

# Cours Système Web PHP

## Corrigé TD5

### PHP, un langage de programmation

14 novembre 2005

## 1 Introduction

## 2 Bases

```
1. function bissextile($a) {
    if ($a%4==0) {
        if ($a%100==0) {
            if ($a%400==0)
                return true;
            else
                return false;
        } else
            return true;
    } else
        return false;
}
```

```
2. function factorielle($n)
{
    $result=1;
    for ($i=1;$i<=$n;++$i) {
        $result*=$i;
    }
    return $result;
}
```

```
3. function factorielle($n)
{
    if ($n<=0)
        return 1;
    else
        return $n*factorielle($n-1);
}
```

4. – Il faut passer par une variable intermédiaire pour échanger.  
– Les arguments de la fonction étant passés par défaut par valeur, les changements ne sont pas répercutés sur les variables sur laquelle la fonction est appelée.

```
– fonction echange(&$a,&$b)
{
    $temp=$b;
    $b=$a;
    $a=$temp;
}
```

### 3 Interpolation

1. `$a=42;`

```
echo "J'ai_$a_ans";
```

2. `echo "J'ai_". $a. "_ans";`

Cette méthode est plus générale (on peut faire des calculs dans la concaténation), mais moins concise.

3. À l'intérieur de guillemets simples, aucune *interpolation* de variable n'est effectuée. De plus, les seuls caractères spéciaux à l'intérieur de guillemets simples sont `\'` et `\\` (par exemple, `\n` n'est pas un caractère spécial à l'intérieur de guillemets simples).

### 4 Manipulation de chaînes de caractères

```
function remplace($a,$b,$c) {
    $position=strpos($c,$a,0);
    while($position!=false) { /* ($position) ou ($position!=false) ne
                               marcherait pas car strpos peut renvoyer 0
                               comme indice valide */
        $c=substr($c,0,$position).$b.substr($c,$position+strlen($a));
        $position=strpos($c,$a,$position+strlen($b));
    }
    return $c;
}
```

### 5 Production de code HTML

```
function table_multiplication($n)
{
    echo "<table>";
    echo "<caption>Table_de_multiplication_de_1_à_$n</caption>";
    echo "<tr>";
    echo "<th></th>";
    for($i=1;$i<=$n;$i=$i+1) {
        echo "<th>$i</th>";
    }
    echo "</tr>";
    for($i=1;$i<=$n;$i=$i+1) {
        echo "<tr><th>$i</th>";
        for($j=1;$j<=$n;$j=$j+1) {
            echo "<td>".($i*$j)."</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}
```

### 6 Tableaux

1. `function affiche_tableau($assoc)`

```
{
    echo "<table>";
    echo "<tr><th>Clef</th><th>Valeur</th></tr>";
    foreach($assoc as $cle => $valeur) {
        echo "<tr><td>$cle</td><td>$valeur</td></tr>";
    }
}
```

```

    }
    echo "</table>";
}

2. function affiche_tableau($assoc)
{
    echo "<table>";
    echo "<tr><th>Clef</th><th>Valeur</th></tr>";
    foreach($assoc as $cle => $valeur) {
        if(is_array($valeur))
            $valeur=implode(", ", $valeur);

        echo "<tr><td>$cle</td><td>$valeur</td></tr>";
    }
    echo "</table>";
}

```

3. Tri par insertion :

```

function tri_insertion($tab)
{
    $N=count($tab);
    for($i=0;$i<$N;++$i) {
        for($j=0;$j<$i;++$j) {
            if($newtab[$j]>$tab[$i])
                break;
        }

        for($k=$i;$k>$j;--$k)
            $newtab[$k]=$newtab[$k-1];

        $newtab[$j]=$tab[$i];
    }

    return $newtab;
}

```