

Structures de Contrôle

If et Elif

```
#!/bin/bash
x=5
if [ $x -eq 0 ];
then
    echo "Valeur de x : 0."
elif [ $x -eq 5 ];
then
    echo "Valeur de x : 5."
elif [ $x -eq 9 ];
then
    echo "Valeur de x : 9."
else
    echo "Valeur non reconnue"
fi
```

```
biotope% ./script.sh
Valeur de x : 5.
```

17

Structures de Contrôle

Case

```
#!/bin/bash
x=5
case $x in
    0) echo "Valeur de x : 0."
    ;;
    5) echo "Valeur de x : 5."
    ;;
    9) echo "Valeur de x : 9."
    ;;
    *) echo "Valeur non reconnue."
esac
```

```
biotope% ./script.sh
Valeur de x : 5.
```

18

Structures de Contrôle

For

```
#!/bin/bash

for i in 1 2 3 4 5;
do
    echo $i
done
```

```
biotope% ./script.sh
1
2
3
4
5
```

```
#!/bin/bash

for i in a b c d e;
do
    echo $i
done
```

```
biotope% ./script.sh
a
b
c
d
e
```

19

Structures de Contrôle

For

```
balustrade
cascade
sandale
vandale
chenal
```

```
biotope% cat mots.txt
balustrade
cascade
sandale
vandale
chenal
```

```
#!/bin/bash

for i in `cat mots.txt`;
do
    echo $i
done
```

```
biotope% ./script.sh
balustrade
cascade
sandale
vandale
chenal
```

20

Structures de Contrôle

For

```
balustrade cascade
sandale
vandale
chenal
```

```
biotope% cat mots.txt
balustrade cascade
sandale
vandale
chenal
```

```
#!/bin/bash
```

```
for i in `cat mots.txt`;
do
    echo $i
done
```

```
biotope% ./script.sh
balustrade
cascade
sandale
vandale
chenal
```

Structures de Contrôle

For

```
biotope% ls -l
total 16
-rw----- 1 gorsen  lasso      42 Nov  3 01:42 mots.txt
-rw----- 1 gorsen  lasso     213 Nov  2 16:15 notes.txt
-rwx----- 1 gorsen  lasso      51 Nov  3 01:34 script.sh
```

```
#!/bin/bash
for i in `ls`;
do
    if [ -r $i ];
    then
        echo $i est lisible
    fi
done
```

```
biotope% ./script.sh
mots.txt est lisible
notes.txt est lisible
script.sh est lisible
```

Structures de Contrôle

For

```
biotope% ls -l
total 16
-rw----- 1 gorsen  lasso      42 Nov  3 01:42 mots.txt
-rw----- 1 gorsen  lasso     213 Nov  2 16:15 notes.txt
-rwx----- 1 gorsen  lasso      51 Nov  3 01:34 script.sh
```

```
#!/bin/bash
for i in *; # Plus besoin de `ls`
do
    if [ -r $i ];
    then
        echo $i est lisible
    fi
done
```

```
biotope% ./script.sh
mots.txt est lisible
notes.txt est lisible
script.sh est lisible
```

Structures de Contrôle

For

```
crepetna:Crepet, Nathalie:CREN1807750:92:87:88:54:70
yosnheat:Yos Nhean, Trakal:YOST19087603:84:73:70:50:73
benelaur:Benel, Aurelien:BENA80207700:84:73:89:67:100
soucypas:Soucy, Pascal:SOUP14067502:95:90:89:87:99
```

```
biotope% cat notes.txt | cut -d: -f1
crepetna
yosnheat
benelaur
soucypas
```


Structures de Contrôle

While

```
#!/bin/bash
compteur=0

while [ $compteur -lt 10 ];
do
    echo -n $compteur
    compteur=`expr $compteur + 1`
done
```

```
biotope% ./script.sh
0123456789
biotope%
```

Guillemets

Introduction au Bash

Guillemets

Types

- Trois types de guillemets :
 - ' Ex. 'chaîne de caractères'
 - " Ex. "chaîne de caractères"
 - ` Ex. `commande shell`
- Les ' et " on sensiblement le même usage, cependant, on note une différence dans l'évaluation des variables, en effet :
 - **echo "valeur de \$variable"**
 - **echo 'valeur de \$variable'**
 Donneront des résultats tout à fait différents

Guillemets

```
biotope% expr 1 + 1
2
```

```
#!/bin/bash
compteur=0

while [ $compteur -lt 10 ];
do
    echo $compteur
    compteur=`expr $compteur + 1`
done
```

```
biotope% ./script.sh
0
1
2
3
4
5
6
7
8
9
biotope%
```

Guillemets

" et '

```
#!/bin/bash
compteur=79

echo "valeur du compteur : $compteur"
echo 'valeur du compteur : $compteur'
echo "valeur du compteur : `expr $compteur + 1`"
echo 'valeur du compteur : `expr $compteur + 1`'
```

```
biotope% ./script.sh
valeur du compteur : 79
valeur du compteur : $compteur
valeur du compteur : 80
valeur du compteur : `expr $compteur + 1`
```

33

Guillemets

Chaînes de Caractères

```
#!/bin/bash
var1="une cascade"
var2="des sandales"
if [ "$var1" = "$var2" ];
then
    echo Les deux chaînes de caractères sont identiques
else
    echo Les deux chaînes de caractères sont différentes
fi
```

```
biotope% ./script.sh
Les deux chaînes de caractères sont différentes
```

34

Guillemets

Chaînes de Caractères

```
#!/bin/bash
var1="une cascade"
var2="des sandales"
if [ "$var1" != "$var2" ];
then
    echo Les deux chaînes de caractères sont différentes
else
    echo Les deux chaînes de caractères sont identiques
fi
```

```
biotope% ./script.sh
Les deux chaînes de caractères sont différentes
```

35

Arithmétique et Booléens

Introduction au Bash

Arithmétique et Booléens Opérateurs

• Opérateurs arithmétiques

- Addition : +
- Soustraction : -
- Multiplication : *
- Division : /
- Modulo : %

• Opérateurs booléens

- Et : &&
- Ou : ||
- Non : !

Arithmétique et Booléens Exemple

```
#!/bin/bash
x=5
y=3
add=$((x + y))
sou=$((x - y))
mul=$((x * y))
div=$((x / y))
mod=$((x % y))
echo "Somme : $add"
echo "Différence : $sou"
echo "Produit : $mul"
echo "Quotient : $div"
echo "Reste : $mod"
```

```
biotope% ./script.sh
Somme : 8
Différence : 2
Produit : 15
Quotient : 1
Reste : 2
```

Arithmétique et Booléens Exemple

```
#!/bin/bash
x=5
y=10
if [ $x -eq 5 ] && [ $y -eq 10 ];
then
    echo Les deux conditions sont satisfaites
else
    echo Les deux conditions ne sont pas satisfaites
fi
```

```
biotope% ./script.sh
Les deux conditions sont satisfaites
```

Arithmétique et Booléens Exemple

```
#!/bin/bash
x=5
y=11
if [ $x -eq 5 ] && [ $y -eq 10 ];
then
    echo Les deux conditions sont satisfaites
else
    echo Les deux conditions ne sont pas satisfaites
fi
```

```
biotope% ./script.sh
Les deux conditions ne sont pas satisfaites
```

Arithmétique et Booléens

Exemple

```
#!/bin/bash
x=6
y=11
if [ $x -eq 5 ] || [ $y -eq 10 ];
then
    echo Une des deux conditions est satisfaite
else
    echo Aucune des deux conditions n'est satisfaite
fi
```

```
biotope% ./script.sh
Aucune des deux conditions n'est satisfaite
```

41

Arithmétique et Booléens

Exemple

```
#!/bin/bash
var1="une cascade"
var2="des sandales"
if [ "$var1" != "$var2" ];
then
    echo Les deux chaînes de caractères sont différentes
else
    echo Les deux chaînes de caractères sont identiques
fi
```

```
biotope% ./script.sh
Les deux chaînes de caractères sont différentes
```

42

Lecture de l'Entrée Standard

Introduction au Bash

Lecture de l'Entrée Standard

```
#!/bin/bash

echo -n "Entrez votre nom : "
read nom
echo "Bonjour $nom"
```

```
biotope% ./script.sh
Entrez votre nom : Nicolas
Bonjour Nicolas
biotope% ./script.sh
Entrez votre nom: Nicolas Gorse
Bonjour Nicolas Gorse
```

44

Lecture de l'Entrée Standard

```
#!/bin/bash
echo -n "Entrez une liste de répertoires à créer : "
read liste
for i in $liste;
do
    mkdir $i
    chmod 755 $i
done
echo Répertoires créés
```

```
biotope% ./script.sh
Entrez une liste de répertoires à créer : a b c d e f g h
Répertoires créés
```

45

Lecture de l'Entrée Standard

```
biotope% ls -l
Total 92
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 a
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 b
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 c
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 d
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 e
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 f
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 g
drwxr-xr-x  2 gorsen  lasso    4096 Nov  4 02:28 h
```

46

Lecture de l'Entrée Standard

```
#!/bin/bash
echo -n "Entrez votre nom : "
read nom
if [ -z $nom ];
then
    echo "Vous n'avez pas de nom !?"
else
    echo "Bonjour $nom"
fi
```

```
biotope% ./script.sh
Entrez votre nom :
Vous n'avez pas de nom !?
```

47

Lecture de l'Entrée Standard

```
#!/bin/bash
boucle=1
while [ $boucle -eq 1 ];
do
    echo -n "Entrez votre nom : "
    read nom
    if [ -n $nom ];
    then
        boucle=0
    fi
done
echo Merci $nom
```

48

Lecture de l'Entrée Standard

```

biotope% ./script.sh
Entrez votre nom :
Entrez votre nom : Nicolas
Merci Nicolas

```

Lecture de l'Entrée Standard

```

#!/bin/bash
boucle=1
compte=1
while [ $boucle -eq 1 ];
do
    echo -n "Entrez votre nom : "
    read nom
    if [ -n $nom ];
    then
        boucle=0
    else
        compte=$((compte + 1))
    fi
done

```

Suite

```

if [ $compte -eq 1 ];
then
    echo Bravo, du premier coup.
elif [ $compte -lt 5 ];
then
    echo Mouais, pas mal...
else
    echo Catastrophique \!
fi
echo Au fait, votre nom est $nom

```

Lecture de l'Entrée Standard

```

biotope% ./script.sh
Entrez votre nom : Nicolas
Bravo, du premier coup.
Au fait, votre nom est Nicolas

```

```

biotope% ./script.sh
Entrez votre nom :
Entrez votre nom :
Entrez votre nom :
Entrez votre nom : Nicolas
Mouais, pas mal...
Au fait, votre nom est Nicolas

```

Lecture de l'Entrée Standard

```

biotope% ./script.sh
Entrez votre nom :
Entrez votre nom : Nicolas
Catastrophique !
Au fait, votre nom est Nicolas

```