# Code TW2-1

< ?php  ..code PHP.. ?> : bloc de php

**date_default_timezone_set('Europe/Paris')** ; : précise le fuseau horaire à utiliser.

**date('H :i :s')** : affiche l'heure 13 :03 :57
Y : 1996 – y : 96

**echo $args, ...** : une ou plusieurs chaînes de car
$n=1234 ;
echo ''le carre de '', $n , '' est '' , $n*$n ;
➔ le carre de 1234 est …

**print($arg)** : une chaine de de caractère

**printf($format, $args…)** : chaine de caractère formatée
printf(''le carre de %d est %d, $n,$n*$n) ;
➔le carre de 1234 est …

if (cond) {instr ;}
elseif (cond) {instr ;}
else {instr ;}

While (cond)
  {instr ;}

for ($i = 0 ; $i<=6 ;$i++)
  {instr ;}

== : égalité simple
=== : égalité des valeurs et des types

Commentaires : /* …. */ ou //

Les constantes :
**const** PI_APPROX = 3.14
ou
**define**(PI_APPROX, 3.14) ;

Ex :
$i = 355 ;
$s = '' La valeur de \$i est $i ''
➔ La valeur de $i est 355

$s = '' La valeur de \$i est {$i} ''
➔ La valeur de $i est 355

$i = 2 ;
$t[2]=666 ;
$s = ''dans la case \$t[{$i}] : {$t[$i]}'' ;
➔ Dans la case $t[2] : 666 ;

**Operateur de Comparaison :**
Egal : ==
Different : != ou <>
Relation d'ordre : <=, <, >=, >
Identité : === (même valeur et type)

**Operateurs logiques :**
Et : &&  and
Ou : ||  or
Non : !
Ou exclusif : xor
➔ Parenthéser !!

**RQ :**
-Pour un opérateur **arithmétique**, les 2 opérandes sont d'abord converties en **nombres**
-Pour la **concaténation**, sont d'abord converties en **chaine**.
-Pour un opérateur **logique**, sont d'abord converties en **booléen**.

**Conversion explicite (cast) :**
Ex :
$res = 12 ;
$s = ''Résultat : '' **.(string)** $res :
//équivalent à ''Résultat : 12''

**Délimitées par simples quotes**
- Seulement 2 séquences « spéciales » : \' qui signifie '
  \\ qui signifie \
- Tous les autres caractères sont traités normalement.

**Délimitées par doubles quotes**
- Les séquences spéciales reconnues :

| | |
|---|---|
| \" | " |
| \\ | \ |
| \n | newline |
| \t | tabulation |
| \$ | $ |
| $ | début de variable |
| \nb octal ou hexa | caractère de ce code ascii |

- **Les noms de variables sont remplacés par la valeur**

- '' \'' '' ➔ ''
- '' \'' '' ➔ \''

| Booléen | → Numérique | → Chaîne |
|---|---|---|
| TRUE | → 1 | → "1" |
| FALSE | → 0. | → "" |
| **Numérique** | **→ Booléen** | **→ Chaîne** |
| x | → x != 0 | → représentation décimale de x. |
| **Chaîne** | **→ Booléen** | **→ Numérique** |
| s | → s!="" && s!="0" | → *n* : si s commence par la représentation du nombre *n*. |
| | → | → 0 : sinon. |
| **Flottant** | **→ Entier** | |
| x | → arrondi "vers 0" de x | |

# Code TW2-1

**isset($v)** : teste si $v est définie
**unset($v)** : supprime $v

**les assertions de type :**
**is_int($v), is_string($v), is_array($v), is_object($v)**

**Les tables :**
$fibo = array(0=>1, 1=>1, 4=>2) ;
//eq à array(1,1,2)

Prolongation de la table :
$racineExacte[25]=5 ;

**count(table) :** nombre d'elt dans un tableau

**RQ :**
Si ex t[$i] n'existe pas : NULL est retournée

**unset($fibo[6])** : supprime l'elt 6 de la table
! les autres associations key value ne sont perturbées.

**la boucle foreach :**
-foreach (nomTable as varClé => varValeur)
-foreach (nomTable as varValeur
Ex :
foreach ($fibo as $nombreFibo)
$nombreFibo = 0
//ne modifie pas le tableau
**Mais :**
foreach ($fibo as $indice=>$nombreFibo)
$fibo[$indice] = 0
//modifie le tableau

**Tris des tables :**
**asort(table) :** ordre ↗ des valeurs
**ksort(table) :** ordre ↗ des clés
**arsort(table) :** ordre ↘ des valeurs
**krsort(table) :** ordre ↘ des clés.

**Modifient les clés (les associations) :**
**sort(table) :** tri ↗ des valeurs avec réindexation à partir de 0.
**rsort(table) :** tri ↘ des valeurs avec réindexation à partir de 0.

**Table de table :**
$cnp= array(array(1),array(1,1), array(1,2,1)) ;
echo $cnp[2][1] ; //=2

**RQ :**
Une variable globale **n'est pas visible à l'intérieur** d'une fonction.
Pour y accéder on doit la déclarer avec **global.**

```
$a=12; $b=20;
function print_ab(){
  $a=0;
  echo "|".$a."| |".$b."|";
}
print_ab();
echo "|".$a."| |".$b."|";

//affiche |0| || |12| |20|
```

Avec **global :**

```
$a=12; $b=20;
function print_ab(){
  global $a;
  echo "|".$a."| |".$b."|";
  $a=100;
}
print_ab();
echo "|".$a."| |".$b."|";
//affiche |12| || |100| |20|
```

**Variables super globales :**
➔Commencent par $_ et s'écrivent en maj :
$_REQUEST, $_GET, $_PUT, $_SESSION

**Variable rémanente :**

- Précédée de **static**
- **Initialisation** uniquement lors du premier appel
- Reste une variable locale à la fonction

```
function testRemanent(){
  static $a = 0;
  echo "|".$a."|";
  $a++;
}
testRemanent(); testRemanent(); testRemanent();
// affiche |0| |1| |2|
```