



# PHP



Les bases du langage



# Dans ce module

- ❑ Syntaxe
  - ❑ *Balisage*
  - ❑ *Commentaire*
- ❑ Variable
  - ❑ *Variables*
  - ❑ *Assignations et types de variables*
  - ❑ *Caractères*
  - ❑ *Afficher et supprimer une variable*
  - ❑ *Concaténation*
- ❑ Différents codes
- ❑ Quelques fonctions
- ❑ Opérateurs
- ❑ Constantes
- ❑ Pratique

## Balisage

- Fichier en « .php » (index.php)
- Balises dans les fichiers :

Balise PHP avec HTML

```
<?php CODE ?>
```

```
<?php
```

```
CODE
```

```
?>
```

Balise avec uniquement du PHP

```
<?php
```

```
CODE
```

## Commentaire

- Commentaire sur 1 ligne :

```
// Comment  
# Comment
```

- Commentaire sur plusieurs lignes :

```
/*  
    Comment 1  
    Comment 2  
*/
```

- Les commentaires php sont supprimés à la compilation donc ils n'apparaissent pas dans l'HTML.

# Commentaire

```
<!DOCTYPE html>
<html>
<head>
  <title>Mon site</title>
</head>
<body>
  <!-- Commentaire HTML visible en front -->
  <?php
    // Commentaire developpeur
    // supprimé à la compilation
    /* On écrit les commentaires en anglais de préférence */
  ?>
</body>
</html>
```

## Commentaire

```
<!DOCTYPE html>
<html>
<head>
  <title>Mon site</title>
</head>
<body>
  <!-- Commentaire HTML visible en front -->
</body>
</html>
```

## Syntaxe d'instruction

- Un grand nombre d'instruction en PHP termine par un point-virgule « ; »
  - *C'est obligatoire pour définir une fin d'instruction.*
  - *Ne pas confondre avec le double-point « : »*
- D'autres instructions sont encadré par 2 accolades « { } »
  - *1 ouvrante et 1 fermante*
  - *Entre les 2 « { } » se trouve du code.*
  - *Dans ce cas l'instruction qui l'utilise n'a pas besoin de point-virgule, mais cela ne s'applique pas aux instructions contenu dans les accolades.*
- Chaque type d'instructions est donc différent.

## Variables

- Élément en informatique qui contient une valeur et doit être nommé.
  - *On définit un nom qui représentera/aura une valeur.*
- En PHP, leur nom doit respecter :
  - *Commencé obligatoirement par le symbole « \$ »*
  - *Contient des lettres (Aa-Zz), chiffres(0-9) ou underscores « \_ »*
  - *Ne pas commencer par des chiffres mais peut l'être par un underscore dans certains cas*
  - *Séparé par des underscores*
  - *Tous autres symboles ou séparateur est impossible*
- Sensible à la case



## Variables

```
// Valide
$var; $varTest; $var_test; // Usage normal
$_var; $Var; // Usage particulier
$VAR; $VAR_TEST; // Constante

// Non-valide
$125; $87var; $var-test;
$$var; $('var');
...
```

## Assignation et quelques Types de variables

```
// Number (Nombre)
$number_integer = 10; // integer (int) (Entier)
$number_float = 65.4; // float / double (Décimal)

// String (Texte ou chaîne de caractères)
    // Texte simple quote (guillemets simple)
$text_string = 'Mon texte en 2021';
$text_string = 'Mon texte d\'avant 2021';
    // Texte double quote (guillemets double)
$text_string = "Mon texte en 2021";
$text_string = "Mon texte d'avant 2021";

// Texte "nombre"
$text_string = '10'; // Attention, ce n'est pas le nombre 10
                // mais les caractères 1 et 0 à la suite
$text_string = '65.4'; // Attention, ce n'est pas le nombre 65.4
                // mais les caractères 6, 5, point et 4 à la suite
```

# Les caractères en informatique

| Dec | Hex | Oct | Chr                 | Dec | Hex | Oct | HTML   | Chr   | Dec | Hex | Oct | HTML   | Chr | Dec | Hex | Oct | HTML   | Chr |
|-----|-----|-----|---------------------|-----|-----|-----|--------|-------|-----|-----|-----|--------|-----|-----|-----|-----|--------|-----|
| 0   | 0   | 000 | NULL                | 32  | 20  | 040 | &#032; | Space | 64  | 40  | 100 | &#064; | @   | 96  | 60  | 140 | &#096; | `   |
| 1   | 1   | 001 | Start of Header     | 33  | 21  | 041 | &#033; | !     | 65  | 41  | 101 | &#065; | A   | 97  | 61  | 141 | &#097; | a   |
| 2   | 2   | 002 | Start of Text       | 34  | 22  | 042 | &#034; | "     | 66  | 42  | 102 | &#066; | B   | 98  | 62  | 142 | &#098; | b   |
| 3   | 3   | 003 | End of Text         | 35  | 23  | 043 | &#035; | #     | 67  | 43  | 103 | &#067; | C   | 99  | 63  | 143 | &#099; | c   |
| 4   | 4   | 004 | End of Transmission | 36  | 24  | 044 | &#036; | \$    | 68  | 44  | 104 | &#068; | D   | 100 | 64  | 144 | &#100; | d   |
| 5   | 5   | 005 | Enquiry             | 37  | 25  | 045 | &#037; | %     | 69  | 45  | 105 | &#069; | E   | 101 | 65  | 145 | &#101; | e   |
| 6   | 6   | 006 | Acknowledgment      | 38  | 26  | 046 | &#038; | &     | 70  | 46  | 106 | &#070; | F   | 102 | 66  | 146 | &#102; | f   |
| 7   | 7   | 007 | Bell                | 39  | 27  | 047 | &#039; | '     | 71  | 47  | 107 | &#071; | G   | 103 | 67  | 147 | &#103; | g   |
| 8   | 8   | 010 | Backspace           | 40  | 28  | 050 | &#040; | (     | 72  | 48  | 110 | &#072; | H   | 104 | 68  | 150 | &#104; | h   |
| 9   | 9   | 011 | Horizontal Tab      | 41  | 29  | 051 | &#041; | )     | 73  | 49  | 111 | &#073; | I   | 105 | 69  | 151 | &#105; | i   |
| 10  | A   | 012 | Line feed           | 42  | 2A  | 052 | &#042; | *     | 74  | 4A  | 112 | &#074; | J   | 106 | 6A  | 152 | &#106; | j   |
| 11  | B   | 013 | Vertical Tab        | 43  | 2B  | 053 | &#043; | +     | 75  | 4B  | 113 | &#075; | K   | 107 | 6B  | 153 | &#107; | k   |
| 12  | C   | 014 | Form feed           | 44  | 2C  | 054 | &#044; | ,     | 76  | 4C  | 114 | &#076; | L   | 108 | 6C  | 154 | &#108; | l   |
| 13  | D   | 015 | Carriage return     | 45  | 2D  | 055 | &#045; | -     | 77  | 4D  | 115 | &#077; | M   | 109 | 6D  | 155 | &#109; | m   |
| 14  | E   | 016 | Shift Out           | 46  | 2E  | 056 | &#046; | .     | 78  | 4E  | 116 | &#078; | N   | 110 | 6E  | 156 | &#110; | n   |
| 15  | F   | 017 | Shift In            | 47  | 2F  | 057 | &#047; | /     | 79  | 4F  | 117 | &#079; | O   | 111 | 6F  | 157 | &#111; | o   |
| 16  | 10  | 020 | Data Link Escape    | 48  | 30  | 060 | &#048; | 0     | 80  | 50  | 120 | &#080; | P   | 112 | 70  | 160 | &#112; | p   |
| 17  | 11  | 021 | Device Control 1    | 49  | 31  | 061 | &#049; | 1     | 81  | 51  | 121 | &#081; | Q   | 113 | 71  | 161 | &#113; | q   |
| 18  | 12  | 022 | Device Control 2    | 50  | 32  | 062 | &#050; | 2     | 82  | 52  | 122 | &#082; | R   | 114 | 72  | 162 | &#114; | r   |
| 19  | 13  | 023 | Device Control 3    | 51  | 33  | 063 | &#051; | 3     | 83  | 53  | 123 | &#083; | S   | 115 | 73  | 163 | &#115; | s   |
| 20  | 14  | 024 | Device Control 4    | 52  | 34  | 064 | &#052; | 4     | 84  | 54  | 124 | &#084; | T   | 116 | 74  | 164 | &#116; | t   |
| 21  | 15  | 025 | Negative Ack.       | 53  | 35  | 065 | &#053; | 5     | 85  | 55  | 125 | &#085; | U   | 117 | 75  | 165 | &#117; | u   |
| 22  | 16  | 026 | Synchronous idle    | 54  | 36  | 066 | &#054; | 6     | 86  | 56  | 126 | &#086; | V   | 118 | 76  | 166 | &#118; | v   |
| 23  | 17  | 027 | End of Trans. Block | 55  | 37  | 067 | &#055; | 7     | 87  | 57  | 127 | &#087; | W   | 119 | 77  | 167 | &#119; | w   |
| 24  | 18  | 030 | Cancel              | 56  | 38  | 070 | &#056; | 8     | 88  | 58  | 130 | &#088; | X   | 120 | 78  | 170 | &#120; | x   |
| 25  | 19  | 031 | End of Medium       | 57  | 39  | 071 | &#057; | 9     | 89  | 59  | 131 | &#089; | Y   | 121 | 79  | 171 | &#121; | y   |
| 26  | 1A  | 032 | Substitute          | 58  | 3A  | 072 | &#058; | :     | 90  | 5A  | 132 | &#090; | Z   | 122 | 7A  | 172 | &#122; | z   |
| 27  | 1B  | 033 | Escape              | 59  | 3B  | 073 | &#059; | ;     | 91  | 5B  | 133 | &#091; | [   | 123 | 7B  | 173 | &#123; | {   |
| 28  | 1C  | 034 | File Separator      | 60  | 3C  | 074 | &#060; | <     | 92  | 5C  | 134 | &#092; | \   | 124 | 7C  | 174 | &#124; |     |
| 29  | 1D  | 035 | Group Separator     | 61  | 3D  | 075 | &#061; | =     | 93  | 5D  | 135 | &#093; | ]   | 125 | 7D  | 175 | &#125; | }   |
| 30  | 1E  | 036 | Record Separator    | 62  | 3E  | 076 | &#062; | >     | 94  | 5E  | 136 | &#094; | ^   | 126 | 7E  | 176 | &#126; | ~   |
| 31  | 1F  | 037 | Unit Separator      | 63  | 3F  | 077 | &#063; | ?     | 95  | 5F  | 137 | &#095; | _   | 127 | 7F  | 177 | &#127; | Del |

asciichars.com

- Ascii sur 7 bits
- Latin-1 sur 8 bits
- UTF-8 sur 8 bits
- UTF-16 sur 16 bits
- ...

## Afficher ou supprimer une variable

- Afficher une variable ou du texte avec la fonctions « echo() » ou « echo » :

```
<h1>Mon site</h1>
<p>
    <?php
        $my_message = 'Hello people !';
        echo "My story :<br>";
        echo $my_message;
    ?>
</p>
```

- Détruire une variable avec la fonctions « unset() » :

```
$myVar = "Hello";
unset($myVar);
// $myVar n'existe plus
```

## Concaténation

- Pour concaténer (assembler) plusieurs variables on utilise le « point » comme suit :

```
<h1>Mon site</h1>
<p>
  <?php
    $my_message = 'Hello people !';
    echo "My story :<br>" . $my_message;
    // équivalent, sans espaces
    echo "My story :<br>".$my_message;
  ?>
</p>
```

```
$my_message = 'Message sans point de grammaire';
$my_end = "Je commence PHP.";
echo "My story :<br>".$my_message.' ' . $my_end;
```

## Différences ou pas

```
<h1>Mon site</h1>
<p>
  <?php
    $my_message = 'Hello people !';
    echo "My story :<br>" . $my_message;
  ?>
</p>
```

```
<h1>Mon site</h1>
<p>
  <?php
    $my_message = 'Hello people !';
    echo "My story :<br>";
    echo $my_message;
  ?>
</p>
```

```
<?php $my_message = 'Hello people !'; ?>
<h1>Mon site</h1>
<p>
  My story : <br> <?php echo $my_message; ?>
</p>
```

## Quelques fonctions en plus

```
$my_var = 1027.621;
$my_var_null = null;

// Development display
var_dump($my_var); // float 1027.621
var_dump($my_var_null); // null

// Type check
var_dump( is_int($my_var) ); // false
var_dump( is_float($my_var) ); // true
var_dump( is_string($my_var) ); // false

// Set check
var_dump( isset($my_var) ); // true
var_dump( isset($my_var_null) ); // false
var_dump( isset($myVarTest) ); // false

// Null check
var_dump( is_null($my_var) ); // false
var_dump( is_null($my_var_null) ); // true
```

```
// Get type
var_dump( gettype($my_var) ); // string(6) "double"
var_dump( gettype( strval($my_var) ) ); // string(6) "string"
```

```
// Conversion
var_dump( intval($my_var) ); // int 1027
var_dump( strval($my_var) ); // string '1027.621' (length=8)
var_dump( gettype( $my_var ) ); // string 'double' (length=6)
```



## Quelques fonctions en plus

```
echo round(7.6214); // 8
echo round(7.6214, 1); // 7.6
echo round(7.6214, 3); // 7.621
```

```
echo round(7.628, 2); // 7.63
echo round(7.625, 2); // 7.63
echo round(7.621, 2); // 7.62
```

```
// Formate un nombre avec des paramètres choisis :
// 1. un nombre de décimal
// 2. un séparateur décimal
// 3. un séparateur des milliers
echo number_format($my_var, 2, ',', ''); // 1027,62
```

```
// Affichage pour le développement uniquement
var_dump($my_var); // float(1027.621)
var_dump( strval($my_var) ); // string(8) "1027.621"
```



## Les constantes magiques

| Constante existante       | Valeur  |
|---------------------------|---|
| <code>__LINE__</code>     | Retourne la ligne courante                    |
| <code>__FILE__</code>     | Retourne le nom du fichier courant            |
| <code>__DIR__</code>      | Retourne le chemin d'accès au dossier courant |
| <code>__FUNCTION__</code> | Retourne le nom de la fonction courante       |
| <code>__CLASS__</code>    | Retourne le nom de la classe                  |

## Les opérateurs arithmétiques

| Symbole | Exemple                | Equivalent    | Opérateur                     |
|---------|------------------------|---------------|-------------------------------|
| +       | \$price = \$price + 45 | \$price += 45 | Addition                      |
| -       | \$price = \$price - 45 | \$price -= 45 | Soustraction                  |
| *       | \$price = \$price * 2  | \$price *= 2  | Multiplication                |
| **      | \$price = \$price ** 2 | \$price **= 2 | Puissance                     |
| /       | \$price = \$price / 2  | \$price /= 2  | Division                      |
| %       | \$price = \$price % 2  | \$price %= 2  | Modulo (reste de la division) |

## Opérateur d'incrémentation et décrémentation

| Symbole | Exemple     | Explication                      |
|---------|-------------|----------------------------------|
| ++      | \$student++ | Incrémentation après utilisation |
| ++      | ++\$student | Incrémentation avant utilisation |
| --      | \$student-- | Décrémentation après utilisation |
| --      | --\$student | Décrémentation avant utilisation |

# Pratique

- ❖ Essayez-vous à réaliser les différentes démonstrations du cours et à les modifier pour mieux comprendre les possibilités et limites.
- ❖ Ensuite, aller sur le site [W3Schools PHP](https://www.w3schools.com/php/) et découvrez de nouvelles bases de PHP. Il n'est pas nécessaire de tout lire mais vous pouvez essayer directement sur le site en cliquant sur les « Try it » ou « Try it Yourself ». Utiliser le menu de gauche pour faire :
  1. PHP Syntax
  2. PHP Comments
  3. PHP Variables
  4. PHP Echo / Print
  5. PHP Data Types
  6. PHP Strings
  7. PHP Numbers
  8. PHP Constants
  9. PHP Operators
  10. PHP If...Else...Elseif
- ❖ Essayer-vous au Quiz : [W3Schools PHP Quiz](https://www.w3schools.com/php/quiz.asp)
- ❖ Essayer-vous aux exercices : [W3Schools PHP Exercices](https://www.w3schools.com/php/exercises.asp)



# PHP

Les bases du langage

Fin du module

