

Cours et exercices : Variables système, Conditions et Boucles dans les scripts Batch

1. Les variables système

Définition

Les variables système (ou variables d'environnement) sont des informations stockées par Windows, accessibles dans un script Batch pour obtenir des données sur le système, l'utilisateur, des chemins ou des paramètres dynamiques.

Syntaxe

%NOM_DE_VARIABLE%

Exemples de variables système courantes

Variable	Contenu / Signification
%USERNAME%	Nom de l'utilisateur connecté
%USERPROFILE%	Dossier personnel de l'utilisateur (ex : C:\Users\Jean)
%TEMP%	Dossier temporaire de l'utilisateur
%DATE%	Date du jour
%TIME%	Heure actuelle
%COMPUTERNAME%	Nom de l'ordinateur
%OS%	Nom du système d'exploitation
%CD%	Dossier courant
%PATH%	Liste des répertoires où Windows cherche les exécutables

Exemple d'utilisation

```
@echo off
echo Utilisateur connecté : %USERNAME%
echo Dossier personnel : %USERPROFILE%
echo Date : %DATE% - Heure : %TIME%
pause
```

Remarque : Pour lister toutes les variables disponibles :

```
set
```

2. Variables personnalisées

Création

On peut créer ses propres variables dans un script Batch avec :

```
set nom_variable=valeur
```

Exemple

```
@echo off
set prenom=Jean
set age=21
echo Bonjour %prenom%, vous avez %age% ans.
pause
```

Lecture et affichage

- Pour afficher la valeur d'une variable :

```
echo %nom_variable%
```
- Pour lister toutes les variables disponibles :

```
set
```

Modification

```
set prenom=Marie
echo Nouveau prénom : %prenom%
```

Lecture interactive (saisie utilisateur)

```
@echo off
set /p nom=Entrez votre nom :
echo Bonjour %nom% !
pause
```

3. Types de variables

Dans les scripts Batch, toutes les variables sont des **chaînes de caractères**.

Il n'existe pas de types numériques ou booléens, mais il est possible d'effectuer des calculs avec set /a.

Exemple de calcul

```
@echo off
set /a a=5
set /a b=3
set /a somme=a+b
echo Le résultat est : %somme%
pause
```

4. Portée des variables

- Les variables créées avec set sont temporaires : elles existent seulement pendant l'exécution du script.
- Pour créer une variable permanente, on utilise setx :

```
setx COURS "Scripting"
```

Cette variable sera disponible dans les futures sessions CMD.

5. Affichage, test et concaténation

Concaténation

```
@echo off
set prenom=Jean
set nom=Dupont
set identite=%prenom% %nom%
echo Bonjour %identite%
pause
```

Tester si une variable est vide

```
if "%prenom%"==" " echo La variable est vide
```

6. Exemple global

```
@echo off
title Exemple de script complet

rem Lecture de variables système
echo Utilisateur : %USERNAME%
echo Machine : %COMPUTERNAME%
echo Dossier actuel : %CD%

rem Lecture de variables saisies par l'utilisateur
set /p ville=Entrez votre ville :
set /p age=Entrez votre âge :

rem Calcul simple
set /a age_prochain=age+1

rem Affichage final
echo.
echo Bonjour %USERNAME%, vous habitez à %ville%.
echo L'année prochaine, vous aurez %age_prochain% ans.
echo Date d'exécution : %DATE% - %TIME%
pause
```

7. La condition IF

Syntaxe de base

```
IF condition (commande_si_vrai)
```

Avec alternative :

```
IF condition (commande_si_vrai) ELSE (commande_si_faux)
```

Comparaison de chaînes

```
IF "%USERNAME%"=="Admin" echo Bonjour administrateur !
```

Comparaison numérique

```
set age=18
```

```
IF %age% GEQ 18 echo Accès autorisé.
```

Opérateurs numériques

Opérateur	Signification
EQU	Égal à
NEQ	Différent de
LSS	Inférieur à
LEQ	Inférieur ou égal à
GTR	Supérieur à
GEQ	Supérieur ou égal à

Test d'existence de fichier ou dossier

IF EXIST "C:\Backup" echo Le dossier existe.

IF NOT EXIST "C:\Backup" echo Dossier non trouvé, création...

Exemple complet

```
@echo off
set /p user=Entrez votre nom :
IF "%user%"=="root" (
    echo Bonjour administrateur !
) ELSE (
    echo Bonjour %user%, accès standard.
)
pause
```

8. Les conditions imbriquées

Les conditions imbriquées permettent d'exécuter des tests à l'intérieur d'autres tests — un peu comme des poupées russes logiques.

Exemple simple :

```
@echo off
set /p user=Entrez votre nom :
set /p pass=Entrez votre mot de passe :

IF "%user%"=="admin" (
    IF "%pass%"=="1234" (
        echo Accès complet autorisé.
    ) ELSE (
        echo Mot de passe incorrect.
    )
) ELSE (
    echo Bonjour %user%, vous n'êtes pas administrateur.
)
pause
```

Principe :

- Le premier IF vérifie si l'utilisateur est "admin".
- Le second IF, imbriqué, vérifie le mot de passe uniquement si la première condition est vraie.
- Cela permet de gérer plusieurs niveaux de vérifications ou d'actions.

Astuce :

- Pour garder ton code lisible, n'hésite pas à indenter les blocs avec des tabulations ou des espaces.

9. Le choix multiple avec CHOICE et ERRORLEVEL

En Batch, il n'existe pas de structure switch native, mais la commande CHOICE combinée à des IF ou GOTO fait parfaitement le travail.

Exemple de menu interactif :

```
@echo off
:menu
cls
echo =====
echo    MENU PRINCIPAL
echo =====
echo [1] Sauvegarder les données
```

```

echo [2] Restaurer les données
echo [3] Quitter
echo.
CHOICE /C 123 /M "Faites un choix : "

```

```

IF ERRORLEVEL 3 GOTO quit
IF ERRORLEVEL 2 GOTO restore
IF ERRORLEVEL 1 GOTO backup

```

```

:backup
    echo Sauvegarde en cours...
    pause
    GOTO menu

```

```

:restore
    echo Restauration en cours...
    pause
    GOTO menu

```

```

:quit
    echo Au revoir !
    pause

```

Explications :

- **CHOICE /C 123** définit les touches valides (ici 1, 2 ou 3).
- **ERRORLEVEL** renvoie le numéro du choix (3 pour la dernière option, etc.).
- Chaque **GOTO** renvoie vers une étiquette (:backup, :restore, :quit)

10. La boucle FOR

Syntaxe générale

FOR %%variable IN (liste) DO commande

- Dans un script .bat, utiliser %%variable
- Dans l'invite de commande, utiliser %variable

Exemple simple

Affiche successivement les nombres de 1 à 5 en répétant la commande echo pour chaque valeur de la liste.

```

@echo off
for %%i in (1 2 3 4 5) do echo Nombre : %%i
pause

```

```

setlocal enabledelayedexpansion
for /L %%i in (1,1,1000) do (
    echo Nombre : %%i

```

```

    for /L %%j in (1,1,10) do (
        set /a result=%%i * %%j
        echo %%i x %%j = !result!
    )

```

```

    echo -----
)
pause

```

Exemple sur des fichiers

Parcourt tous les fichiers .txt du dossier et affiche leur nom un par un.

```

@echo off
for %%f in (*.txt) do echo Fichier trouvé : %%f
pause

```

Exemple sur des dossiers

Liste de tous les sous-dossiers présents dans C:\Users, donc un par un les dossiers utilisateurs du système.

/d indique de parcourir les dossiers seulement

```

@echo off

```

```
for /d %%x in (C:\Users\*) do echo Dossier utilisateur : %%x  
pause
```

Exemple pratique : suppression de tous les sous-dossiers temporaires

Vide le dossier temporaire de l'utilisateur en supprimant silencieusement tous ses sous-dossiers. Penser à bien fermer les applications avant de supprimer le dossier TEMP

```
@echo off  
set TEMP_DIR=%TEMP%  
for /d %%x in ("%TEMP_DIR%\*") do rd /s /q "%%x"  
echo Tous les sous-dossiers ont été supprimés.  
pause
```

Boucle numérique

Exécute cinq fois la commande echo, en affichant "Cycle 1" à "Cycle 5" successivement.

/L indique au for d'itérer sur une plage numérique — la syntaxe est (début,pas,fin), donc for /L %%i in (1,1,5) produit 1,2,3,4,5 ; sans /L, for parcourt une liste littérale d'éléments.

```
@echo off  
for /L %%i in (1,1,5) do echo Cycle %%i  
pause
```

Boucle sur un fichier texte

Parcourt chaque ligne du fichier liste.txt

```
@echo off  
for /F %%l in (liste.txt) do echo Ligne lue : %%l  
pause
```

TD

Exercices 1 : Variables système

1. Affichez à l'écran le **nom de l'utilisateur** et le **nom de l'ordinateur**.
2. Affichez le **répertoire courant** et le **répertoire temporaire**.
3. Vérifiez si la variable %OS% contient bien Windows_NT et affichez un message correspondant.
4. Créez un script qui affiche la **date** et l'**heure** du jour avec un message personnalisé.
5. Lister toutes les variables disponibles avec la commande set.

Exercices 2 : Variables personnalisées

1. Créez une variable prenom et une variable nom, puis affichez le texte : Bonjour <prenom> <nom>.
2. Modifiez la valeur de la variable prenom et réaffichez le message.
3. Demandez à l'utilisateur de saisir son **ville** et son **âge**, puis affichez : Vous habitez à <ville> et vous avez <âge> ans.
4. Créez une variable numérique a et b, calculez leur somme et affichez le résultat.
5. Listez toutes vos variables personnalisées avec la commande set et observez leur valeur.

Exercices 3 : Conditions IF

1. Écrivez un script qui demande un **nom d'utilisateur** et affiche :
 - "Bonjour administrateur" si le nom est root
 - "Bonjour <nom>" sinon.
2. Demandez à l'utilisateur de saisir un **nombre** et affichez "Nombre positif" si le nombre est supérieur ou égal à 0, sinon "Nombre négatif".
3. Vérifiez si le fichier C:\Windows\System32\notepad.exe existe et affichez un message.
4. Créez un script qui teste si la variable ville est vide et affiche "Variable vide" ou "Ville saisie : <ville>".
5. Testez différents opérateurs numériques (EQU, NEQ, GTR, LEQ) avec des variables et affichez le résultat.

Exercices 4 : Boucles FOR

1. Affichez les nombres de 1 à 10 avec une boucle FOR.
2. Affichez tous les fichiers .txt dans le dossier courant.
3. Affichez tous les dossiers présents dans C:\Users.
4. Créez un script qui supprime tous les sous-dossiers du dossier temporaire %TEMP%.
5. Parcourez un fichier texte liste.txt et affichez chaque ligne à l'écran.
6. Créez une boucle qui affiche : Cycle 1, Cycle 2, ..., Cycle 5.
7. Parcourez une liste de mots (Chat Chien Oiseau Poisson) et affichez Animal : <mot> pour chaque mot.

