



I106B

17. Arithmétique

[HTTPS://WEB.ARCHIVE.ORG/WEB/20161119161957/HTTP://RYANSTUTORIALS.NET:80/BASH-SCRIPTING-TUTORIAL/BASH-ARITHMETIC.PHP](https://web.archive.org/web/20161119161957/http://ryanstutorials.net:80/bash-scripting-tutorial/bash-arithmetic.php)

let

2

- ▶ Les variables sont toutes de type *string*.
- ▶ Il faut expliciter quand on veut que les variables soient interprétées comme des entiers.
- ▶ Commande **let** : évaluation d'une instruction d'affectation

→ Cf. `help let`

Exemples

3

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

```
# Basic arithmetic using let
```

```
let a=5+4 # aucun espace
```

```
echo $a # 9
```

```
let "a = 5 + 4" # espaces permis si guillemets
```

```
echo $a # 9
```

```
let a++
```

```
echo $a # 10
```

```
let b=2*a # $ non nécessaire pour accès en lecture
```

```
echo $b # 20
```

```
let "c = 30 + $1"
```

```
echo $c # 30 + 1er argument de la ligne de commande
```

- ▶ Comme `let` mais :
 - Expression plutôt qu'instruction
 - Renvoie le résultat sur son `stdout`
 - Tous les paramètres doivent être séparés par des espaces
 - Cf. `man expr`

Exemples

5

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

```
# Basic arithmetic using expr
```

```
expr 5 + 4      # 9 --> espaces requis
```

```
expr "5 + 4"    # 5 + 4 --> chaîne non évaluée!
```

```
expr 5+4        # 5+4 --> chaîne non évaluée!
```

```
expr 5 \* $1     # 5 * 1er argument  
# (attention au globbing!)
```

```
expr 11 % 2      # 1
```

```
a=$(expr 10 - 3)  # equivalent à: let a=10-3
```

```
echo $a           # 7
```

```
expr 10 + $a      # 17
```

```
# $ nécessaire pour accès en lecture
```

Syntaxe (())

6

► `$(expr expression)` ↔ `$((expression))`

```
a=$( (4 + 5) )
```

```
b=$( (10 * a) ) # pas de globbing
```

```
echo $a $b # 9 90
```

► `let assignment` ↔ `((assignment))`

```
(( a = 4 + 5 ))
```

```
(( b = 10 * a ))
```

```
echo $a $b # 9 90
```