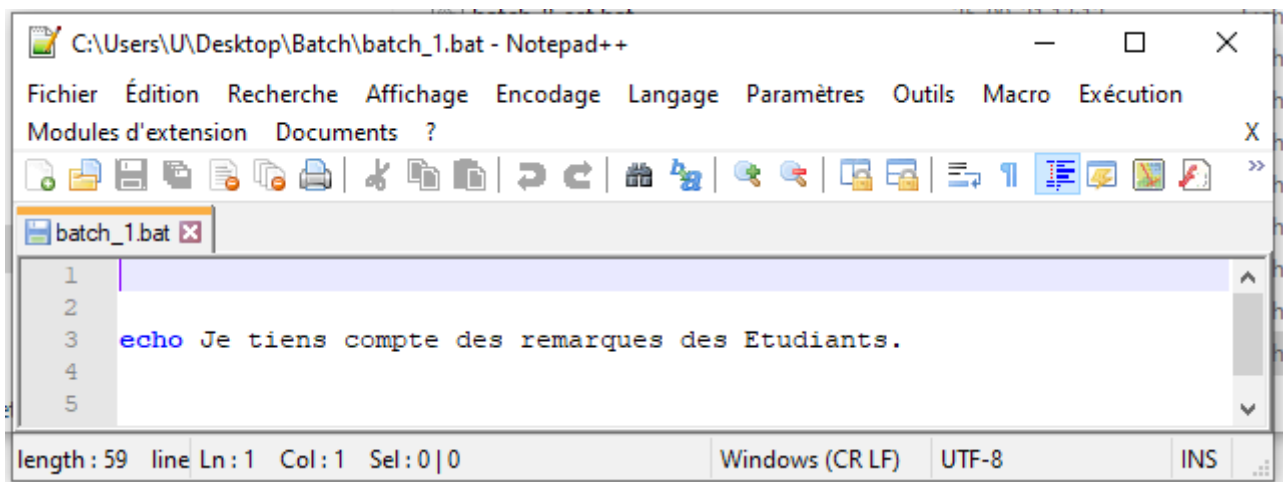


INTRODUCTION AUX SCRIPTS BATCH SIMPLES

L'historique des scripts batch est expliquée en classe.

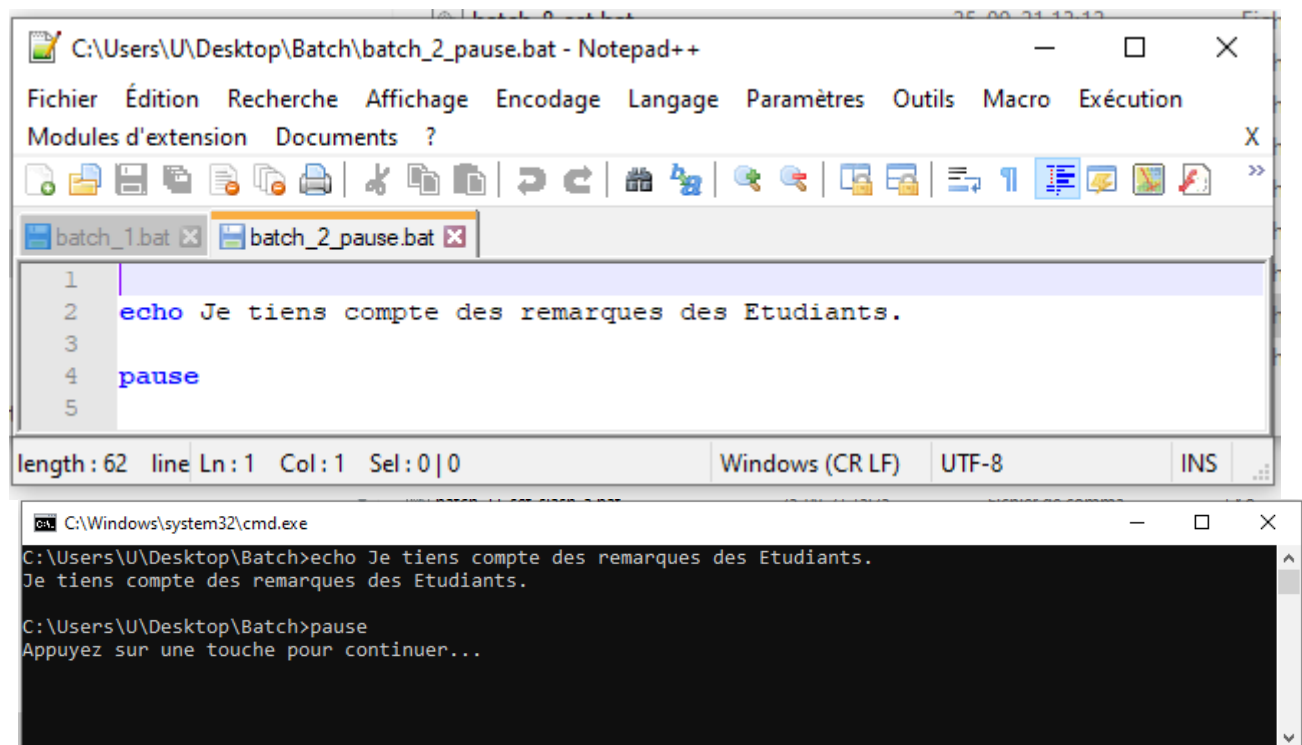
La commande la plus utilisée est echo

Elle permet de «printer» un message



```
C:\Users\U\Desktop\Batch\batch_1.bat - Notepad++
Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramètres  Outils  Macro  Exécution
Modules d'extension  Documents  ?
batch_1.bat
1
2
3 echo Je tiens compte des remarques des Etudiants.
4
5
length: 59 line Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS
```

Le message s'affiche tellement vite dans une invite dos que nous n'avons pas le temps de le voir. Il est donc préconiser de mettre une pause dans le programme. La pause est levée par Enter.

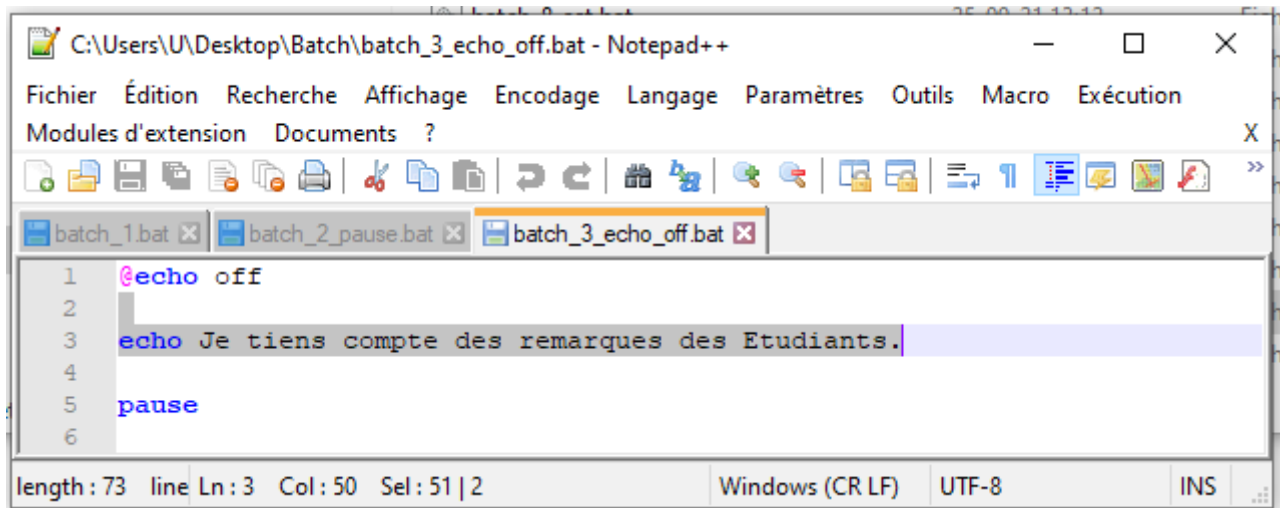


```
C:\Users\U\Desktop\Batch\batch_2_pause.bat - Notepad++
Fichier  Édition  Recherche  Affichage  Encodage  Langage  Paramètres  Outils  Macro  Exécution
Modules d'extension  Documents  ?
batch_1.bat batch_2_pause.bat
1
2 echo Je tiens compte des remarques des Etudiants.
3
4 pause
5
length: 62 line Ln: 1 Col: 1 Sel: 0 | 0 Windows (CR LF) UTF-8 INS

C:\Windows\system32\cmd.exe
C:\Users\U\Desktop\Batch>echo Je tiens compte des remarques des Etudiants.
Je tiens compte des remarques des Etudiants.

C:\Users\U\Desktop\Batch>pause
Appuyez sur une touche pour continuer...
```

Pour décharger l'écran de l'invite dos et ne voir que le message envoyé, nous utilisons la commande @echo off.



```
1 @echo off
2
3 echo Je tiens compte des remarques des Etudiants.
4
5 pause
6
```

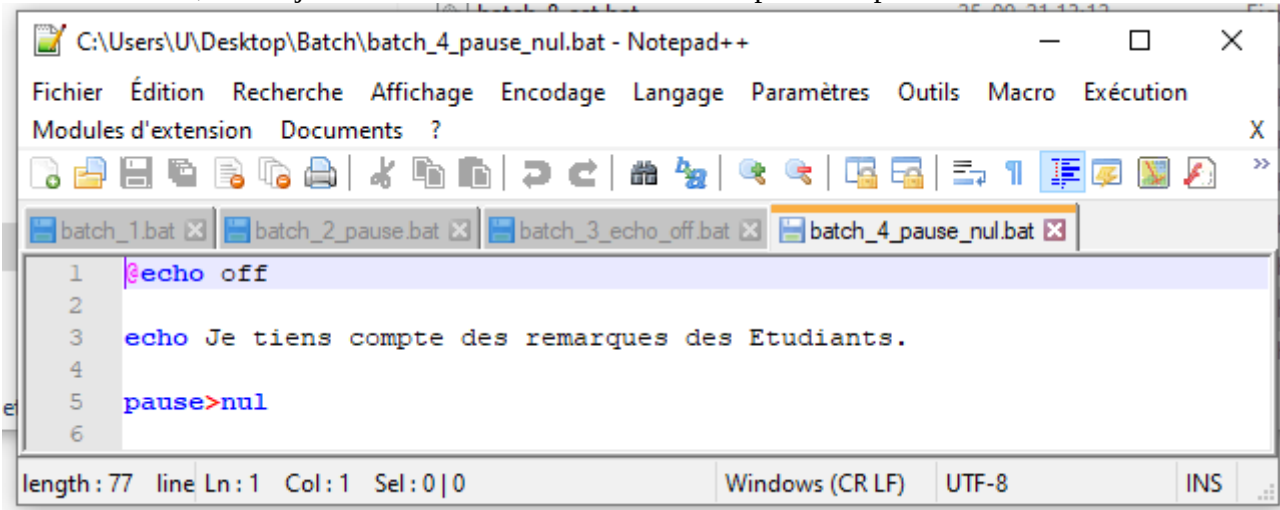


```
C:\Windows\system32\cmd.exe
Je tiens compte des remarques des Etudiants.
Appuyez sur une touche pour continuer...
```

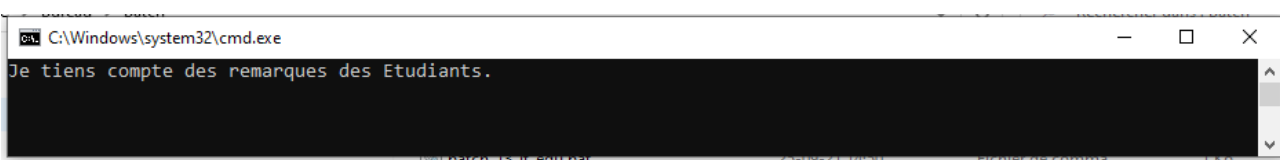
Ce n'est pas suffisant pour voir le message seul.

Nous souhaiterions enlever le message «Appuyer sur une touche pour continuer».

Pour se réaliser, nous ajoutons transformons la commande pause en pause>nul.

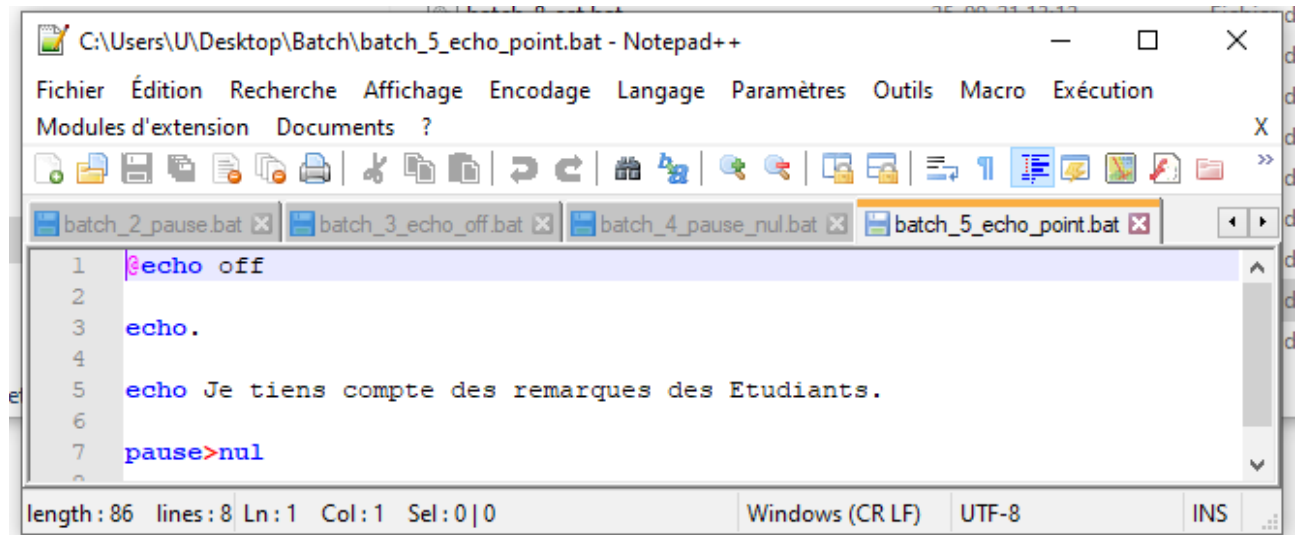


```
1 @echo off
2
3 echo Je tiens compte des remarques des Etudiants.
4
5 pause>nul
6
```



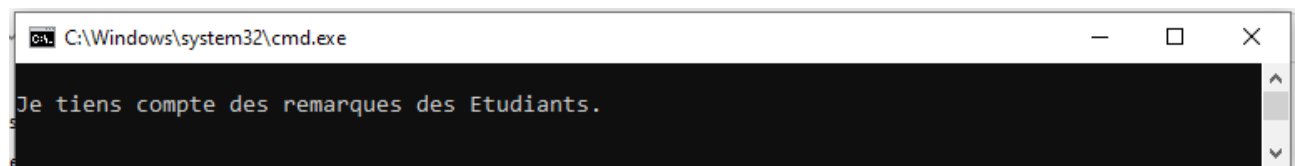
```
C:\Windows\system32\cmd.exe
Je tiens compte des remarques des Etudiants.
```

Si nous mettons des lignes d'espaces de ligne dans le script, l'invite dos n'en tiens pas compte. Pour ajouter des lignes, nous ajoutons echo.



```
1 @echo off
2
3 echo.
4
5 echo Je tiens compte des remarques des Etudiants.
6
7 pause>nul
```

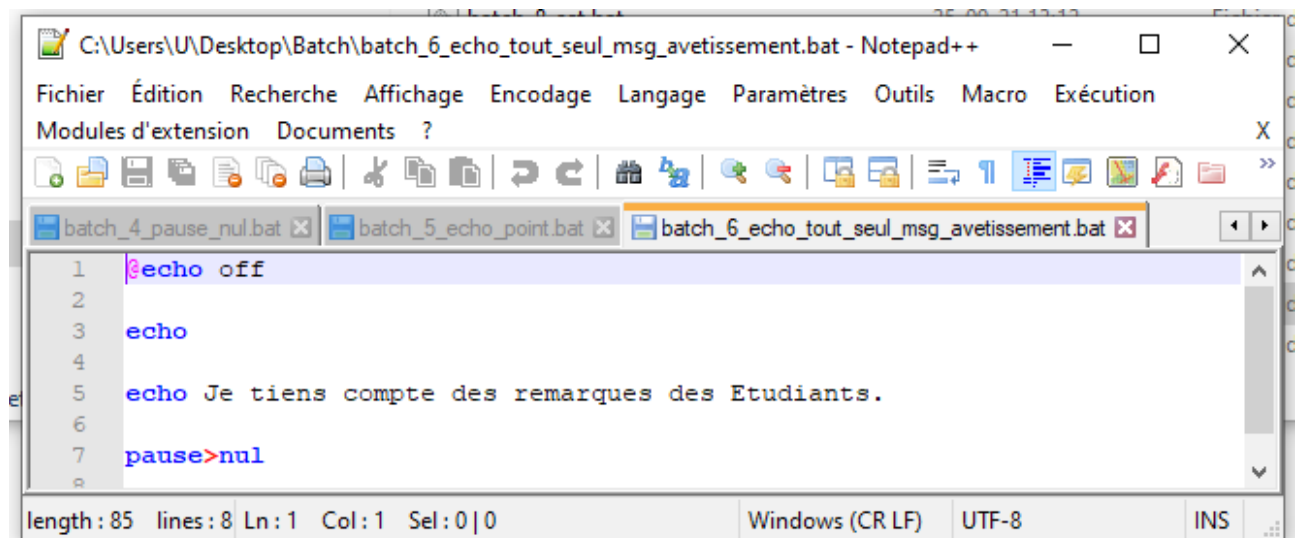
length: 86 lines: 8 Ln: 1 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS



```
C:\Windows\system32\cmd.exe

Je tiens compte des remarques des Etudiants.
```

Il est préférable de ne pas utiliser echo sans suite. L'invite dos vous affichera «Commande ECHO désactivée».



```
1 @echo off
2
3 echo
4
5 echo Je tiens compte des remarques des Etudiants.
6
7 pause>nul
```

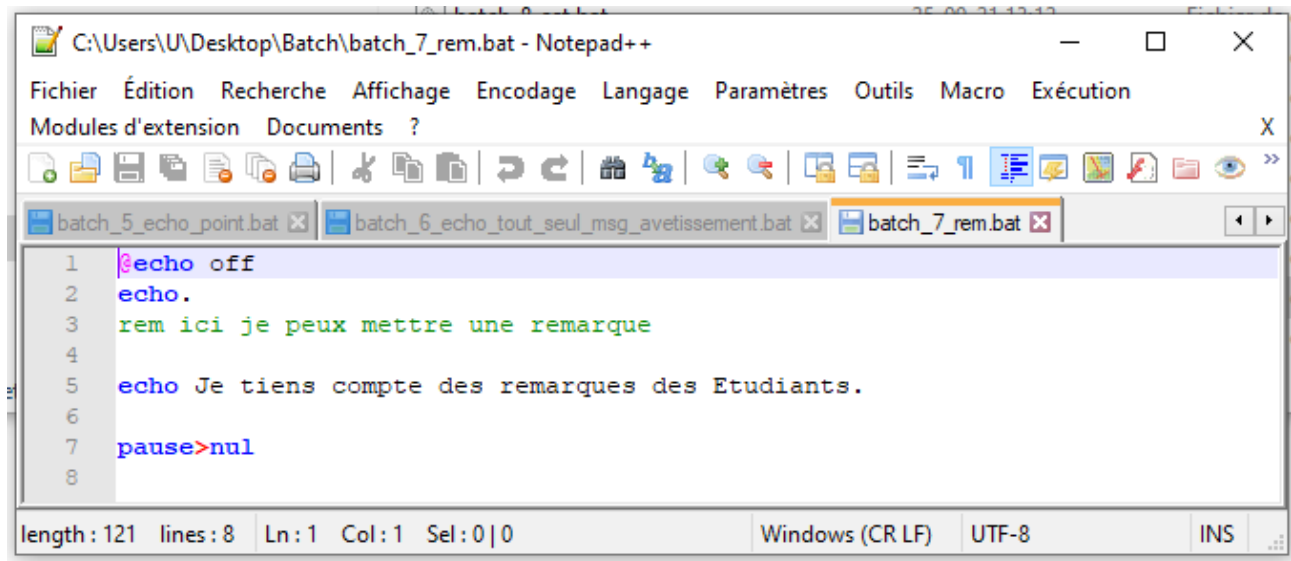
length: 85 lines: 8 Ln: 1 Col: 1 Sel: 0|0 Windows (CR LF) UTF-8 INS



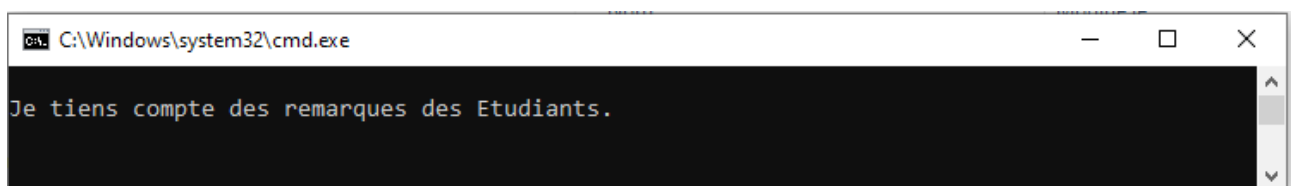
```
C:\Windows\system32\cmd.exe

Commande ECHO désactivée.
Je tiens compte des remarques des Etudiants.
```

Nous pouvons ajouter des commentaires dans le script pour le rendre plus lisible. Nous ajoutons le commentaire après la commande rem.



```
1 @echo off
2 echo.
3 rem ici je peux mettre une remarque
4 echo Je tiens compte des remarques des Etudiants.
5 pause>nul
```



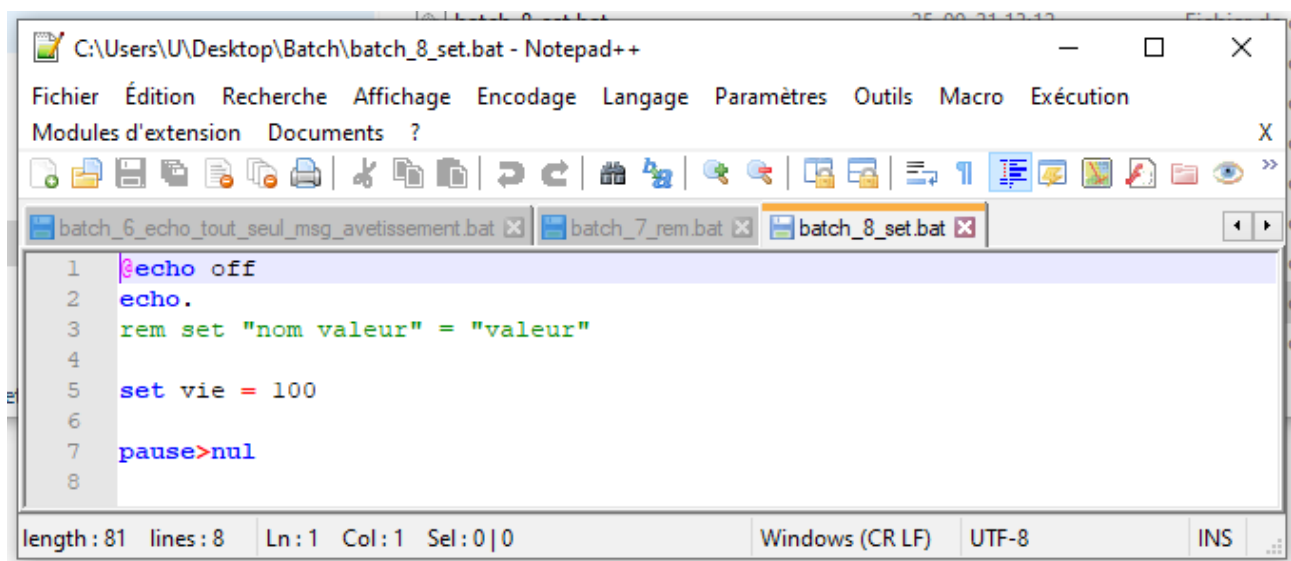
```
C:\Windows\system32\cmd.exe
Je tiens compte des remarques des Etudiants.
```

Nous pouvons placer une valeur dans une variable.

La variable est un nom associé à une place mémoire (Nous demandons au processeur de trouver une place mémoire que nous choisissons d'appeler la variable).

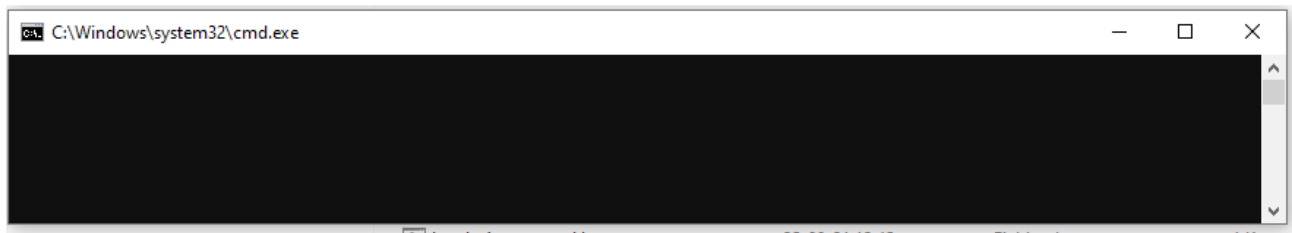
La valeur est l'élément que nous plaçons dans la zone mémoire.

Dans l'exemple ci-dessous, vie est la variable et 100 la valeur que nous plaçons dans la variable vie.

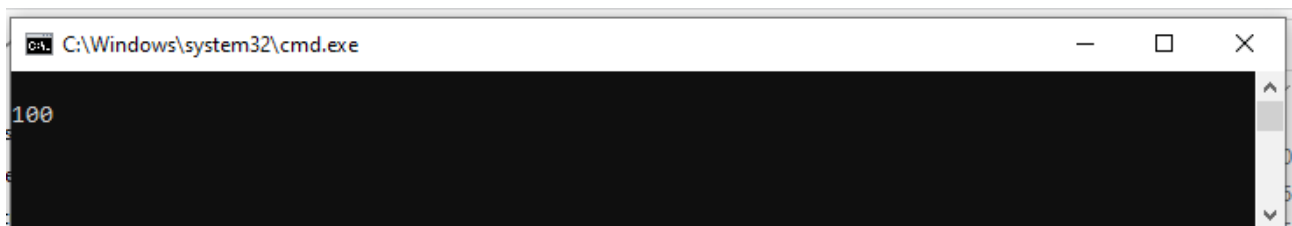
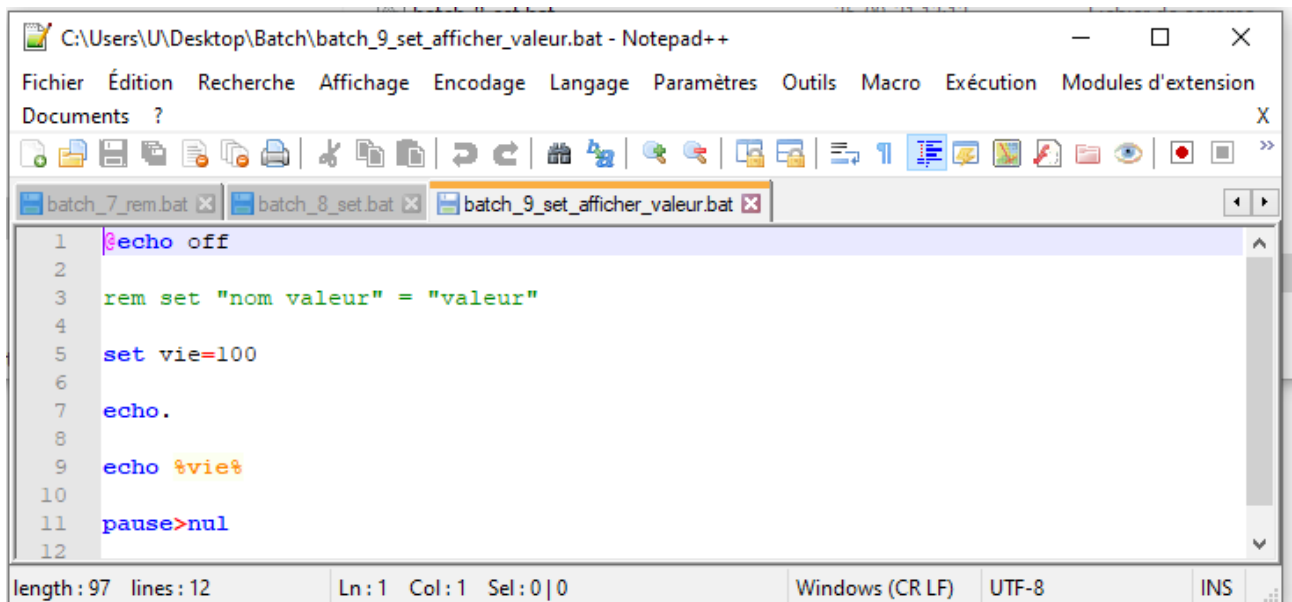


```
1 @echo off
2 echo.
3 rem set "nom valeur" = "valeur"
4 set vie = 100
5 pause>nul
```

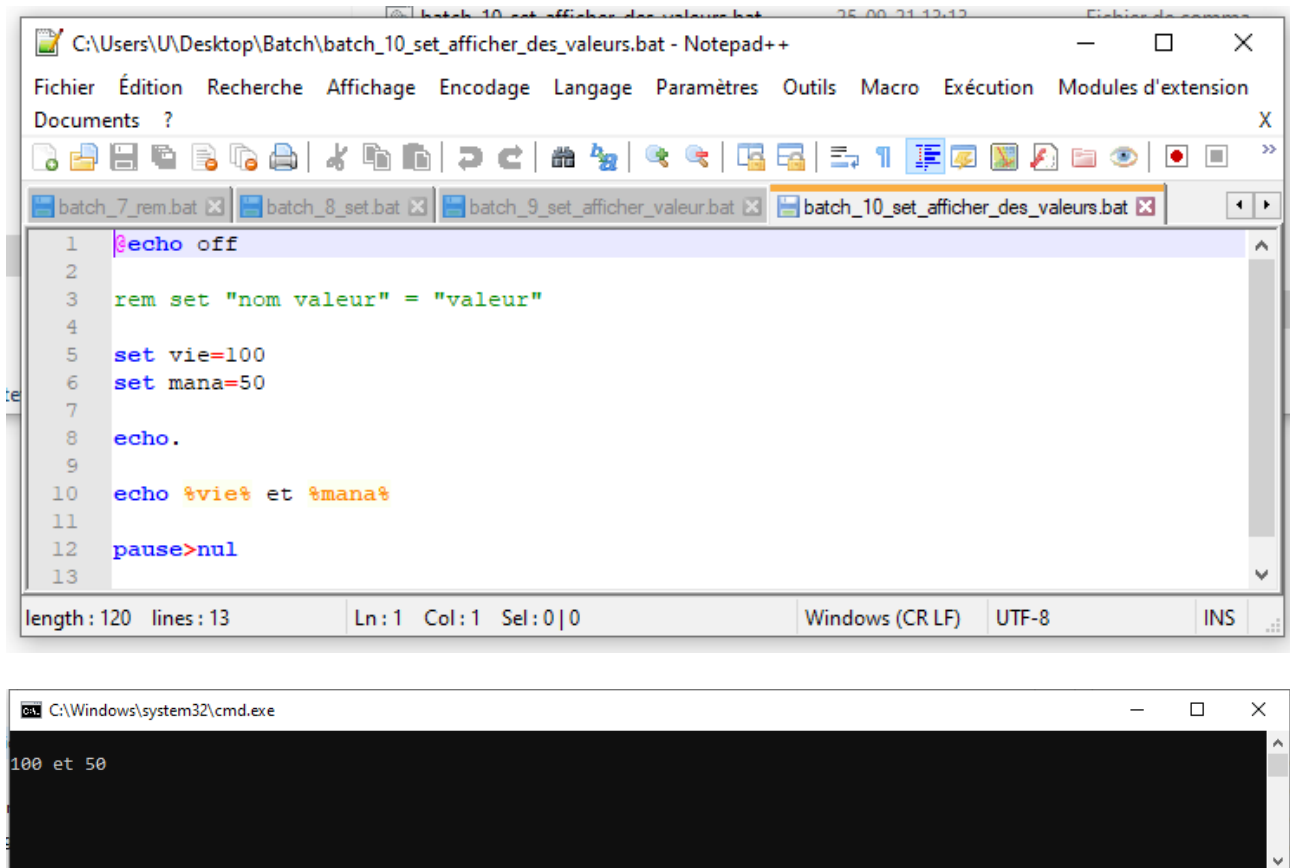
Il ne se passe rien. C'est normal. Nous n'avons placé que la valeur 10 dans la variable vie.



Nous allons simplement afficher la valeur (10 ici) de la variable vie (ici) avec la commande echo. Dans la commande echo nous n'oublions pas de mettre des % de chaque côté de la variable. Sinon echo affichera simplement le nom de la variable (en fait ce que vous avez écrit comme dans un echo classique) et non la valeur.



Nous pouvons bien évidemment travailler avec plusieurs variables comme dans l'exemple ci-dessous.



The image shows two windows. The top window is Notepad++ editing a batch file named `batch_10_set_afficher_des_valeurs.bat`. The script contains the following lines:

```
1  @echo off
2
3  rem set "nom valeur" = "valeur"
4
5  set vie=100
6  set mana=50
7
8  echo.
9
10 echo %vie% et %mana%
11
12 pause>nul
13
```

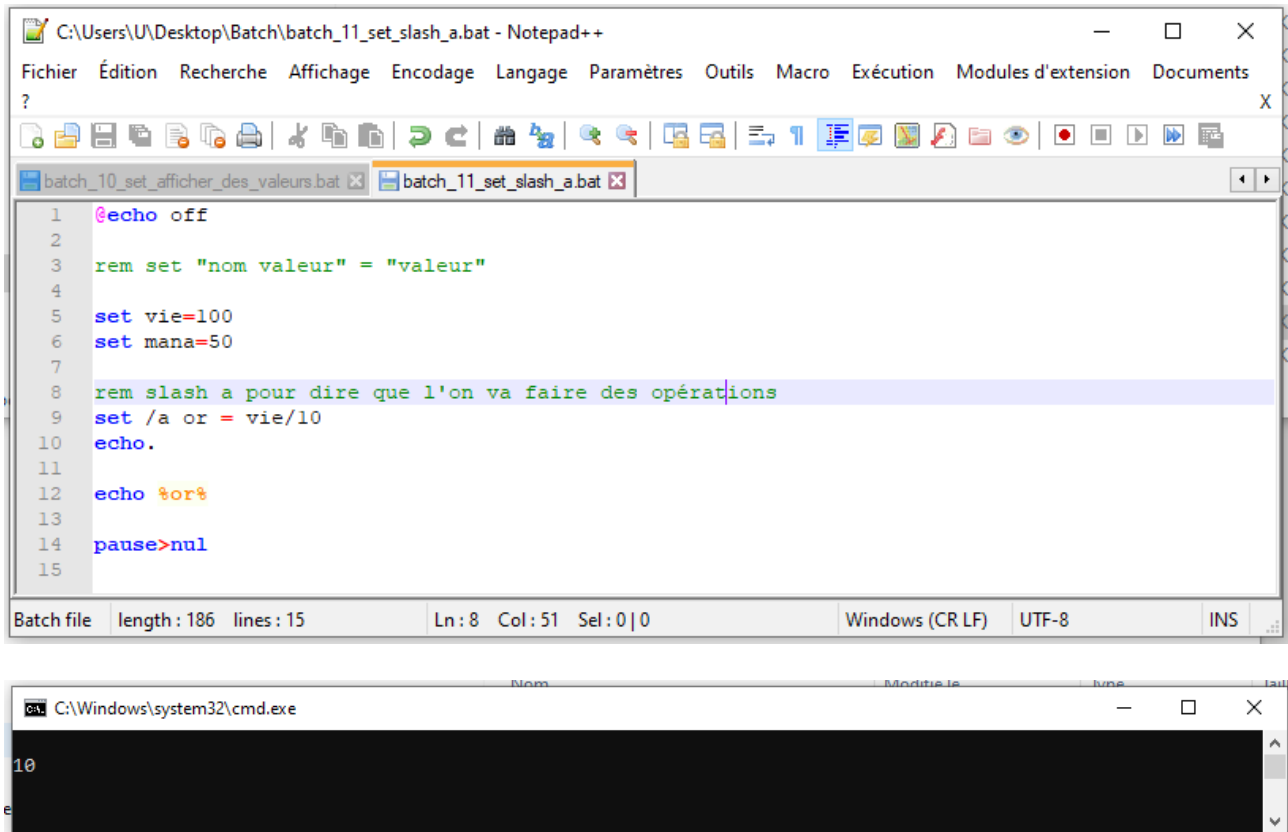
The bottom window is a Command Prompt running `C:\Windows\system32\cmd.exe`. It displays the output of the batch script:

```
100 et 50
```

Nous pouvons réaliser des opérations sur les variables comme en mathématique:

- + pour l'addition
- pour la soustraction
- * pour la multiplication
- / pour la division

La commande qui permet de préciser que nous allons réaliser des opérations est `set /a`.



The image shows two windows. The top window is Notepad++ editing a batch file named `batch_11_set_slash_a.bat`. The script contains the following code:

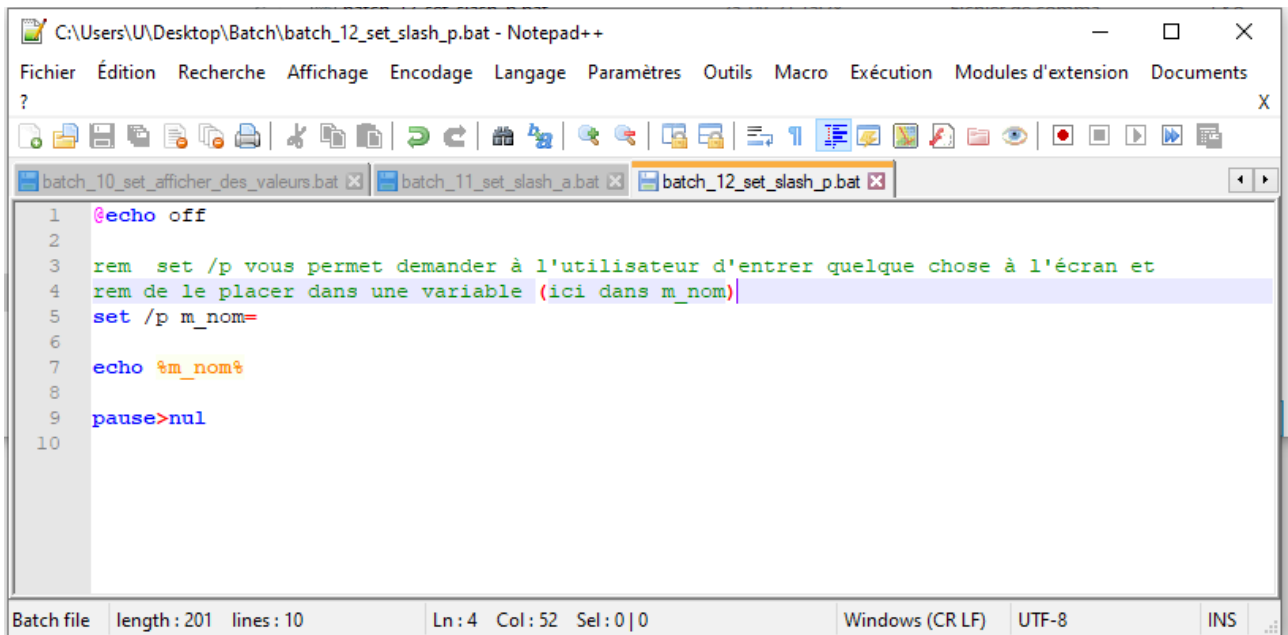
```
1 @echo off
2
3 rem set "nom valeur" = "valeur"
4
5 set vie=100
6 set mana=50
7
8 rem slash a pour dire que l'on va faire des opérations
9 set /a or = vie/10
10 echo.
11
12 echo %or%
13
14 pause>nul
15
```

The bottom window is a Windows Command Prompt (`C:\Windows\system32\cmd.exe`) showing the output of the batch script, which is the number `10`.

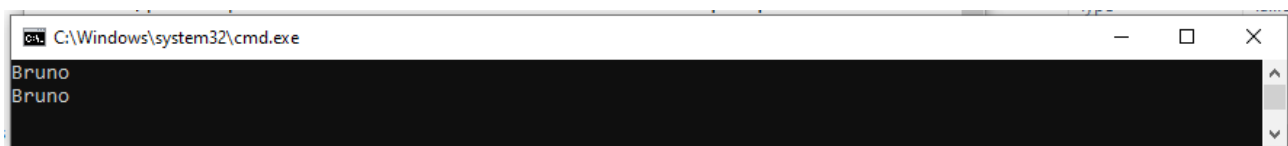
Nous allons rendre le script interactif. Cela veut dire que nous allons demander à l'utilisateur d'entrer des informations dans l'invite dos et mémoriser ces informations dans des variables. La commande `set /p` permet de récupérer la suite de caractères introduite par l'utilisateur dans l'invite dos.

Dans le programme ci-dessous, la suite de caractères est placée dans la variable `m_nom` grâce à la commande `set /p`.

nous utilisons ensuite la commande `echo` et les `%` pour afficher cette valeur à l'écran en précisant le nom de la variable `m_nom`



```
1 @echo off
2
3 rem set /p vous permet demander à l'utilisateur d'entrer quelque chose à l'écran et
4 rem de le placer dans une variable (ici dans m_nom)
5 set /p m_nom=
6
7 echo %m_nom%
8
9 pause>nul
10
```



```
C:\Windows\system32\cmd.exe
Bruno
Bruno
```


Nous introduisons ici la condition «si condition alors action».

Le premier type de condition que nous voyons est le suivant:

Si la valeur d'une variable **est la même que** la valeur d'une autre variable alors action.

Si se traduit par **if**

est la même que se traduit par **equ**

If %variable% equ %autre_varibale% alors action.

Nous pouvons choisir une multitude d'actions, de commandes donc.

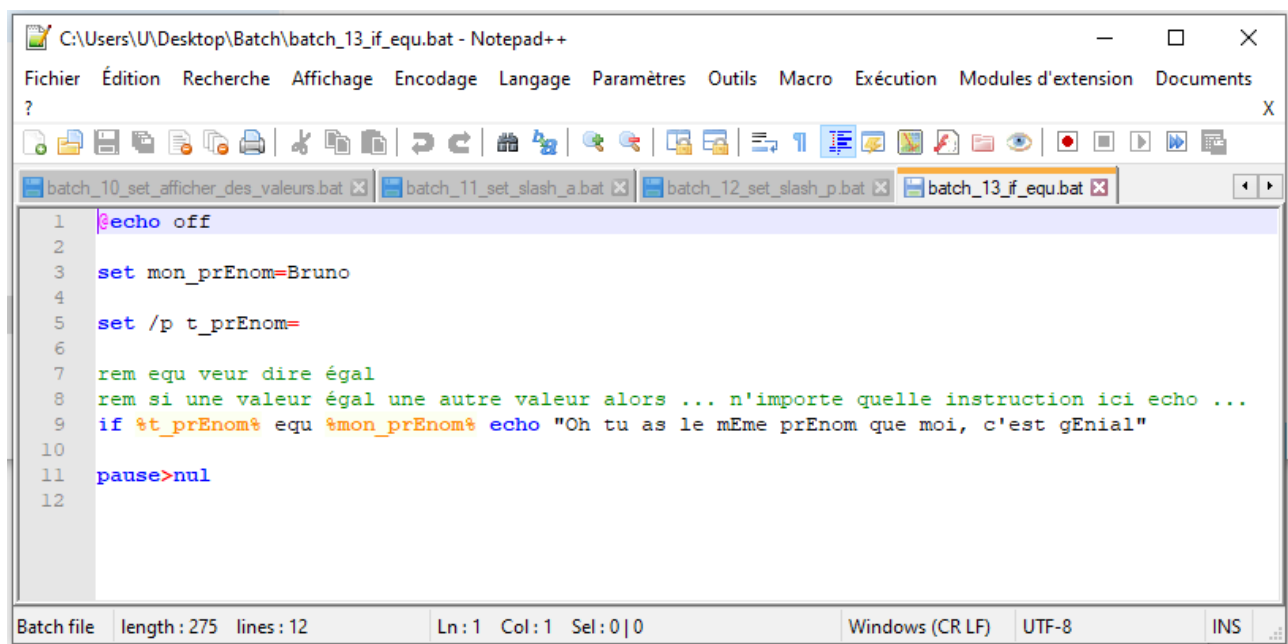
Dans l'exemple ci-dessous, nous utilisons la commande echo

t_prEnom est la variable.

mon_prEnom est la deuxième variable (dont j'ai fixé la valeur avec set mon_prEnom=Bruno.

Donc au final, nous avons

if %t_prEnom% equ %mon_prEnom% echo « Oh tu as le même prénom que moi, c'est génial»



```
1 @echo off
2
3 set mon_prEnom=Bruno
4
5 set /p t_prEnom=
6
7 rem equ veut dire égal
8 rem si une valeur égal une autre valeur alors ... n'importe quelle instruction ici echo ...
9 if %t_prEnom% equ %mon_prEnom% echo "Oh tu as le mEme prEnom que moi, c'est gEnial"
10
11 pause>nul
12
```



```
C:\Windows\system32\cmd.exe
Bruno
"Oh tu as le mEme prEnom que moi, c'est gEnial"
```

Le deuxième type de condition que nous voyons est le suivant:

Si la valeur d'une variable **n'est pas la même que** la valeur d'une autre variable alors action.

Si se traduit par **if**

n'est pas la même que se traduit par **neq**

If %variable% neq %autre_varibale% alors action.

Nous pouvons choisir une multitude d'actions, de commandes donc.

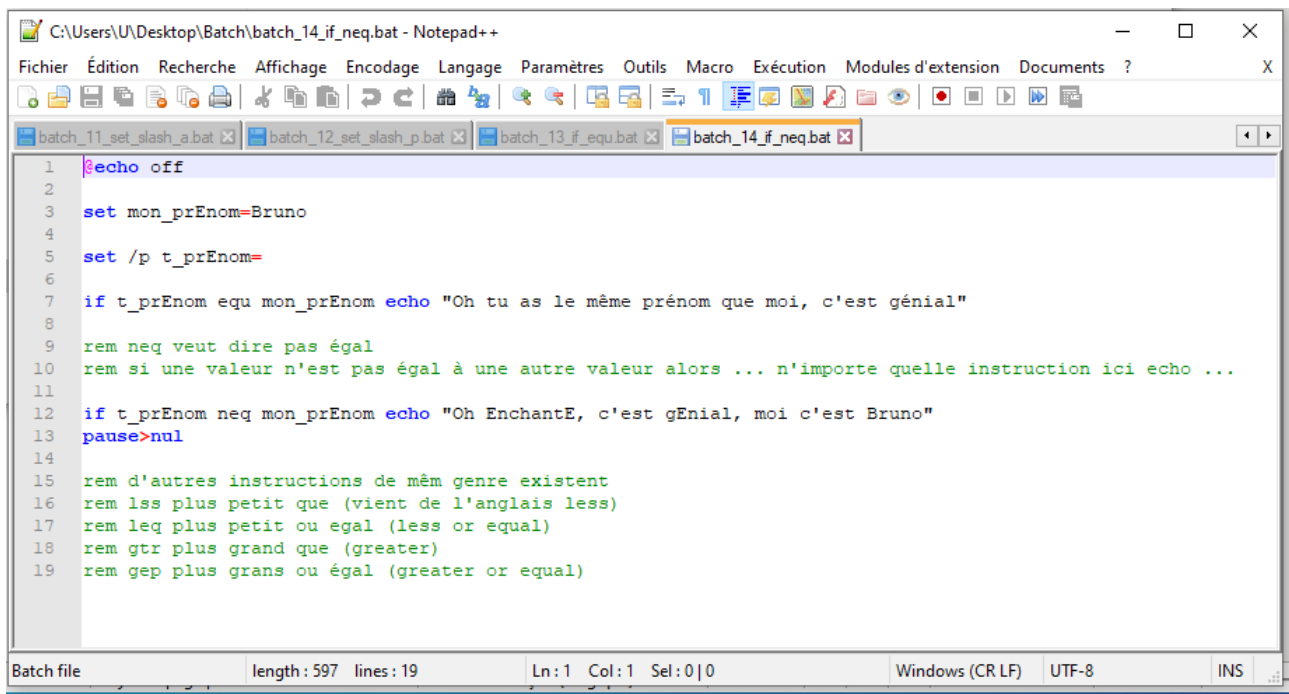
Dans l'exemple ci-dessous, nous utilisons la commande echo

t_prEnom est la variable.

mon_prEnom est la deuxième variable (dont j'ai fixé la valeur avec set mon_prEnom=Bruno).

Donc au final, nous avons

if %t_prEnom% neq %nom_prEnom% echo « Oh EnchantE, c'est génial, moi c'est Bruno»



```
1 echo off
2
3 set mon_prEnom=Bruno
4
5 set /p t_prEnom=
6
7 if t_prEnom equ mon_prEnom echo "Oh tu as le même prénom que moi, c'est génial"
8
9 rem neq veut dire pas égal
10 rem si une valeur n'est pas égal à une autre valeur alors ... n'importe quelle instruction ici echo ...
11
12 if t_prEnom neq mon_prEnom echo "Oh EnchantE, c'est gEnial, moi c'est Bruno"
13 pause>nul
14
15 rem d'autres instructions de mêm genre existent
16 rem lss plus petit que (vient de l'anglais less)
17 rem leq plus petit ou égal (less or equal)
18 rem gtr plus grand que (greater)
19 rem gep plus grans ou égal (greater or equal)
```



```
C:\Windows\system32\cmd.exe
Jacques
Oh EnchantE, c'est gEnial, moi c'est Bruno
```

D'autres instructions de même genre existent :

lss plus petit que (vient de l'anglais less)

leq plus petit ou égal (less or equal)

gtr plus grand que (greater)

gep plus grans ou égal (greater or equal)