

## Ecriture dans une base de données



La base de données "marathon2000.asp" étant prête à être exploitée, la première étape de la programmation va constituer à **remplir cette base**, à y inscrire des informations directement depuis le navigateur et non plus en mode "feuille de données" d'Access.

Il faut prévoir 3 formulaires, un pour chaque table à remplir (on prévoit d'abord les lots, puis on inscrit les participants - l'épreuve terminée, on saisit enfin les arrivées ou les abandons de la course).

### Préparation des fichiers

Nous allons imaginer 3 fichiers distincts. Il faut les nommer, en évitant espaces, accents, caractères spéciaux. Il faut aussi leur donner une extension ".asp" et non plus ".htm" - en effet, cette modification est indispensable si l'on veut que le moindre bout de code asp soit reconnu.

Fichiers	rôle
saisie_lot.asp	saisir un nouveau lot offert par un sponsor, et mis en jeu
saisie_inscription.asp	saisir un nouveau participant
saisie_arrivee.asp	saisir les arrivées ou les abandons, ainsi que la durée de la course

L'écriture de ces fichiers comporte deux parties :

- un formulaire pour saisir les informations ;
- du code ASP pour dire comment traiter les informations renvoyées par le formulaire.

Sur la même page, on peut traiter à la fois la saisie **et** l'enregistrement dans la base (ce que nous ferons pour enregistrer les arrivées).

Cependant pour détailler cet exemple, nous utiliserons deux fichiers distincts (saisie et enregistrement) pour les lots et l'inscription. Nous avons donc deux fichiers supplémentaires à créer :

Fichiers	rôle
enregistrement_lot.asp	enregistrer le lot saisi
enregistrement_inscription.asp	enregistrer le participant, et lui affecter un numéro de dossard

---

### Les fichiers "saisie\_lot.asp" et "saisie\_inscription.asp" :

C'est le fichier qui permet d'entrer les différentes récompenses offertes pour les gagnants de l'épreuve. C'est donc l'interface entre la table "récompenses" et l'utilisateur qui saisit les informations.

Cette page se présente sous l'aspect d'un formulaire, qui va permettre de saisir tous les champs existants dans la table (sauf la référence du lot qui s'incrémentera automatiquement au moment où un nouvel enregistrement sera créé).

### **Le formulaire :**

- il est important de préciser le nom de tous les champs que nous allons y placer - c'est en effet ce nommage qui permettra par la suite de retrouver les informations saisies dans chacun d'eux.

exemple :  
`<INPUT type="text" name="nomlot" size="20">`  
 (champ texte de taille 20 caractères, la variable "nomlot" contiendra la valeur saisie)

- les données du formulaires sont envoyées à la page "enregistrement\_lot.asp". Nous utiliserons la méthode POST (mais GET fonctionne tout aussi bien).

`<form method="POST" name="SaisieLot" action="enregistrement_lot.asp">`

sous FrontPage : propriétés du formulaire

a) envoyer à (autre) - script personnalisé ASP

b) options : Action : traitement\_lot.asp ; méthode : POST


**\*remarque** : comme cette page ne présente aucun code asp, on aurait tout aussi bien pu la nommer "saisie\_lot.htm".

On fait la même chose pour le fichier "saisie\_inscription.asp" (un formulaire, qui renvoie vers la page "enregistrement\_inscription.asp", et dont on a nommé précisément tous les champs).


Les fichiers "enregistrement\_lot.asp" et "enregistrement\_inscription.asp" :

C'est le fichier qui récupère les informations que l'utilisateur a saisi dans chaque champ du formulaire "SaisieLot". Comme le formulaire a été envoyé avec la méthode POST, les informations sont récupérées avec `Request.Form("Nom_du_champ")` - [voir la FAQ pour plus de précisions](#) sur ce sujet.

**Attention** : tout le code ASP est à mettre entre "`<%`" et "`%>`" !


 **On définit des variables**, pour recevoir le contenu de ces informations (ce qui nous évite de répéter "Request.Form(...) sans arrêt) :

flot = `Request.Form("nomlot")`  
 la variable "**flot**" contient l'information saisie dans le champ "**nomlot**" du formulaire  
ex : si on a rentré dans ce champ "voyage aux Baléares", alors lot="voyage aux Baléares"

 **On ouvre la base** qui contient la table "recompenses". Ici, nous travaillons en [DSN-Less](#) ce qui permet d'éviter d'avoir à déclarer la base ODBC...

`DSN_BASE = "DBQ=" & Server.MapPath("marathon2000.mdb") & ";Driver={Microsoft Access Driver (*.mdb)};DriverId=25"`

La base s'appelle marathon2000, et est dans le même répertoire que notre fichier.

 **On crée une connexion** à la base, que l'on nomme ConnLot

```
Set ConnLot = Server.CreateObject("ADODB.Connection")
```

et **on ouvre cette connexion** :

```
ConnLot.Open DSN_BASE
```

Enfin, **on crée un recordset**, que l'on nomme RsLot :

```
Set RsLot = Server.CreateObject("ADODB.Recordset")
```

On **définit une requête SQL**, que l'on nomme RequeteSQL, qui utilise la fonction "INSERT TO" pour enregistrer de nouvelles données dans la table (cette fonction ajoute une ligne à la table, en fonction des paramètres précisés) :

- entre parenthèse et séparés par des virgules, **les noms des champs** de la table
- après "Values", entre parenthèses, apostrophes, guillemets et "&", les **variables** contenant les informations **dans le même ordre**.

```
RequeteSQL = "Insert Into recompenses (lot, sponsor, description, place_lot) Values ( ' " & flot & " ', ' " & fsponsor & " ', ' " & fdescription & " ', ' " & fplacelot & " ' ) "
```

On **ouvre le recordset** en **exécutant la Requête** :

```
RsLot.Open RequeteSQL,ConnLot
```

On **vide le recordset** ; on **ferme** et on **vide la connexion**.

```
set RsLot=nothing
```

```
ConnLot.close
```

```
Set ConnLot=nothing
```

Et voilà, l'enregistrement est terminé! Il ne reste plus qu'à laisser un message de confirmation pour signaler que tout s'est bien passé! C'est du code HTML, mais on peut y mêler de l'ASP :

```
<DIV align="center">
```

```
Merci! Le lot <B><%=flot%></B> offert par le sponsor <%=fsponsor%> a bien été enregistré!
```

```
</DIV>
```

On fait la même chose pour le fichier "enregistrement\_inscription.asp".

---

Le fichier "saisie\_arrivee.asp" :

On a dit que la saisie et l'enregistrement pouvaient se faire sur la même page. En effet, le formulaire, au lieu de renvoyer vers un fichier "enregistrement" **peut renvoyer sur le même fichier "saisie\_arrivee.asp"**! Gain de place et de temps!

Dans ce cas, le traitement du formulaire se fera **avant** la première ligne de HTML :

```
<%'-----Déclaration des variables-----%>

(... code ASP...)

<%'-----Fin déclaration des variables-----%>

<%if fdossard<>"" then%>

<%'-----Traitement du formulaire -----%>

(... code ASP...)

<%'----- Fin traitement du formulaire -----%>

<%end if%>

<HTML>

<BODY>

(... la page contenant le formulaire...)

</BODY>

</HTML>
```

#### Remarque :

Avant le traitement du formulaire, on a rajouté **une condition** :  
 "si la variable "fdossard" n'est pas vide, alors on l'enregistre dans la table"...

sous entendu : sinon, c'est qu'aucun formulaire n'a encore été validé, ce qu'il se passe lorsque l'on charge la page pour la première fois!

#### En résumé :

1. on charge la page : il ne se passe rien côté base, on ne fait que remplir le formulaire ;
2. on valide la formulaire : on recharge la page, mais cette fois, avec les valeurs que l'on a saisies ;
3. "fdossard" n'étant plus vide, on traite le formulaire : enregistrement d'une nouvelle ligne dans la table "arrivee"!



