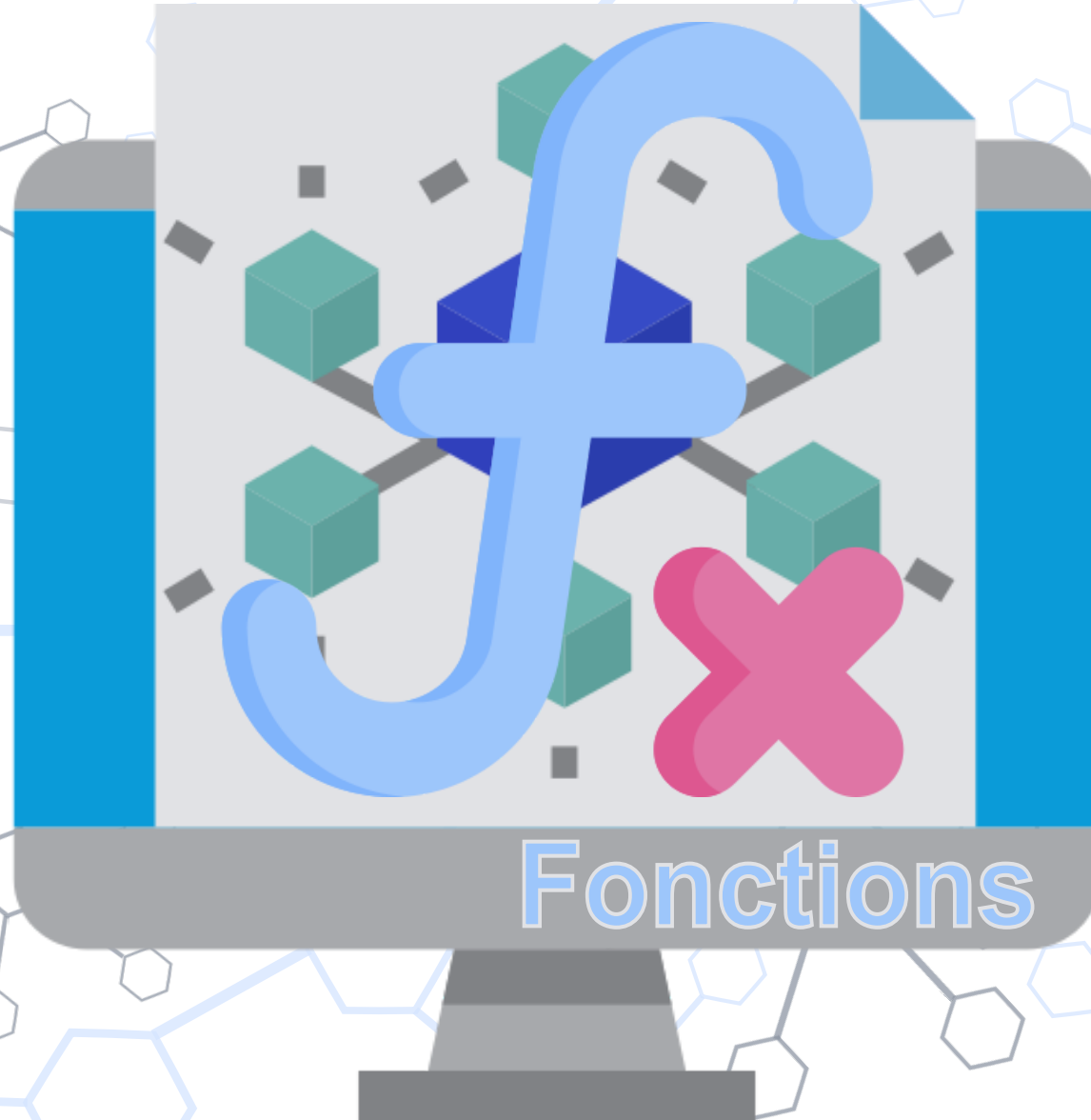


# Structures avancées PHP



oo  
m



# Reproductibilité des comportements

Comment faire si le même comportement doit être reproduit à des endroits différents d'un **même algorithme** ou des algorithmes différents ?

Plusieurs solutions :



Utiliser une boucle avec le comportement reproductible à l'intérieur

**NON**

**Le même comportement** doit être reproduit à des endroits différents de l'application



Faire du copié-collé du comportement partout il est nécessaire

**OUI, mais...**

Toute **modification du comportement** doit être **reproduite** partout (et sans erreur)

**Difficulté de réutilisation du comportement** dans des algorithmes différents



**Factoriser le comportement** et l'appeler où il est nécessaire

**OUI** (notions de procédure et de fonction)

**Maintenance unique** du comportement

**Réutilisation** dans des algorithmes différents

**Adaptabilité du comportement** en fonction des besoins

# Factorisation du code (1/2)

**Reporter le comportement dans une structure réutilisable**

**Algo**

*Pour effectuer des actions*

## **PROCEDURE**

- *modification de valeurs*
- *affichage*

*Pour obtenir un résultat*

## **FONCTION**

- *traitement et renvoi de valeurs*

**PROCEDURE** *nom*

**DEBUT**

*Actions*

**FIN**

**FONCTION** *nom* : **type**

**DEBUT**

*Actions*

**RETOURNER** *valeur*

**FIN**

## Factorisation du code (2/2)

Adapter le comportement suivant des paramètres

```
PROCEDURE nom (parametre : type)  
DEBUT  
    Actions  
FIN
```

```
FONCTION nom (parametre : type) : type  
DEBUT  
    Actions  
    RETOURNER valeur  
FIN
```

# Procédures et fonctions prédéfinies

## Comportements fournis par un tiers

Algo

PROCEDURE SAISIR (message : CHAINE, parametre : VARIABLE)

FONCTION SAISIR (message : CHAINE) : VARIABLE

PROCEDURE AFFICHER (message : CHAINE)

FONCTION ALEA (parametre : ENTIER) : ENTIER

## Utilisation (appel)

**SAISIR** ("Quel est votre prenom ?", prenom)

nom <- **SAISIR** ("Quel est votre nom ?")

**AFFICHER** ("Bonjour " & prenom)

secret <- **ALEA** (500)

# Procédures personnalisées

## Déclaration

Algo

**PROCEDURE** AfficheParite (**Nombre** : ENTIER)

**DEBUT**

SI (**Nombre** mod 2 = 0) ALORS

AFFICHER("Le nombre " & **Nombre** & " est pair")

SINON

AFFICHER("Le nombre " & **Nombre** & " est impair")

FIN SI

**FIN**

## Utilisation

AfficheParite(15)

*Le nombre 15 est impair*

...

AfficheParite(2\*2)

*Le nombre 4 est pair*

# Fonctions personnalisées

Algo

## Déclaration

**FONCTION** Carre(Nombre : ENTIER) : ENTIER

### VARIABLES

nbTemp : ENTIER

### DEBUT

nbTemp <- Nombre\*Nombre

**RETOURNER** nbTemp

### FIN

## Utilisation

**Val** <- Carre(5)

*Val vaut 25*

...

**Val2** <- 3

**Val3** <- Carre(Val2)

*Val3 vaut 9*

...

**AFFICHER**(Carre(4))

*Affiche le nombre 16*

# Visibilité des variables

Algo

PROCEDURE **maProc** (**parametre** : ENTIER)

VARIABLES

**variableB** : ENTIER

DEBUT

**parametre** accessible

**variableB** accessible

**variableA** accessible

FIN

ALGO monAlgo

VARIABLES

**variableA** : ENTIER

DEBUT

**variableA** accessible

**maProc**() accessible

**parametre** accessible

**variableB** inaccessible

FIN

ALGO

**variableA**

PROCEDURE **maProc**(**parametre**)

**variableB**

Globale

Locale



# Fonctions PHP

Pas de distinction fonctions et procédures

PHP

**PROCEDURE** *nom*

Algo

**DEBUT**

*Actions*

**FIN**

**FONCTION** *nom* : **type**

**DEBUT**

*Actions*

**RETOURNER** *valeur*

**FIN**

**function** *nom*

PHP

{

*Actions*

}

**function** *nom*

{

*Actions*

**return** *résultat*

}

# Fonction avec paramètre

**Afficher un message avec une casse aléatoire**

**function** afficheTexteBizarre(**string** \$leTexte)

```
{  
    for ($i = 0; $i < strlen($leTexte); $i++)  
    {  
        if (rand(1,2) == 1)  
        { echo strtolower($leTexte[$i]); }  
        else  
        { echo strtoupper($leTexte[$i]); }  
    }  
}
```

afficheTexteBizarre("la vie est un long fleuve tranquille");

la Vie eST UN LONG fLeUve TRANQuILLe

# Fonction avec paramètres multiples

```
function afficheTexteBizarre(string $leTexte, int $repetitions)
{
    for ($r = 0; $r < $repetitions; $r++)
    {
        for ($i = 0; $i < strlen($leTexte); $i++)
        {
            if (rand(1,2) == 1)
            { echo strtolower($leTexte[$i]); }
            else
            { echo strtoupper($leTexte[$i]); }
        }
    }
}
```

```
afficheTexteBizarre('À vaincre sans péril, on triomphe sans gloire.', 3);
```

À VAincRE saNS péRil, on TRlompHE sANS GIOiRe.  
À VaiNCRE SAnS Péril, On TrIOMPHe SanS GIOiRe.  
À VaINCRe sans péRIL, on triomPHe SANS GloiRe.

# Fonction avec paramètre par défaut

```
function afficheTexteBizarre(string $leTexte = 'texte par défaut')
{
    for ($i = 0; $i < strlen($leTexte); $i++)
    {
        if (rand(1,2) == 1)
        { echo strtolower($leTexte[$i]); }
        else
        { echo strtoupper($leTexte[$i]); }
    }
}
```

```
afficheTexteBizarre();
```

**TExTE par DéFaUt**

**Attention à l'ordre :**

function maFonction (parametre1, parametre2=valeur)

~~function maFonction (parametre1=valeur, parametre2)~~

# Fonction avec paramètres variables

```
function afficheTexteBizarre(string ... $lesTextes)
{
    foreach ($lesTextes as $leTexte)
    {
        for ($i = 0; $i < strlen($leTexte); $i++)
        {
            if (rand(1,2) == 1)
            { echo strtolower($leTexte[$i]); }
            else
            { echo strtoupper($leTexte[$i]); }
        }
    }
}
```

```
afficheTexteBizarre('veni vici vidi', 'Alea jacta est', 'Cogito, ergo sum');
```

```
veni VICI ViDi
AleA jACTa EST
COgiTO, ERgo sUM
```

# Fonction avec paramètre tableau

```
function afficheTexteBizarre(array $lesTextes)
{
    foreach ($lesTextes as $leTexte)
    {
        for ($i = 0; $i < strlen($leTexte); $i++)
        {
            if (rand(1,2) == 1)
            { echo strtolower($leTexte[$i]); }
            else
            { echo strtoupper($leTexte[$i]); }
        }
    }
}
```

```
$tableau = array('veni vici vidi', 'Alea jacta est', 'Cogito, ergo sum');
afficheTexteBizarre($tableau);
```

```
veni VICI ViDi
AleA jACTa EST
COgiTO, ERgo sUM
```

## Fonction avec résultat

**Calculer le carré d'un nombre**

```
function donneCarre(int $leNombre) : int  
{  
    return $leNombre*$leNombre;  
}
```

```
$carre = donneCarre(5);  
echo($carre);
```

25

# Fonction avec résultats multiples

## Calculer les résultats d'une table de multiplication

```
function donneTableMultiplication(int $laTable) : array
{
    $tableResultat = [];
    for ($i = 1; $i <= 10; $i++)
    {
        $tableResultat[] = $i * $laTable;
    }
    return $tableResultat;
}
```

```
$tabAffichage = donneTableMultiplication(5);
foreach($tabAffichage as $ligne)
{
    echo($ligne) . ' ';
}
```

5 10 15 20 25 30 35 40 45 50



# Visibilité variables PHP

```
function testeVisibilitéVariables()  
{  
    $varLocaleB = (string) "B";  
}
```

```
testeVisibilitéVariables();  
echo $varLocaleB;
```

Warning:  
Undefined variable \$varLocaleB

```
function testeVisibilitéVariables()  
{  
    echo $varGlobaleA;  
}
```

```
$varGlobaleA = (string) "A";  
testeVisibilitéVariables();
```

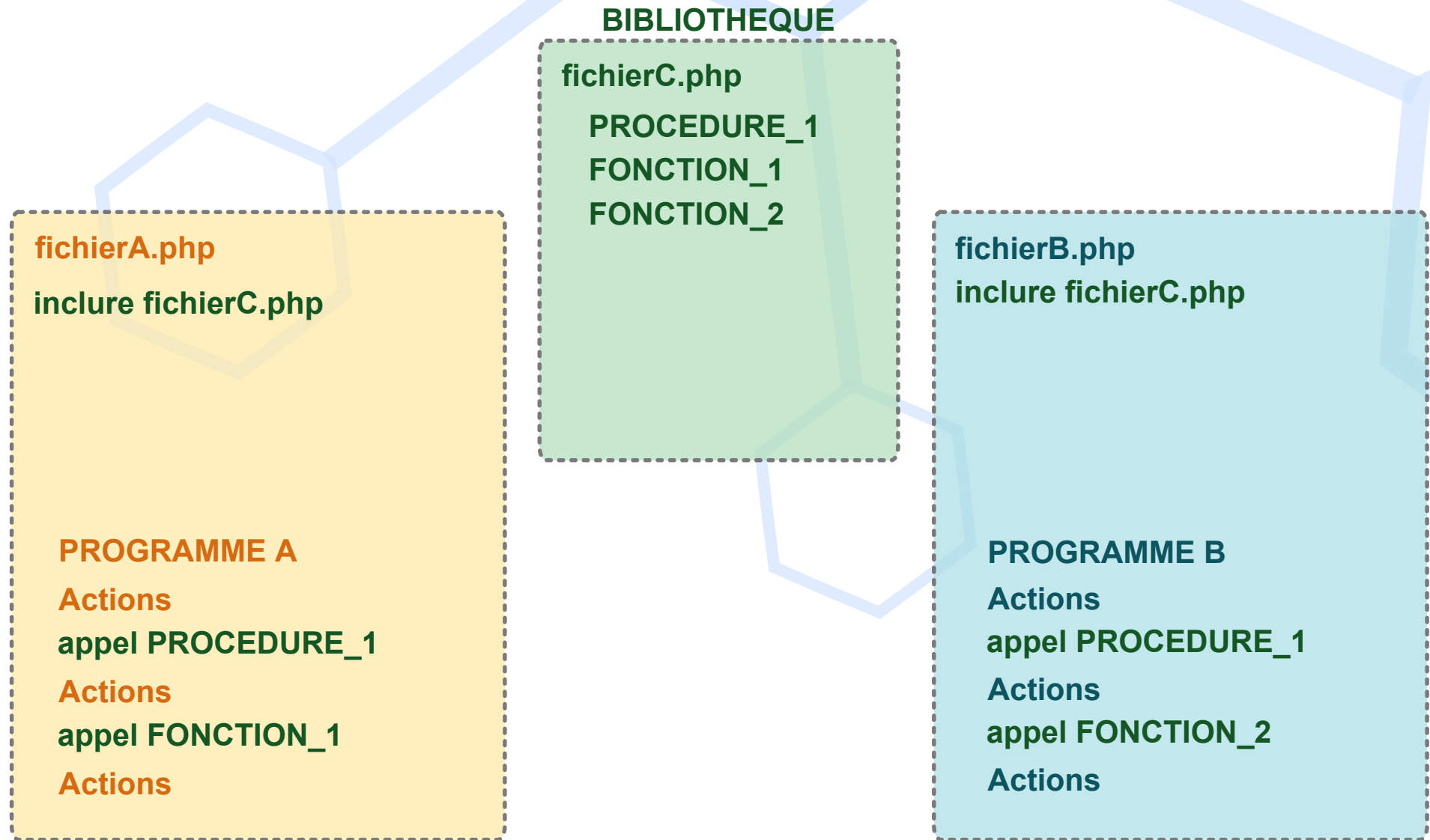
Warning:  
Undefined variable \$varGlobaleA

```
function testeVisibilitéVariables()  
{  
    global $varGlobaleA;  
    echo $varGlobaleA;  
}
```

```
$varGlobaleA = (string) "A";  
testeVisibilitéVariables();
```

A

# Organisation du code



# Inclusion de fichiers externes

```
include("bibliotheque.php");
```

Inclusion physique du fichier

```
include_once("bibliotheque.php");
```

Evite les inclusions multiples  
(redéclarations)

```
require_once("bibliotheque.php");
```

Arrêt si le fichier n'existe pas

## Utiliser des chemins relatifs :

dossierA/bibliothequeA.php

../sources/bibliothequeB.php

## Eviter les chemins absolus :

/var/www/html/monsite/sources/bibliothequeC.php

# Conditions d'utilisation

## **Vous êtes libres :**

Version 2526.1



de reproduire, distribuer et communiquer cette création au public



de modifier cette création

- **Johan RUDENT**
- Web : ***www.rudent.fr***
- Mail : ***contact@rudent.fr***

## **Selon les conditions suivantes :**



**Pas d'Utilisation Commerciale** — Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.



**Paternité** — Vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).



**Partage des Conditions Initiales à l'Identique** — Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/legalcode>