Syntaxe PHP

- **Balise PHP**: <?php code ?>

- **Déclarer une variable** :
  $var = 1;
  $var2 = ’texte’ ;
  Une variable n’a pas de type

- **Concaténation** : $var1 **.** $var2

- **Affichage de texte** : echo « texte » ;

- **Définir une constante** : define(« nom_constante »,valeur_constante) ;

- **Fonctions utiles** :
  - strlen() : longueur de la chaîne de caractère
  - str_word_count(« la maison »): renvoie 2
  - strpos(«chaine ca », « mot ») : renvoie la position de la 1ère lettre du mot cherché
  - str_replace(« World », « Dolly », « Hello World ») : renvoie Hello Dolly

- **Création d’un tableau** :
  - $nom_tableau[indice] = valeur
  - $nom_tableau2 = array(val1,val2, « val3 »)

- **Création d’un tableau associatif** :
  - $tabas["couleur"]= "rouge";$tabas["saveur"]= "sucrée";$tabas["forme"]= "rond";
  - $tabas = array ("couleur" => "rouge","saveur" => "sucrée","forme" => "rond")

- **Parcours d’un tableau associatif**
  - foreach ($tabas as $cle => $valeur){ echo " Clé = $cle. Valeur = $valeur "; }

- **Structure de contrôles possibles** : IF, ELSEIF,ELSE , WHILE, DO WHILE, FOR , SWITCH, BREAK

- **Fonctions** :

```
<?php
        function action ($arg1, $arg2,...$argn)
        { echo "Exemple de fonction. \n";
        return $retval;
        }
?>
```

- *Variable peut être local, global (accéder à des variables non déclarées dans une fonction) ou static (utiliser dans les variables dans les fonctions, pour qu’elles ne soient pas supprimés)*
-
- **Récupérer une valeur définie dans un balise HTML** : $nom = $_GET[«nom»]
  => GET permet de récupérer la valeur dans la balise ayant comme identifiant nom
  => Même chose avec POST, sauf que POST permet de récupérer des valeurs dans l’URL
  POST est mieux que GET

Syntaxe PHP/ORACLE

Oci8 permet d'accéder à des fonctions permettant d'interagir avec la base ORACLE dans le code PHP.

- **oci_connect** : Établit une connexion avec un serveur Oracle
- **oci_parse** : Prépare une requête SQL avec Oracle
- **oci_execute** : Exécute une commande SQL Oracle

Exécution d'une requête

```
$stmt = oci_parse($dbConn,"SELECT * FROM MyTable");
oci_execute($stmt);
```

$dbConn : nom de la base

oci_execute() renvoie une valeur positive si la fonction s'exécute correctement

En cas d'erreur, le message fourni par ORACLE est affiché via la fonction oci_error($stmt)

```
$stmt = oci_parse($dbConn,"SELECT * FROM MyTable");
oci_execute($stmt);
echo "<em>Erreur dans l'exécution de la requête. </em><br/>";
echo "<em>Message : </em>".oci_error($stmt);
```

- **oci_fetch** : Lit la prochaine ligne d'un résultat Oracle dans un buffer interne

Afficher le résultat d'une requête: résultat est renvoyé sous forme d'un tableau

```
while (oci_fetch($stmt))
{
echo "ID : " . oci_result($stmt, 1); . "</br>";
echo "Nom : " . oci_result($stmt, 2); . "</br>";
echo "Service ID : " . oci_result($stmt, 3); . "</br>";
echo "Salaire : " . oci_result($stmt, 4); . "</br>";
}
```

=> Penser aux points pour la concaténation

- **oci_close** : Ferme une connexion Oracle
- **oci_error** : Retourne la dernière erreur Oracle
- **oci_num_rows** : Retourne le nombre de lignes affectées durant la dernière commande Oracle

Possibilité de faire de la programmation objet sous PHP : notions de classes, héritages et encapsulations