

Les variables, paramètres et attributs en PHP

The slide has a blue header bar with the title 'Basic PHP Variables'. Below the header, there is a bulleted list of four items:

- The values stored in computer memory are called **variables**
- The values, or data, contained in variables are classified into categories known as **data types**
- The name you assign to a variable is called an **identifier**

En PHP, les termes *variable*, *paramètre* et *attribut* ont des significations distinctes, bien que parfois ils puissent être utilisés de manière interchangeable dans certains contextes. Voici un aperçu de ce que chacun de ces termes signifie :

1. Variable

The diagram shows a PHP assignment statement: `$name = value;`. It is divided into two main parts: 'VARIABLE NAME' (`$name`) and 'VARIABLE VALUE' (`value`). A bracket below the assignment operator is labeled 'EXPRESSION'. Another bracket spanning the entire line is labeled 'STATEMENT'.

Variable is a container for values. Equal operator is used to assign a value to a variable. When a value has been assigned to a variable, we can say the variable has been initialized.

All variable names in PHP are represented by a \$ (dollar sign), followed by the variable name. Variable names must start with a letter or underscore.

Variable names are case sensitive. Variable values can be modified, and overwritten anytime during the program.

Une *variable* en PHP est un conteneur qui stocke des données, que ce soit un nombre, une chaîne de caractères, un tableau, un objet, etc. Les variables en PHP sont toujours précédées du symbole \$.

Exemple de variable :

```
<?php  
$nom = "Alice";
```

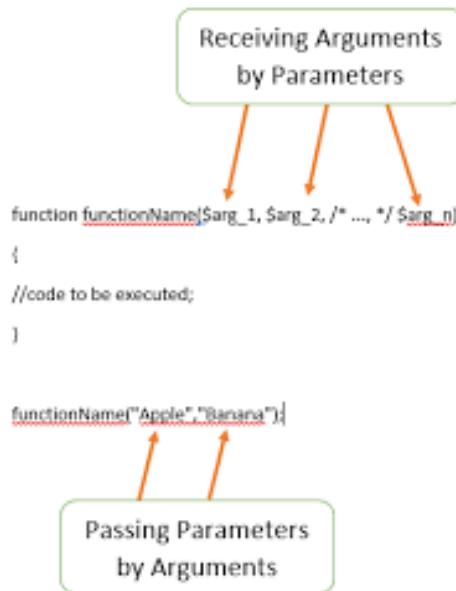
```

$age = 25;
$utilisateur = array("nom" => "Alice", "age" => 25);
?>

```

Dans cet exemple : - \$nom est une variable qui contient la chaîne "Alice".
- \$age est une variable qui contient le nombre 25. - \$utilisateur est une variable qui contient un tableau associatif.

2. Paramètre



Un *paramètre* est une variable qui est passée à une fonction ou à une méthode lorsqu'elle est appelée. Les paramètres permettent de transmettre des informations à une fonction pour qu'elle puisse les utiliser.

Exemple de paramètre :

```

<?php
function saluer($nom) {
    echo "Bonjour, " . $nom;
}

saluer("Alice"); // Appel de la fonction avec "Alice" comme paramètre
?>

```

Dans cet exemple : - \$nom est un paramètre de la fonction `saluer`. - Lors de l'appel de `saluer("Alice")`, "Alice" est passé en tant que paramètre.

3. Attribut (ou Propriété)



Un *attribut* (souvent appelé *propriété* en PHP) est une variable définie au sein d'une classe et qui représente les données ou l'état d'un objet. Les attributs sont accessibles via les instances de la classe.

Exemple d'attribut :

```
<?php
class Utilisateur {
    public $nom;
    public $age;

    public function __construct($nom, $age) {
        $this->nom = $nom;
        $this->age = $age;
    }

    public function afficherNom() {
        echo "Nom: " . $this->nom;
    }
}

$utilisateur = new Utilisateur("Alice", 25);
$utilisateur->afficherNom(); // Affiche "Nom: Alice"
?>
```

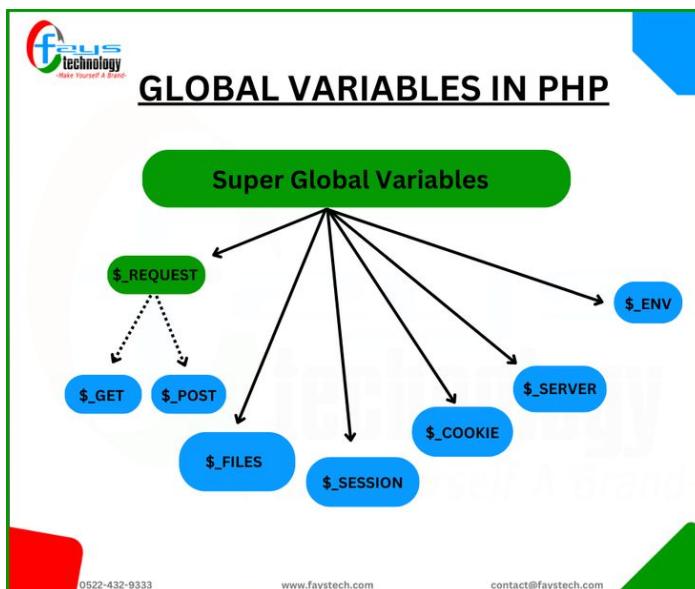
Dans cet exemple : - \$nom et \$age sont des attributs (ou propriétés) de la classe Utilisateur. - \$this->nom fait référence à l'attribut nom de l'instance actuelle de Utilisateur.

Résumé des différences :

- **Variable** : Conteneur pour stocker des données, peut être utilisée n'importe où dans le code.
- **Paramètre** : Spécialement utilisé dans les fonctions/méthodes pour transmettre des informations.
- **Attribut (Propriété)** : Variable définie au sein d'une classe, utilisée pour représenter l'état d'un objet.

Autres Données en PHP

- Variables Superglobales:



- Constantes (Invariables):



PHP Constants

```
define('TEXT', 'Hello World!');  
A constant can also be defined using const construct.  
<?php  
const MSG = "WELCOME";  
echo MSG;  
?>
```

www.educba.com

Conclusion

Ces concepts sont fondamentaux pour bien comprendre la programmation en PHP et leur utilisation appropriée est cruciale pour écrire un code clair et efficace.