

Exercice

- Ecrire un programme qui demande à l'utilisateur de saisir un nombre compris entre 10 et 20. Si le nombre n'appartient pas à cet intervalle, vous lui demandez de ressaisir la valeur jusqu'à ce que la réponse convienne.
- En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

```
$nbr = readline("saisir un nombre entier :");
while ($nbr < 10 || $nbr > 20) {
    if ($nbr < 10) {
        echo "plus grand!";
    } else {
        echo "plus petit!";
    }
    $nbr = readline("ressaisir un nombre entier :");
}
echo "ok";
```

Exercice

- Ecrire un programme qui demande à l'utilisateur de saisir un nombre compris entre 10 et 20. Si le nombre n'appartient pas à cet intervalle, vous lui demandez de ressaisir la valeur jusqu'à ce que la réponse convienne.
- En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

```
$nbr = readline("saisir un nombre entier :");
while ($nbr < 10 || $nbr > 20) {
    if ($nbr < 10) {
        $nbr = readline("plus grand !");
    } else {
        $nbr = readline("plus petit !");
    }
}
echo 'ok';
```

Exercice - Boucle

Ecrire un programme qui demande un nombre de départ et affiche ensuite les dix nombres suivants

```
<?php
$nbr = readline("saisir un nombre entier :");
$max = $nbr + 10;
while ($nbr <= $max) {
    echo $nbr . " ";
    $nbr++;
}
?>
```

Exercice - Boucle

Ecrire un programme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres.

Exemple :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

...

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Exercice – Boucle- solution

```
<?php
$nbr = readline("saisir un nombre entier :");
$max = $nbr;
for ($i = 1; $i <= 19; $i++) {
    $nbr = readline("resaisir un nombre entier :");

    if ($max < $nbr) {
        $max = $nbr;
    }
    if ($i == 19) {
        echo "le plus grand de ces nombres est " . $max;
    }
}

// echo "le plus grand de ces nombres est " . $max;

?>
```

Syntaxe de base : Les tableaux

■ Définition

- Les tableaux en PHP sont très différents des tableaux en html.
- En PHP, un tableau correspond à une variable spéciale qui va être capable de stocker plusieurs valeur en même temps.

■ Principe

- Création à l'aide de la fonction `array()`
- Uniquement des tableaux à une dimension
 - Les éléments d'un tableau peuvent pointer vers d'autres tableaux
- Les éléments d'un tableau peuvent appartenir à des types distincts
- L'index d'un tableau en PHP commence de 0
- Pas de limites supérieures pour les tableaux
- La fonction `count()` pour avoir le nombre d'éléments d'un tableau

■ Types ::

- Les tableaux numérotés ou indicés.
- Les tableaux associatifs

Syntaxe de base : Les tableaux - Tableau indicé

- Accéder aux éléments par l'intermédiaire de numéros

```
$tableau[indice] = valeur;
```

```
$jour[3] = "Mercredi";
```

```
$note[0] = 20;
```

```
$tableau = array(valeur0, valeur1,..., valeurN);
```

```
$jour = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi",  
  "Vendredi", "Samedi");
```

```
$note = array(20, 15, 12.6, 17, 10, 20, 11, 18, 19);
```

```
$variable = $tableau[indice];
```

```
$JJ = $jour[6]; // affecte "Samedi" à $JJ
```

```
echo $note[1] + $note[5];
```



Syntaxe de base : Les tableaux - Tableau associatif

- Les éléments sont référencés par des **chaînes de caractères** associatives en guise de nom: la clé d'index

```
$tableau["indice"] = valeur;
```

```
$jour["Dimanche"] = 7
```

```
$jour["Mercredi"] = "Le jour des enfants"
```

```
$tableau = array(ind0 => val0, ind1 => val1,..., indN => valN);
```

```
$jour = array("Dimanche" => 1, "Lundi" => 2, "Mardi" => 3, "Mercredi" => 4,  
"Jeudi" => 5, "Vendredi" => 6, "Samedi" => 7);
```

```
$variable = $tableau["indice"];
```

```
$JJ = $jour["Vendredi"]; //affecte 6 à $JJ
```

```
echo $jour["Lundi"]; //retourne la valeur 2
```



Syntaxe de base : Tableaux multidimensionnels

- Pas d'outils pour créer directement des tableaux multidimensionnels
- L'imbrication des tableaux est possible

```
$tab1 = array(Val0, Val1,..., ValN);  
$tab2 = array(Val0, Val1,..., ValN);  
// Création d'un tableau à deux dimensions  
$tableau = array($tab1, $tab2);
```

```
$mois = array("Janvier", "Février", "Mars", "Avril", "Mai", "Juin", "Juillet", "Août", "Septembre", "Octobre", "Novembre", "Décembre");  
$jour = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi");  
$element_date = array($mois, $jour);
```

```
$variable = $tableau[indice][indice];
```

```
$MM = $element_date[0][0]; //affecte "Janvier" à $MM  
echo $element_date[1][5] . " 7 " . $element_date[0][2] . "2022"; // retourne "Jeudi 7 Mars 2022"
```



Syntaxe de base : Lecture des éléments d'un tableau **indiqué**

```
$tab = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi");
```

- Avec une boucle for :

```
<?php
for($i=0;$i<count($tab);$i++){
    echo $tab[$i]."<br />";
}
?>
```

- Avec une boucle while :

```
<?php
$i=0;
while($i<count($tab)){
    echo $tab[$i]."<br />";
    $i++;
}
?>
```

Syntaxe de base : Lecture des éléments d'un tableau **indiqué**

```
$tab = array("Dimanche", "Lundi", "Mardi", "Mercredi", "Jeudi", "Vendredi", "Samedi");
```

- Avec La boucle foreach :

```
<?php
// Méthode 1 :

$i = 0;
foreach($tab as $item) {
    echo "La cellule n° ". $i . " : " . $item . "<br>"; $i++;
}

// Méthode 2 :

foreach($tab as $item) {
    echo $item."<br>";
}

// Méthode 3 :

foreach($tab as $key=>$item) {
    echo "clè ". $key . " valeur ". $item . "<br>";
}
?>
```