

Document réponse 1 : Analyse des pages « index.html » et « acces.php »

Activité 1

<http://127.0.0.1/tp3web/acces.php?login=admin&motpasse=1234>

Activité 2

Donnez le nom des **informations** transmises par votre formulaire.

Le formulaire transmet les informations « login » et « motpasse ».

Donnez le nom de la balise permettant de créer un **objet** dans un formulaire.

La balise `<input />` permet de créer un objet de formulaire.

Donnez le nom de l'**attribut** permettant de différencier les objets dans un formulaire.

L'attribut « type » permet de différencier les objets de formulaire.

Donnez le nom de l'**attribut** permettant au serveur d'identifier le **script** auquel sont destinées les informations contenues dans un formulaire. Donnez le nom du script associé à votre formulaire.

L'attribut est : « action », le script est : « acces.php ».

Activité 3

Quelles balises utilise-t-on pour identifier le code PHP dans le fichier « acces.php » ?

Pour identifier du code PHP, on utilise `< ?php` et `>`



En PHP, un mot précédé de \$ est une variable.

Comment le script de traitement « acces.php » est-il capable de récupérer les données du formulaire ?

Indication: essayez de trouver des mots identiques dans chacun des fichiers.

Le script de traitement est capable de récupérer les données du formulaire car il utilise des variables avec des noms identiques à ceux des objets.

Exemple : `$_GET["login"]` reçoit la valeur transmise par l'objet de type texte nommé login.



On rappelle qu'une **structure alternative simple** s'écrit :

```

si (condition) alors action 1
                        sinon action 2 (ou rien)
fin si

```

Ecrivez l'**algorithme** correspondant au code ci-dessous sur le DR1 :

```

if (($_GET["login"] == "admin") && ($_GET["motpasse"] == "1234"))           // Remarque && ⇔ ET
    { header('Location: index1.php'); }
else
    echo "Accès refusé";

```

Rappel : un algorithme ne doit pas contenir de code. Il décrit simplement le fonctionnement attendu

si (le login et le mot de passe sont correct) **alors** afficher les grandeurs physiques
sinon afficher "Accès refusé"

fin si

Activité 4

Remplacez && par || (accessible par les touches `Alt Gr` `6`) dans le fichier « *acces.php* » et testez le fonctionnement.

Que remarquez-vous ? Quelle **opération logique** réalise l'opérateur || ? Répondez sur le **DR1**.

Indications : Testez toutes les possibilités (Ex : l'identifiant juste et le mot de passe faux etc.)

L'accès aux grandeurs physiques est possible si l'identifiant OU le mot de passe sont valides. Opérateur OU.

Activité 5

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title> Affichage de grandeurs physiques </title>
  <link rel="stylesheet" href="styles/style.css">
</head>

<body>
<?php // Ouverture du fichier contenant les données
  $FValeur = fopen("data/data.txt","r");
?>
  <div>
    <p> Grandeurs physiques </p>
    <table border='1'>
      <tr>
        <td>Température = </td>
        <td>
          <?php
            $Temp = fgets($FValeur,10); // Lecture première valeur
            echo "$Temp";
          ?>
        </td>
        <td>°C</td>
        <td rowspan='2'>
          <?php
            $Lum = fgets($FValeur,10); //Lecture deuxième valeur
            if ($Lum == 1) echo "<img src=\"img/SoleilWeb.gif\"/>";
            else echo "<img src=\"img/LuneWeb.gif\"/>";
          ?>
        </td>
      </tr>
      <tr>
        <td>Humidité =</td>
        <td> <?php
          $Hum = fgets($FValeur,10); // Lecture troisième valeur
          echo "$Hum";
        ?>
        </td>
        <td> % </td>
      </tr>
    </table>
  </div>
  <?php // Fermeture du fichier contenant les données
  fclose($FValeur);
?>
</body>
</html>
```