

## Ecriture des scripts pour DiapoEngarde

Un script est un programme d'affichage écrit dans un fichier que l'on place dans le répertoire  
Ressources1\diapomodels\scripts

**Suggestion de lecture : commencer par lire les exemples présentés à la fin.**

### SYNTAXE

Un fichier script comporte :

- une parenthèse ouvrante au début
- des commandes, une commande est généralement écrite sur une ligne pour faciliter la lecture, mais ça n'est pas obligatoire
- des lignes vides
- des commentaires (parties qui sont ignorées par l'interpréteur de scripts et qui constituent des explications pour les personnes qui lisent le script) : les commentaires commencent par un « ; » et se terminent en fin de ligne
- une parenthèse ouvrante à la fin

Une commande a la syntaxe suivante :

(if <condition> <condition> ... <condition> then <action>)

Il y a 4 sortes de conditions :

(state <x>)

state signifie état général de la compétition,

les valeurs possibles de <x> sont : start poules tableaux finished

start : début de compétition | poules : on est dans les poules | tableau x: on est dans les tableaux | finished : la compétition est terminée

LA CONDITION EST VERIFIEE si au moment où on l'évalue l'état de la compétition est <x>

(substate <x>)

substate signifie sous-etat de l'état général

les valeurs possibles de <x> sont : creation ongoing

creation : on en est à la création des poules dans l'un des tours de poules ou on en est à la création du premier tableau

ongoing : les poules ou les tableaux sont en cours

LA CONDITION EST VERIFIEE si au moment où on l'évalue le sous-état de la compétition est <x>

(round <n>)

round signifie tour de poule

les valeurs possibles de <n> sont les nombres 1 2 3 etc. vus comme des numéros de tours de poules.

LA CONDITION EST VERIFIEE si au moment où on l'évalue la compétition est au tour de poules <x>

(tableau <nom> <etat-de-tableau>)

tableau signifie qu'on parle d'un tableau

<nom> est le nom du tableau dont on parle, par exemple a128, b32

<etat-de-tableau> est un état de tableau, parmi : empty preparation ongoing complete

empty : le tableau est vide | preparation : le tableau a une partie de ses matchs | ongoing : le tableau a tous ses matchs et n'est pas terminé | complete : le tableau est terminé

LA CONDITION EST VERIFIEE si au moment où on l'évalue le tableau <nom> est dans l'état <x>

UN SUITE DE CONDITIONS <condition> <condition> ... <condition> SI TOUTES LES CONDITIONS SONT VERIFIEES

Il y a 2 sortes d'actions :

(show <typeDeDocument> <nomDeModèle>)

show signifie afficher le document,

<typeDeDocument> est le type de document, parmi : list time\_piste poule\_ranking poule tableau\_ranking tableau general\_ranking

list : une liste de tireurs, d'arbitres, etc. | time\_piste : une liste de tireurs ou d'arbitres avec l'heure et la piste où ils vont | poule\_ranking : un classement dans les poules | poule : une liste de poules |

tableau\_ranking : un classement dans les tableaux | tableau : un tableau | general\_ranking : le classement général

<nomDeModèle> est le nom du fichier qui contient le modèle écrit entre guillemets, par exemple

"tableau\_a128.txt" ; si ce modèle est dans un sous-répertoire de son type, il faut mettre le chemin à parti du répertoire de son type, exemple : "special\tableau\_a128.txt"

L'ACTION DEMANDE D'AFFICHER SELON DE MODELE INDIQUE

Variante de cette sorte d'action : (show <typeDeDocument> <nomDeModèle> <n>)

<n> est un entier, elle demande d'afficher <n> fois le document.

(script <nomDeScript>)

show signifie lancer le script,

< nomDeScript > est le nom du fichier qui contient le script écrit entre guillemets, par exemple

"z\_script\_FIE-junior\_main.txt", et qui se trouve dans le répertoire des scripts.

L'ACTION DEMANDE DE LANCER LE SCRIPT INDIQUE

IL DOIT Y AVOIR EXACTEMENT UNE ACTION DANS UNE COMMANDE

## FONCTIONNEMENT

Lorsque l'on lance un script par le menu Script, l'interpréteur passe en revue les commandes et évalue leurs conditions jusqu'à ce que les conditions d'une commande soient vérifiées (il s'arrête donc sur la première commande dont les conditions sont vérifiées). Il lance alors l'action de cette commande.

Si aucune commande n'a ses conditions vérifiées, le script ne lance rien.

Lorsque le script lance un affichage, le script est réévalué toutes les 5 secondes. Si aucune commande n'est applicable, l'affichage n'est pas modifié ; si la nouvelle évaluation demande d'afficher le même document, l'affichage n'est pas modifié ; si elle demande d'afficher un autre document, l'affichage est interrompu et remplacé par l'affichage du nouveau document.

En cas d'erreur de syntaxe dans une commande, la commande est ignorée.

Lorsque l'on lance un script par le menu Script, une analyse du script est effectuée. S'il y a des erreurs, elles sont indiquées et l'utilisateur choisit de continuer ou non. Attention : les erreurs ne sont pas toutes de vraies erreurs. Par exemple, si la formule n'a pas encore été choisie, l'analyse indique que les tableaux n'existent pas. L'analyse des erreurs sert surtout lors de la mise au point des scénarios.

## EXEMPLES

### Premier exemple

```
(  
; cet exemple est fait pour les compétitions comportant ou non des poules  
; et comportant un tableau direct unique, comme c'est le cas en Coupe du Monde Junior  
  
; on montre les tireurs quand la formule n'est pas choisie  
(if (state start) then (show list "z_fencer.txt"))  
  
; on montre les tireurs quand on est au premier tour de poules et que les poules ne sont pas créées  
(if (state poules) (substate creation) (round 1) then (show list "z_fencer.txt"))  
  
; on montre les poules et le classement provisoire quand on est dans les poules  
(if (state poules) then (show poule "z_Poules1-rank6.txt"))  
  
; on montre le classement à la fin des poules quand on est prêt à créer le premier tableau  
; attention : dès que le tableau sera créé, cela sera remplacé par l'affichage du tableau  
(if (state tableaux) (substate creation) then (show poule_ranking "z_end-of-poules-rank.txt"))  
  
; on montre le tableau de 512 253 128 64 quand il est en cours  
; mettre ";" devant les lignes correspondant à un tableau qui n'existe pas  
; si l'on veut éviter des messages d'erreur (ce n'est pas obligatoire)  
  
; (if (tableau a512 ongoing) then (show tableau "z_a512-rank2.txt"))  
(if (tableau a256 ongoing) then (show tableau "z_a256-rank2.txt"))  
(if (tableau a128 ongoing) then (show tableau "z_a128-rank2.txt"))  
(if (tableau a64 ongoing) then (show tableau "z_a64-rank2.txt"))  
  
; maintenant, quand le tableau de 32 16 8 4 2 est en cours  
; on montre le tableau avec le tableau précédent  
(if (tableau a32 ongoing) then (show tableau "z_a64-rank2.txt"))  
(if (tableau a16 ongoing) then (show tableau "z_a32-rank2.txt"))  
(if (tableau a8 ongoing) then (show tableau "z_a16-rank2.txt"))  
(if (tableau a4 ongoing) then (show tableau "z_a8-rank2.txt"))  
(if (tableau a2 ongoing) then (show tableau "z_a4-rank8.txt"))  
  
; on montre le classement général à la fin
```

```
(if (state finished) then (show general_ranking "z_general-rank.txt"))  
)
```

## Deuxième exemple

```
(  
; on montre deux fois la liste des tireurs avec l'heure et la piste de leur poule  
(if (state poules) then (show time_piste "z_fencer-time-piste.txt" 2))  
; puis on lance le script general du premier exemple  
(if (state poules) then (script "z_script_FIE-junior_main.txt"))  
  
; on montre deux fois la liste des tireurs avec l'heure et la piste de leur match  
(if (state tableaux) then (show time_piste "z_fencer-time-piste.txt" 2))  
; puis on lance le script general du premier exemple  
(if (state tableaux) then (script "z_script_FIE-junior_main.txt"))  
)
```