

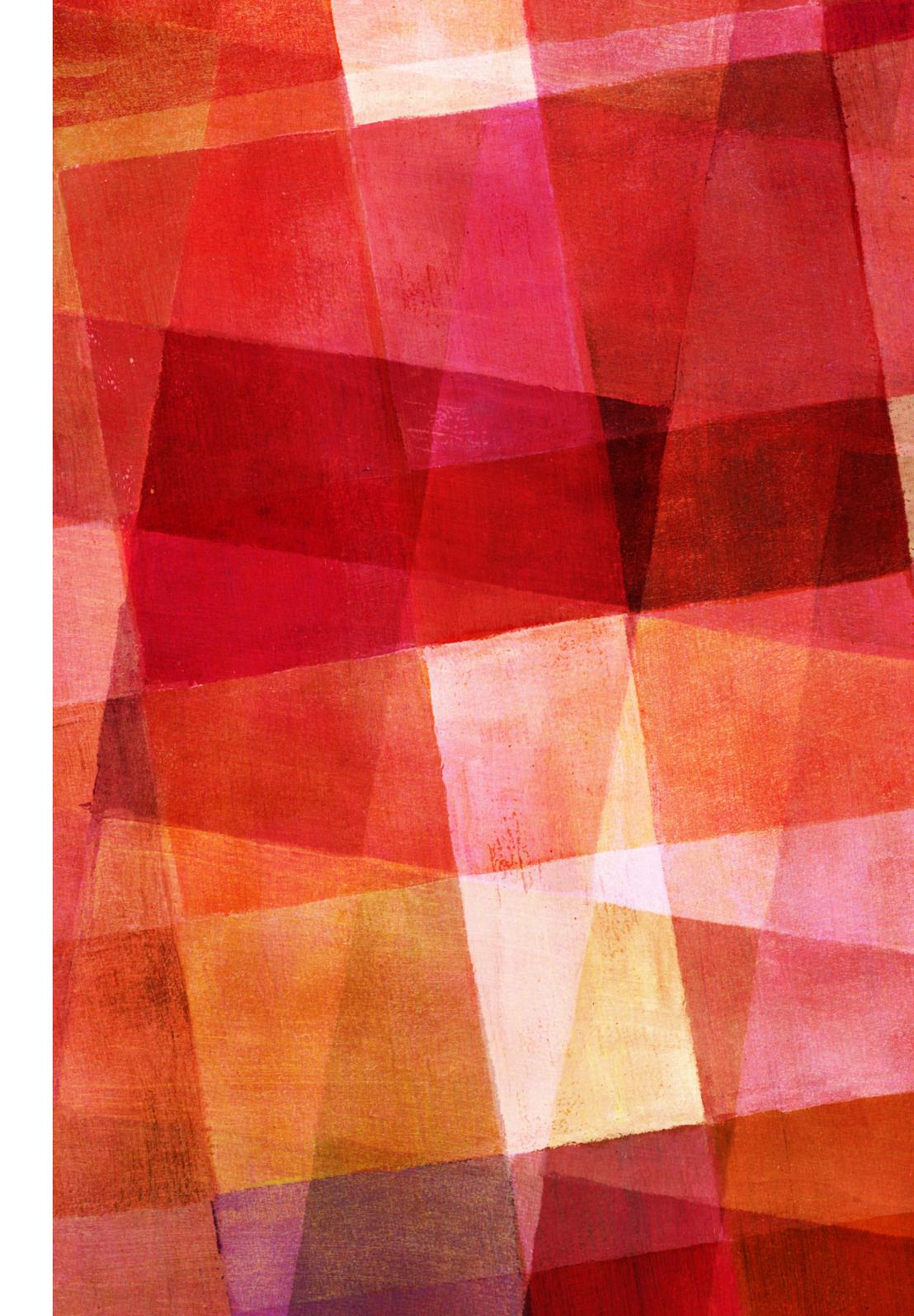
# PHP – FONCTION, BOUCLES ET VARIABLES

---

# REMARQUES SUR LE DERNIER EXERCICE DU TD3

---

- il faut créer une variable pour la note
- il faut réfléchir pour les questions ouvertes ;-)
- il est possible de faire un commentaire « évolutif »



# AUJOURD'HUI ET ...

---

1.habitudes à prendre

2.Fonctions

3.les boucles

1.for

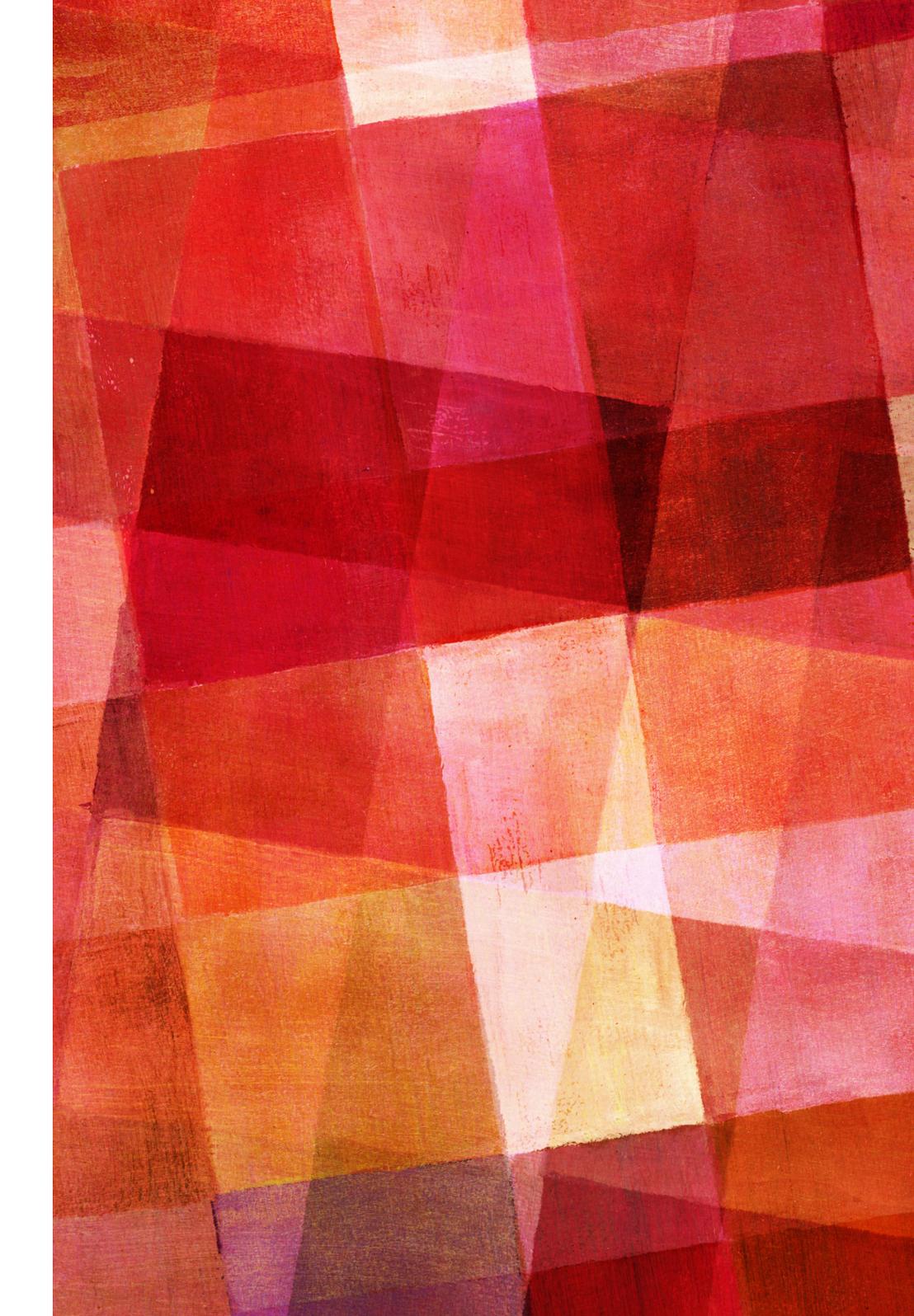
2.while

4. la portée des variables

# LES BONNES HABITUDES À PRENDRE

---

- ranger !!!!
- mettre le php dans du html afin de garder css, encodage, etc.
- regrouper les « déclarations de variables » et les « déclarations de fonctions » en début de programme
- mettre des commentaires
  - // quand sur une ligne
  - /\* \*/ quand sur plusieurs lignes



# AUJOURD'HUI ET ...

---

1.habitudes à prendre

2.Fonctions

3.les boucles

1.for

2.while

4. la portée des variables

# LES FONCTIONS

---

- Série d'instructions qui effectuent des actions et retourne une valeur
- Se définit en début de programme
- S'appelle dans le corps du programme
- Permet de ne pas répéter x fois une même série d'instructions

# EXEMPLE DE FONCTION

---

```
<?php
```

```
// déclaration de la fonction
```

```
function division ($params1, $params2) {
```

```
$resultat=$params1/$params2;
```

```
return ($resultat); }
```

```
....
```

```
// appel de la fonction avec le résultat mis dans une variable
```

```
$sous=200;
```

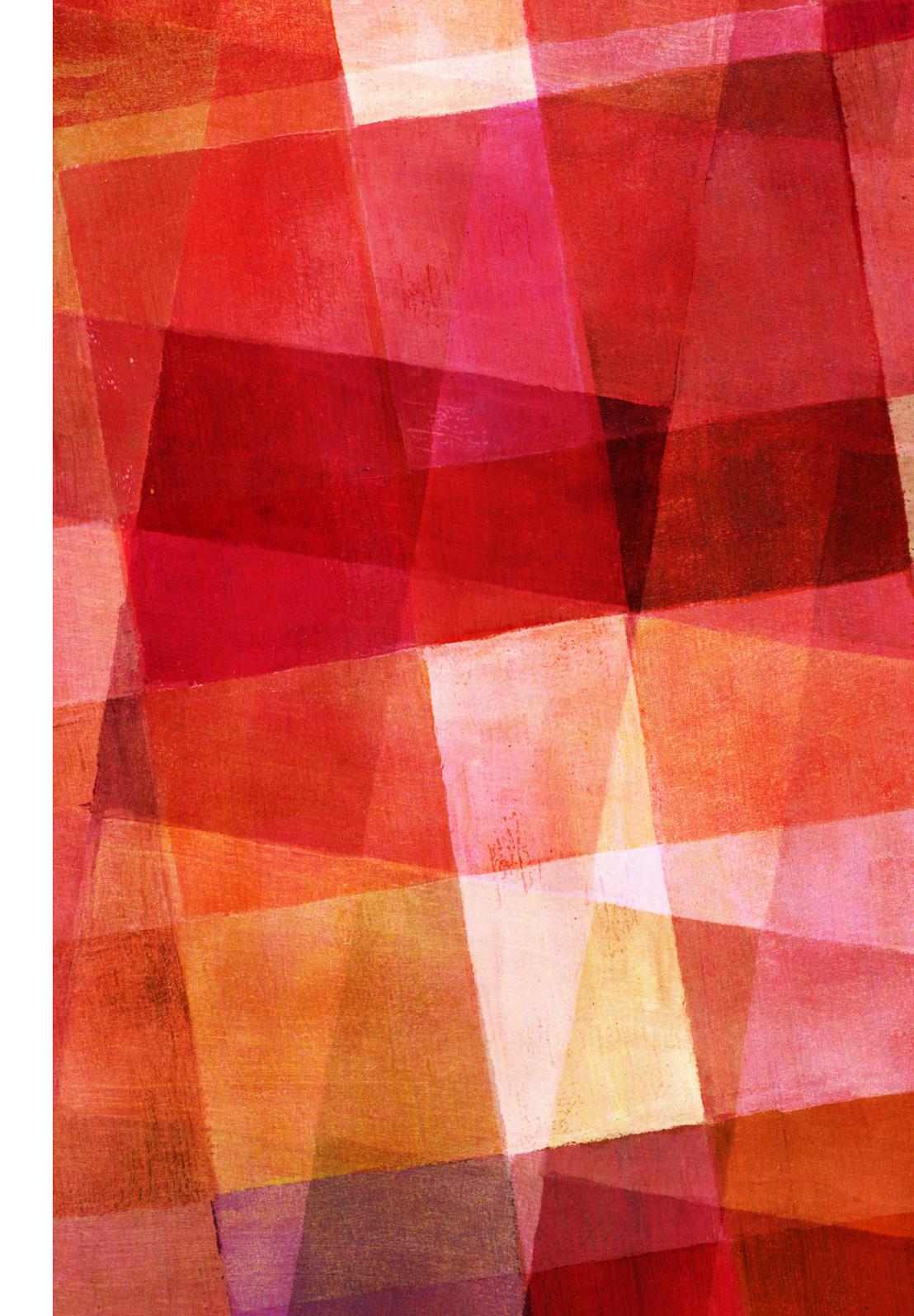
```
$membres=10;
```

```
$div=division ($sous, $membres) ;
```

```
Print "les " . $membres . "auront chacun " . $div . " des " . $sous . " récoltés ce we.
```

```
“;
```

```
?>
```



# AUJOURD'HUI ET ...

---

1.habitudes à prendre

2.Fonctions

3.les boucles

1.for

2.while

4. la portée des variables

# LES BOUCLES : LE « FOR »

---

► for s'utilise quand on sait par avance combien de boucle on va faire

► syntaxe en php :

for (condition initiale ; condition nécessaire à la poursuite de la boucle ; opération à effectuer avant le nouveau test de condition)

```
{  
    instruction  
}
```

► exemple

```
<?php  
print « je sais compter jusqu'à 10 ... </br> »  
for($i=1; $i<=10; $i++)  
{  
    print « $i </br> »;  
}  
?>
```

# LES BOUCLES : LE « WHILE »

---

► s'utilise quand on attend une réponse précise

► syntaxe :

```
while (condition)
```

```
{
```

```
    instructions
```

```
}
```

► exemple :

```
<? php
```

```
    print « quel est le numéro correspondant au département de l'Isère ? »;
```

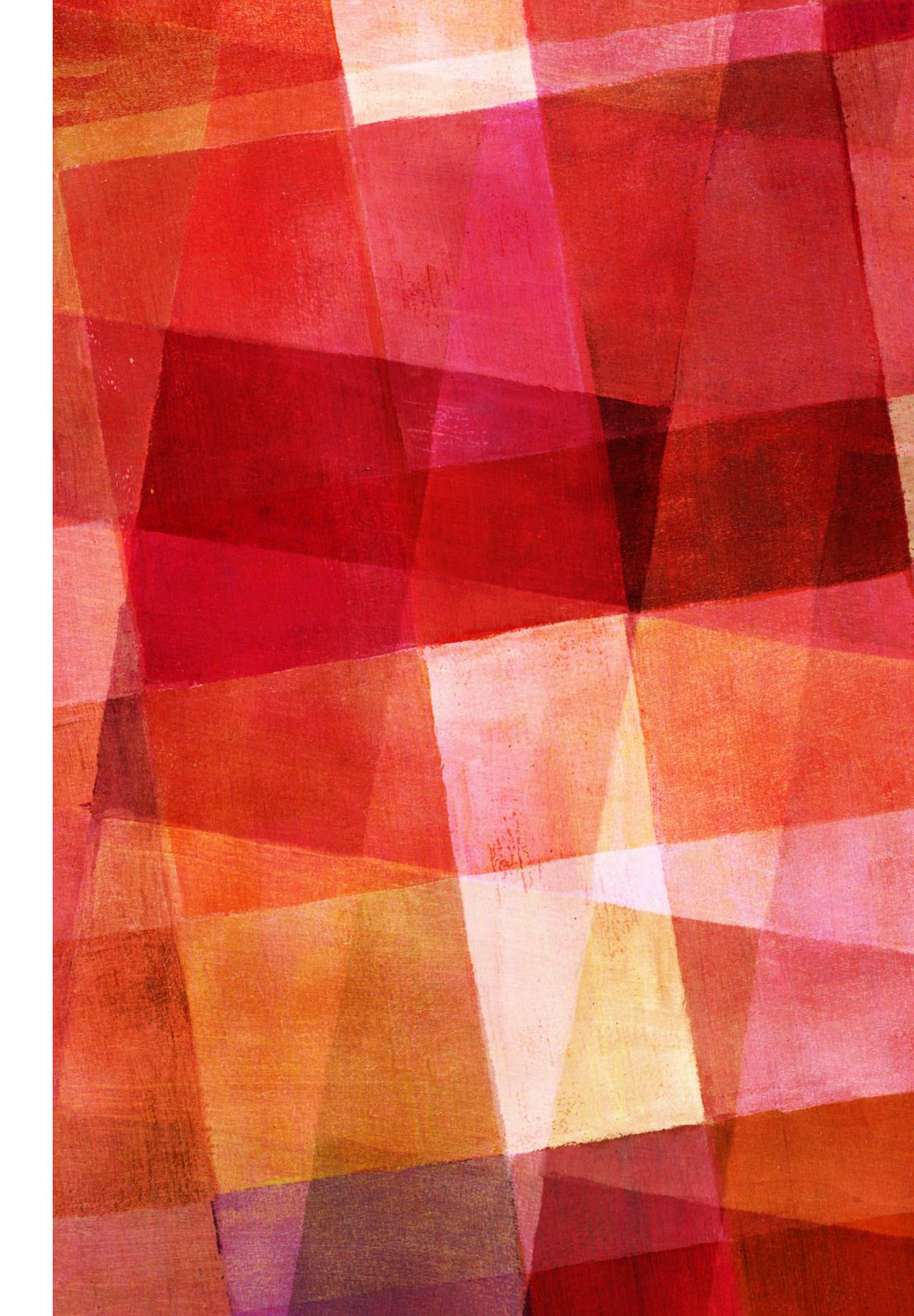
```
    while ($dep <> 38)
```

```
    {
```

```
        print « Ce n'est pas la bonne réponse ... encore une fois ? »;
```

```
        ....
```

```
    }
```



# AUJOURD'HUI ET ...

---

1.habitudes à prendre

2.les boucles

1.for

2.while

3. la portée des variables

# LA PORTÉE DES VARIABLES

---

- les variables peuvent être : locale, globale, de session
  - variable locale
    - sa valeur n'existe que localement (à l'intérieur d'une fonction par exemple)
  - variable globale
    - il faut indiquer que c'est une variable globale
    - sa valeur existe dans ensemble du programme (du fichier)
  - variable de session
    - il faut indiquer que c'est une variable de session
    - sa valeur existe durant toute la session et dans tous les fichiers liés

# VARIABLE LOCALE

---

➤ que vaut **\$i** ?

```
<html>
  <head> exo1</head>
  <body>
    <?php
      $i=1;
      function addition ()
      {
        $i=2;
        $j=3;
        $res=2+3;
        return $res;
      }
      //appel de la fonction
      $resultat = addition();
      print "le résultat de l'addition est $resultat </br> et i vaut $i" ;
    ?>
  </body>
</html>
```

# PORTÉE DES VARIABLES ... SUITE ...

---

- qu'est ce qui va être affiché ?

```
<html>
  <head> <title>exo1-bis</title></head>
  <body>
    <?php
      $a = 1;
      function affiche()
      {
        echo $a;
      }
      //appel de la fonction
      affiche();
    ?>
  </body>
</html>
```

# PORTÉE DES VARIABLES ... ENCORE ...

---

► que vaut **\$i** ? 1 ou 2 ?

```
<html>
  <head> exo2</head>
  <body>
    <?php
      $i=1;
      function addition ()
      {
        global $i ;
        $i=2;
        $j=3;
        $res=2+3;
        return $res;
      }
      //appel de la fonction
      $resultat = addition();
      print "le résultat de l'addition est $resultat </br> et i vaut $i"
    ?>
  </body>
</html>
```

# PORTÉE DES VARIABLES ... TOUJOURS ...

---

- Est ce que le programme va afficher quelques chose ? si oui quoi ?

```
<html>
  <head> <title>exo2-bis</title></head>
  <body>
    <?php
      $a = 1;
      function affiche()
      {
        global $a ;
        echo $a;
      }
      //appel de la fonction
      affiche();
    ?>
  </body>
</html>
```

# PORTÉE DES VARIABLES ... FIN

---

➤ Quand vous récupéré une variable avec un POST .. elle existe que dans le fichier ... pour pouvoir l'utiliser partout vous devez indiquer qu'il s'agit d'une variable de session

➤ exemple :

➤ .... dans une autre page

```
$_SESSION['toto']=$_POST['nom'];
```

➤ puis ....

```
<html>
  <head> </head>
  <body>
    <?php
      session_start();
      $age=$_POST['age'];
      print "bonjour ". $_SESSION['toto']. " vous avez ". $age ."ans </br> </br>";
    ?>
  </body>
</html>
```

## CE MATIN :

---

- Faites les exercices 1 et 2
- L'exercice 3 est à faire pour le site à rendre mais pas forcément maintenant.