

---

# expr

Évalue des expressions et écrit le résultat sur stdout

**expr** évalue une expression et écrit le résultat sur la sortie standard. les opérandes sont soit des entiers soit des chaînes. Les chaînes ne sont pas entre guillemet pour **expr**, mais peut être requis pour le shell. Une opérande chaîne ne devrait pas être un parenthèse ou un opérateur de **expr**. Vous ne devriez pas passer d'entier négatif ou de chaîne avec un '-' final comme premier argument de **expr**. Cela peut être évité avec des parenthèses.

## Codes de status

- 0 si l'expressions est ni null ni 0
- 1 si l'expression est null ou 0
- 2 si l'expression est invalide
- 3 si une erreur interne s'est produite.

## Expressions chaîne

**STRING : REGEX** Effectue une correspondance de motif. Les arguments sont convertis en chaîne et le second est considéré comme une expression régulière, avec un '^' implicite devant. s'il y'a correspondance et que **REGEX** utilise '(' et ')', retourne la partie de la chaîne qui correspond. Dans l'expression régulière, '+', '?', et '|' sont des opérateurs qui correspondent respectivement à une correspondance d'un ou plus, zéro ou un, et séparés.

**match STRING REGEX** correspondance de motif alternatif. identique au précédent.

**substr STRING POSITION LENGTH** retourne le sous-chaîne de **STRING** commençant à **POSITION** avec **LENGH** longueur. si **POSITION** ou **LENGTH** sont des valeurs négatives, retourne une chaîne null.

**index STRING CHARSET** retourne la première position de **STRING** où le premier caractère dans **CHARSET** a été trouvé. retourne 0 si rien n'est trouvé.

**length STRING** retourne la longueur de **STRING**

**+ TOKEN** Interprète **TOKEN** comme une chaîne, même si c'est un mot clé comme **MATCH** ou un opérateur comme '/' pour que **expr** interprète les mots clés comme chaîne, vous devez utiliser ""

## Expressions numériques

**+ -** addition et soustraction. les arguments sont convertis en entiers ; une erreur se produit s'ils ne peuvent être convertis.

**\* / %** mulitplication, division, modulo. les arguments sont convertis en entiers ; une erreur se produit s'ils ne peuvent être convertis

## Relations pour expr

**|** retourne le premier argument s'il est non null ni zéro. sinon le deuxième argument s'il est non null ni zéro.

**&** retourne le premier argument s'il est non null ni zéro, sinon retourne 0. N'évalue pas le deuxième argument.

---

< <= == != > > > Compare les arguments et retourne 1 si la relation est vrai, 0 sinon. == est équivalent à =.

## Exemples

ajouter 1 à la variable shell foo

**foo='expr \$foo + 1'**

afficher la partie non-répertoire du nom de fichier stocké dans \$fname

**expr \$fname : '.\*\.[.\*\]' '1' \$fname**

exemple montrant que \+ est un opérateur

**expr aaa : 'a\+'**

affiche 3

**expr abc : 'a(\.)c'**

affiche b

**expr index abcdef cz**

affiche 3