

QCM sur la Programmation en Shell (30 questions)

Partie 1 : Bases des scripts shell (6 questions)

Q1) Quel est le symbole permettant d'indiquer qu'un fichier est un script shell Bash ?

- a) //
- b) `#!/bin/bash`
- c) `# Script`
- d) `/* */`

Correction : La bonne réponse est **b)** `#!/bin/bash`. Cela indique au système que le fichier doit être interprété comme un script Bash.

Q2) Comment rendre un script exécutable ?

- a) `chmod +r fichier.sh`
- b) `chmod +x fichier.sh`**
- c) `chmod +w fichier.sh`
- d) `chmod -x fichier.sh`

Correction : La bonne réponse est **b)**. La commande `chmod +x` ajoute les permissions d'exécution à un fichier.

Q3) Quelle commande est utilisée pour exécuter un script shell ?

- a) `run script.sh`
- b) `./script.sh`**
- c) `bash script.sh`
- d) Les réponses b et c

Correction : La bonne réponse est **d)**. Les scripts Bash peuvent être exécutés soit directement (`./script.sh`) soit en appelant explicitement l'interpréteur Bash (`bash script.sh`).

Q4) Que fait la commande `echo` dans un script shell ?

- a) Affiche une ligne de texte**
- b) Exécute un programme externe
- c) Supprime une variable
- d) Termine le script

Correction : La bonne réponse est **a)**. La commande `echo` sert à afficher une ligne de texte dans le terminal.

Q5) Dans quel cas un commentaire est-il valide dans un script Bash ?

- a) Lorsqu'il commence par le symbole `#`**
- b) Lorsqu'il est écrit entre `/* */`

- c) Lorsqu'il commence par //
- d) Toutes les propositions

Correction : La bonne réponse est **a)**. En Bash, les commentaires commencent toujours par #.

Q6) Quelle commande permet de visualiser les variables d'environnement du système ?

- a)** env
- b) ls
- c) var
- d) sysvar

Correction : La bonne réponse est **a)**. La commande `env` affiche toutes les variables d'environnement disponibles.

Partie 2 : Variables en shell (6 questions)

Q7) Quelle syntaxe est correcte pour affecter une valeur à une variable utilisateur ?

- a)** variable=value
- b) variable = value
- c) variable: value
- d) variable -> value

Correction : La bonne réponse est **a)**. En Bash, il ne doit pas y avoir d'espace autour du signe = lors de l'affectation d'une variable.

Q8) Qu'affiche la commande suivante ?

```
a=5
b=10
echo $((a + b))
```

- a) a + b
- b)** 15
- c) 510
- d) Rien

Correction : La bonne réponse est **b)**. L'opération arithmétique est effectuée grâce à `$(())`, ce qui donne 15.

Q9) Quelle commande permet d'afficher le contenu d'une variable appelée `nom` ?

- a)** echo \$nom
- b) print nom
- c) show \$nom
- d) display nom

Correction : La bonne réponse est **a)**. En Bash, `echo $nom` affiche la valeur stockée dans la variable `nom`.

Q10) Quel est le rôle de la variable spéciale `$?` ?

- a) Contient le PID du processus courant
- b) Contient le code de retour de la dernière commande exécutée**
- c) Contient le nombre d'arguments passés au script
- d) Contient la liste des arguments passés au script

Correction : La bonne réponse est **b)**. La variable `$?` stocke le code de retour de la dernière commande exécutée.

Q11) Quelle commande affiche toutes les variables d'environnement d'un script ?

- a) `set`**
- b) `unset`
- c) `env`
- d) `listvars`

Correction : La bonne réponse est **a)**. La commande `set` affiche toutes les variables (environnement et utilisateur).

Q12) Quelle commande permet de supprimer une variable utilisateur appelée `temp` ?

- a) `delete temp`
- b) `remove temp`
- c) `unset temp`**
- d) `clear temp`

Correction : La bonne réponse est **c)**. La commande `unset` supprime une variable en Bash.

Partie 3 : Contrôle de flux (6 questions)

Q13) Quelle syntaxe est correcte pour une condition `if` ?

a)

```
if [ condition ]; then
    commandes
fi
```

b)

```
if condition then
    commandes
end
```

c)

```
if condition {  
    commandes  
}
```

d) Toutes les propositions

Correction : La bonne réponse est **a)**. En Bash, la syntaxe correcte pour `if` inclut `[condition]` suivi de `then` et `fi`.

Q14) Quelle commande retourne un code de retour nul ?

- a) Une commande qui échoue
- b) Une commande qui réussit**
- c) Une commande inexistante
- d) Aucune des propositions

Correction : La bonne réponse est **b)**. Une commande réussie retourne un code de retour nul (0).

Q15) Quelle commande teste si un fichier appelé `test.txt` existe ?

- a) `[-f test.txt]`**
- b) `[-e test.txt]`
- c) Les réponses a et b
- d) Aucune des propositions

Correction : La bonne réponse est **c)**. Les deux options vérifient l'existence d'un fichier (`-f` pour les fichiers réguliers et `-e` pour tous les types).

Q16) Quel opérateur logique correspond à "ou" en Bash ?

- a) `-o`**
- b) `||`
- c) Les réponses a et b
- d) Aucune des propositions

Correction : La bonne réponse est **c)**. Les opérateurs `-o` et `||` signifient "ou" en Bash, selon le contexte.

Q17) Quelle est la syntaxe correcte d'une boucle `while` ?

a)

```
while [ condition ]; do  
    commandes  
    incrémentation  
done
```

b)

```
while { condition }  
  commandes  
end
```

c)

```
while condition {  
  commandes  
}
```

d) Toutes les propositions

Correction : La bonne réponse est **a)**. La syntaxe correcte d'une boucle `while` en Bash utilise `[condition]`, `do` et `done`.

Q18) Quelle est la différence entre une boucle `while` et une boucle `until` ?

- a) `while` s'exécute tant que la condition est fausse
- b) `until` s'exécute tant que la condition est vraie
- c) `while` s'exécute tant que la condition est vraie et `until` jusqu'à ce qu'elle soit vraie
- d) Aucune différence

Correction : La bonne réponse est **c)**. `while` s'exécute tant que la condition est vraie, tandis que `until` s'exécute tant qu'elle est fausse.

Partie 4 : Opérations sur les fichiers et redirection (6 questions)

Q19) Quelle commande permet de lister le contenu d'un fichier appelé `data.txt` ?

- a) `cat data.txt`
- b) `less data.txt`
- c) `more data.txt`
- d) Toutes les propositions

Correction : La bonne réponse est **d)**. Toutes ces commandes permettent d'afficher le contenu d'un fichier, mais de différentes manières.

Q20) Que fait la commande `>` en Bash ?

- a) Redirige le flux de sortie vers un fichier en mode ajout
- b) Redirige le flux de sortie vers un fichier en mode écrasement
- c) Redirige le flux d'entrée d'un fichier
- d) Aucune des propositions

Correction : La bonne réponse est **b)**. La commande `>` écrase le contenu du fichier cible avec la sortie redirigée.

Q21) Quelle est la différence entre `>` et `>>` ?

- a) > remplace le contenu du fichier, >> ajoute à la fin
- b) > ajoute à la fin, >> remplace le contenu du fichier
- c) Les deux font la même chose
- d) Aucune des propositions

Correction : La bonne réponse est **a)**. > écrase le fichier, tandis que >> ajoute du contenu à la fin.

Q22) Quelle commande permet d'afficher les 10 dernières lignes d'un fichier appelé `log.txt` ?

- a) `tail log.txt`
- b) `tail -n 10 log.txt`
- c) Les réponses a et b
- d) Aucune des propositions

Correction : La bonne réponse est **c)**. Par défaut, `tail` affiche les 10 dernières lignes, mais `tail -n 10` est une alternative explicite.

Q23) Quelle commande permet de compter le nombre de lignes d'un fichier ?

- a) `wc -l fichier`
- b) `lines fichier`
- c) `count fichier`
- d) `grep -n fichier`

Correction : La bonne réponse est **a)**. La commande `wc -l` compte le nombre de lignes dans un fichier.

Q24) Quel symbole est utilisé pour connecter deux commandes avec un tube (pipe) ?

- a) `|`
- b) `>`
- c) `<`
- d) `>>`

Correction : La bonne réponse est **a)**. Le symbole `|` est utilisé pour rediriger la sortie d'une commande comme entrée d'une autre.

Partie 5 : Commandes avancées et diverses (6 questions)

Q25) Quelle commande permet d'exécuter un script à intervalle régulier ?

- a) `cron`
- b) `at`
- c) `scheduler`
- d) `loop`

Correction : La bonne réponse est **a)**. `cron` permet de planifier l'exécution de scripts ou de commandes à des intervalles réguliers.

Q26) Quelle commande affiche les processus en cours d'exécution ?

- a) `ps`
- b) `jobs`
- c) `top`
- d) Toutes les propositions

Correction : La bonne réponse est **d)**. Les commandes `ps`, `jobs` et `top` affichent des informations sur les processus en cours, mais sous des formats différents.

Q27) Quelle commande permet de terminer un processus ayant pour PID 1234 ?

- a) `kill 1234`
- b) `end 1234`
- c) `stop 1234`
- d) `close 1234`

Correction : La bonne réponse est **a)**. La commande `kill` permet de terminer un processus en spécifiant son PID.

Q28) Quel est le rôle de la commande `chmod` ?

- a) Changer le propriétaire d'un fichier
- b) Changer les droits d'accès d'un fichier
- c) Les réponses a et b
- d) Aucune des propositions

Correction : La bonne réponse est **b)**. La commande `chmod` modifie les droits d'accès des fichiers ou répertoires.

Q29) Que signifie le modificateur `+x` avec `chmod` ?

- a) Supprime les droits d'exécution
- b) Ajoute les droits d'exécution
- c) Remplace les droits d'exécution
- d) Aucune des propositions

Correction : La bonne réponse est **b)**. Le modificateur `+x` ajoute les permissions d'exécution au fichier.

Q30) Quelle commande permet d'afficher l'espace disque utilisé par un fichier ou un dossier ?

- a) `du`
- b) `df`
- c) `ls -lh`
- d) `space`

Correction : La bonne réponse est **a**). La commande `du` affiche l'utilisation disque d'un fichier ou d'un répertoire.