

# Passage de paramètres par valeur vs par référence



# Contenu

- **Le passage par valeur, c'est quoi?**
  - En PHP
- **Le passage par référence, c'est quoi?**
  - En PHP
- **Particularité pour les objets**
- **Et en JavaScript?**

# Le passage par valeur, c'est quoi?

# Rappel sur les fonctions

- **Nous avons vu que les fonctions peuvent prendre des paramètres**
- **Exemple en PHP:**

```
function direBonjour($prenom, $nom) {  
    echo "Bonjour $prenom $nom !";  
}
```

```
direBonjour('Homer', 'Simpson');
```

```
// Résultat: Affiche « Bonjour Homer Simpson! »
```

# Passage des arguments

- **Souvent, les arguments passés aux paramètres sont des variables**
- **Exemple en PHP:**

```
function direBonjour($prenom, $nom) {  
    echo "Bonjour $prenom $nom !";  
}
```

```
$monPrenom = 'Homer';
```

```
$monNom = 'Simpson';
```

```
direBonjour($monPrenom, $monNom);
```

```
// Résultat: Affiche « Bonjour Homer Simpson! »
```

# Passage des arguments

- **Dépendamment des cas et du langage de programmation, un argument est passé soit:**
  - Par **valeur**
  - Par **référence**

# Passage par valeur

**Prenons ce code PHP qui déclare une fonction puis l'appelle en lui passant une variable**

```
function maFonction($parametre1) {
```

```
    // ...
```

```
}
```

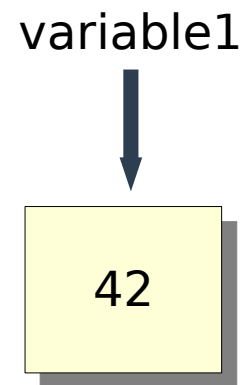
```
$variable1 = 42;
```

```
maFonction($variable1);
```

# Passage par valeur

```
$variable1 = 42
```

**En mémoire:**

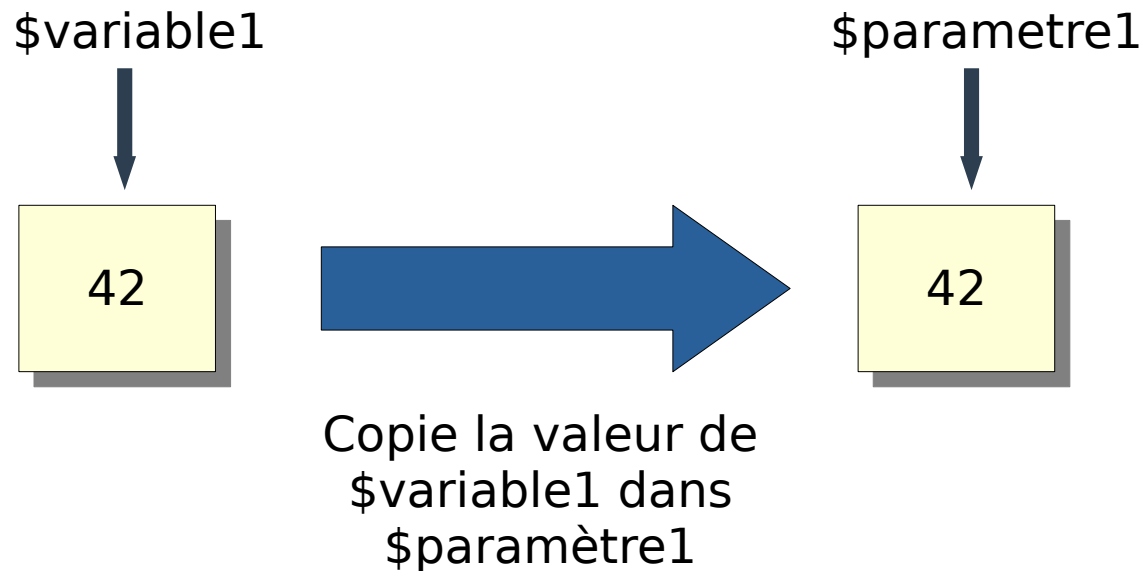




# Passage par valeur

`maFonction($variable1)` ← passe `$variable1` comme argument pour `$parametre1`

## En mémoire:



# Le passage par référence, c'est quoi?

# Passage par référence

Dans cet exemple, l'argument assigné à paramètre1 est **passé par référence**:

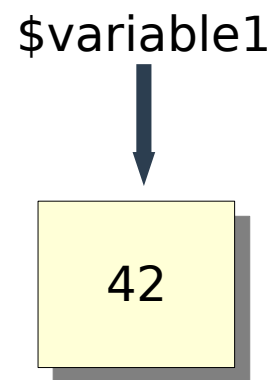
```
function maFonction(&$parametre1) {  
    // ...  
}
```

```
$variable1 = 42;  
maFonction($variable1);
```

# Passage par référence

```
$variable1 = 42;
```

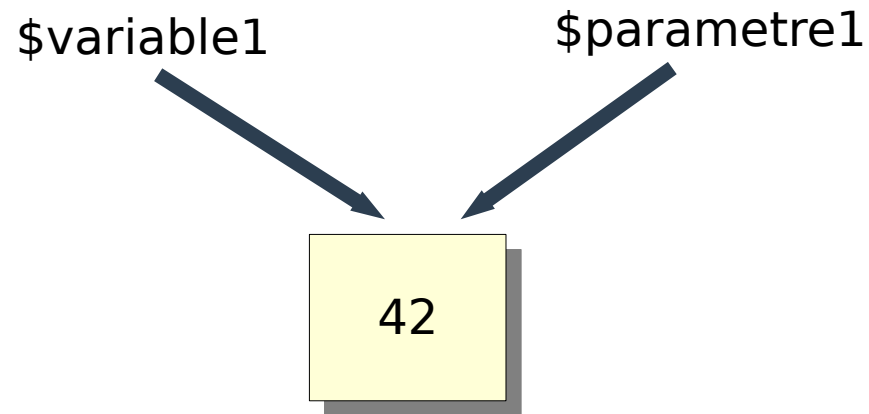
**En mémoire:**



# Passage par référence

`maFonction($variable1);` ← passe `$variable1` comme argument pour `$parametre1`

**En mémoire:**



# Passage par référence

- **C'est comme si on créait un deuxième numéro de téléphone pour une même personne**

# Modifier une variable reçue en paramètre

- Si le corps d'une fonction assigne une nouvelle valeur à un paramètre reçu **par valeur**, la variable d'origine **n'est pas modifiée**

```
function maFonction($parametre1) {  
    $parametre1 = "Allo";  
    echo $parametre1; // Affiche "Allo"  
}
```

```
$variable1 = "Bonjour";  
maFonction($variable1);  
echo $variable1; // Affiche "Bonjour"
```

# Modifier une variable reçue en paramètre

- **Si le corps d'une fonction assigne une nouvelle valeur à un paramètre reçu **par référence**, la variable d'origine **est modifiée****

```
function maFonction(&$parametre1) {  
    $parametre1 = "Allo";  
    echo $parametre1; // Affiche "Allo"  
}
```

```
$variable1 = "Bonjour";  
maFonction($variable1);  
echo $variable1; // Affiche "Allo"
```



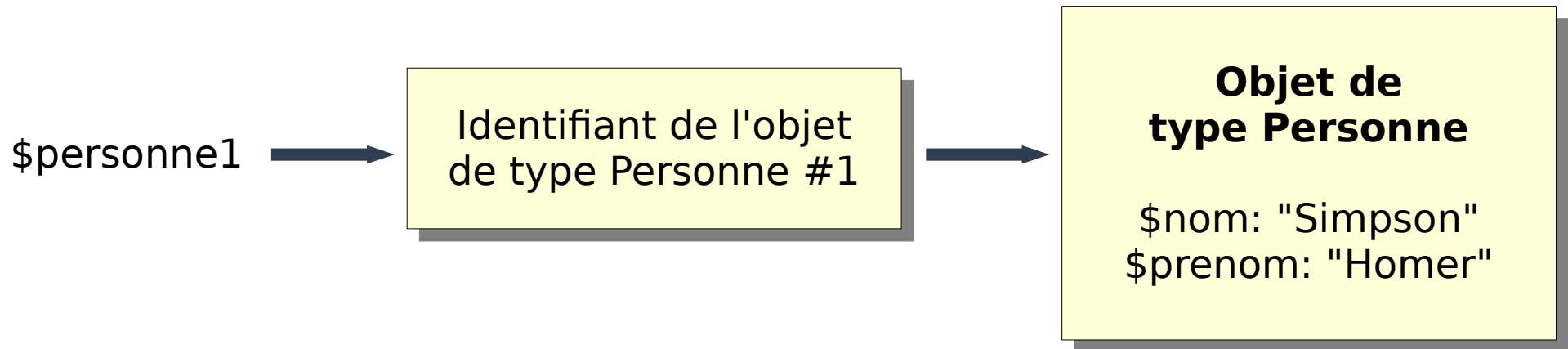
# Particularité pour les objets

# Particularité pour les objets

- **Quand on assigne un objet à une variable, la variable ne contient pas l'objet directement**
- **Elle contient l'identifiant de l'objet**
  - Concept du fonctionnement interne de PHP
  - Est lié à l'adresse mémoire, qui permet de savoir à quel endroit un objet est stocké dans la mémoire vive de l'ordinateur
  - Dans d'autres langages, on parlerait de **pointeur**

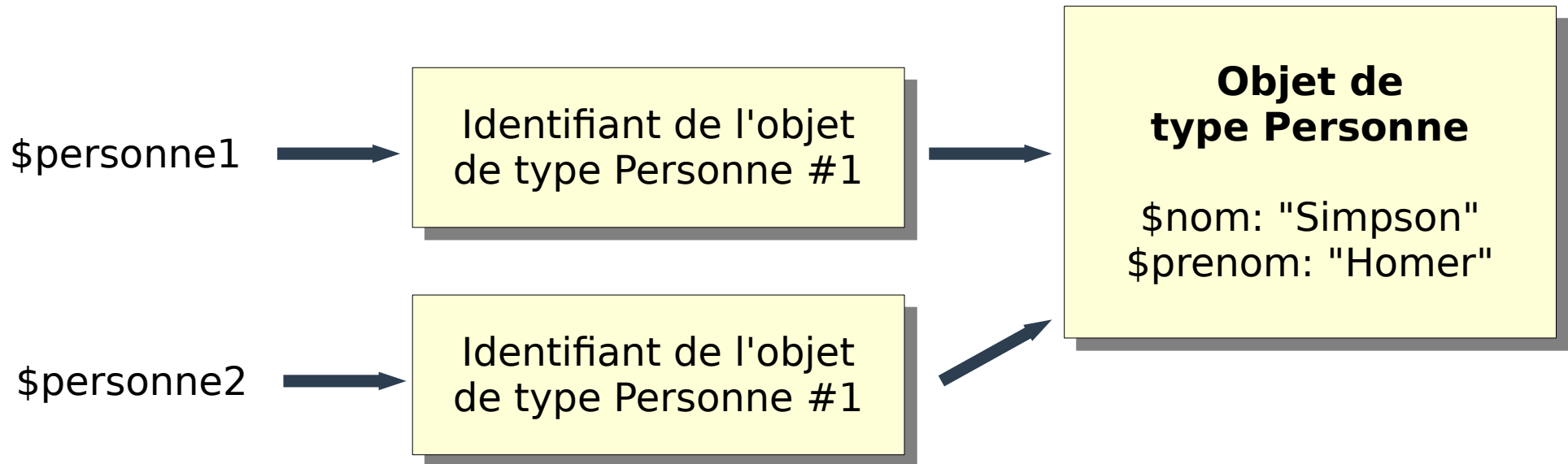
# Exemple

```
$personne1 = new Personne('Simpson', 'Homer');
```



# Exemple

```
$personne1 = new Personne('Simpson', 'Homer');  
$personne2 = $personne1;
```

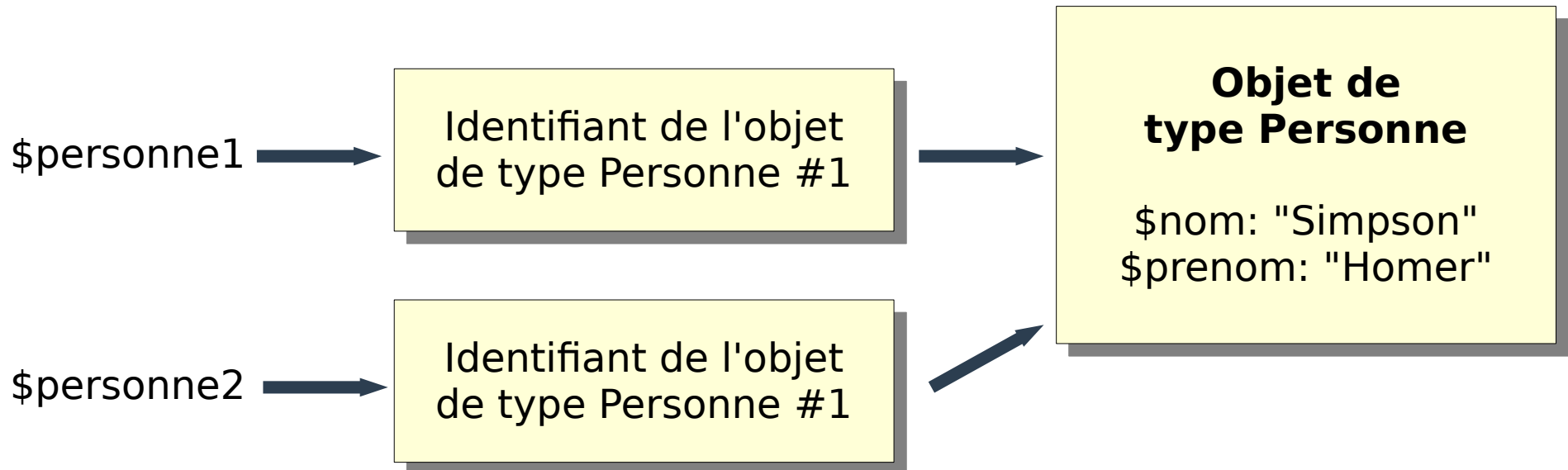


# Particularité pour les objets

- **Quand on passe un objet par valeur à une fonction, c'est l'identifiant qui est passé par valeur**
  - Si on modifie les attributs de l'objet reçu, ceux de l'objet d'origine sont modifiés
  - Si on assigne le paramètre à un nouvel objet, la variable d'origine ne changera pas

# Exemple

```
function maFonction($parametre1) { /* ... */ }  
$personne1 = new Personne('Simpson', 'Homer');  
maFonction($personne1);
```



# Exemple

```
function maFonction($parametre1) {  
    $parametre1->setPrenom('Bart');  
}
```

```
$personne1 = new Personne('Simpson', 'Homer');  
maFonction($personne1);  
echo $personne1->getPrenom(); // Affiche "Bart"
```

# Exemple

```
function maFonction($parametre1) {  
    $parametre1 = new Personne('Simpson', 'Bart');  
}  
  
$personne1 = new Personne('Simpson', 'Homer');  
maFonction($personne1);  
echo $personne1->getPrenom(); // Affiche "Homer"
```

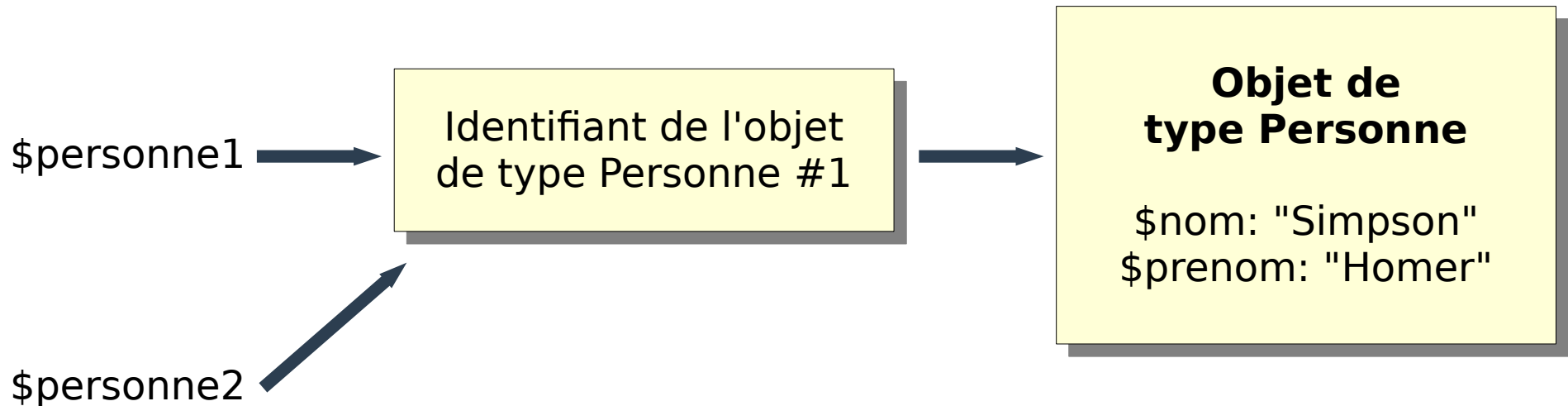


# Particularité pour les objets

- Quand on passe un objet par référence à une fonction, l'**identifiant** est passé par référence

# Exemple

```
function maFonction(&$parametre1) { /* ... */ }  
$personne1 = new Personne('Simpson', 'Homer');  
maFonction($personne1);
```



# Exemple

```
function maFonction(&$parametre1) {  
    $parametre1 = new Personne('Simpson', 'Bart');  
}  
  
$personne1 = new Personne('Simpson', 'Homer');  
maFonction($personne1);  
echo $personne1->getPrenom(); // Affiche "Bart"
```

**Et en JavaScript?**

# Et en JavaScript?

- **En JavaScript, il est impossible de passer des arguments par référence**
- **Le fonctionnement pour les objets est le même qu'en PHP**
- **Ce qui est vrai pour les objets est aussi vrai pour les tableaux, puisqu'ils sont des objets en JS**

# Fin de la présentation

Des questions?



Photo par Emily Morter sur Unsplash