# Découverte du langage GREL : exercices

V.1

Dans la colonne **cote**, ouvrir le menu *Editer les cellules > Transformer*

Quelle est l’expression saisie par défaut ? Valider et constater l’effet sur les données.

Saisissez les formules suivantes, sans valider. Observez la prévisualisation des résultats et notez les dans le tableau. Consultez l’aide mémoire GREL ou le wiki Openrefine en cas de besoin.

## Concepts de base

|  |  |
| --- | --- |
| value | Copie du contenu actuel |
| =value | Erreur |
| value+z | null |
| value+"z" | Ajout de z au contenu actuel |
| z | null |
| "z" | z |
| "y"+"z" | yz |
| 'y'+"z" | yz |
| 'y'+1945 | y1945 |
| 1945+1000 | 2945 |
| 1945/2 | 972 |
| 1=1 | Null |
| 1==1 | true |
| "a"=="b" | false |
| "a"!="b" | true |

## Chaînes de caractères

|  |  |
| --- | --- |
| "Bonjour".length() | 7 |
| length("Bonjour") | 7 |
| "Bonjour"[0] | B |
| "Bonjour"[6] | r |
| "Bonjour"[7] | Erreur |
| "Bonjour"[-1] | r |
| "Bonjour"[-2] | u |
| "Bonjour"[0,1] | B |
| "Bonjour"[1,2] | o |
| "Bonjour"[1,7] | onjour |
| "Bonjour"[1,-1] | onjou |
| "Bonjour".slice(1,7) | onjour |
| "Bonjour".substring(1,7) | onjour |
| "Bonjour".indexOf("o") | 1 |
| "Bonjour".lastIndexOf("o") | 4 |
| "Bonjour".toUppercase() | BONJOUR |
| "Bonjour".toLowercase() | Bonjour |
| "Bonjour".replace("B","b") | bonjour |
| "Bonjour".replace(/./,"b") | bbbbbbb |
| "Bonjour".split("o") | [ "B", "nj", "ur" ] |
| "Bonjour".split(/[nu]/) | [ "Bo", "jo", "r" ] |
| splitByLengths("01/04/2016", 2,3,5) | [ "01", "/04", "/2016" ] |
| "né le 1/2/2019".find("/") | [ "/", "/" ] |
| "né le 1/2/2019".find(/[0-9]+/) | [ "1", "2", "2019" ] |
| "né le 1/2/2019".find(/[0-9]+\/[0-9]+\/[0-9]+/) | [ "1/2/2019" ] |
| "né le 1/2/2019".match(/né.\*/) | [] |
| "né le 1/2/2019".match(/(le.\*)/) | null |
| "né le 1/2/2019".match(/né le\s+(.\*)/) | [ "1/2/2019" ] |

## Tableaux

|  |  |
| --- | --- |
| [4,2,"bc","ab",3,null].length() | 6 |
| [4,2,"bc","ab",3,null][0] | 4 |
| [4,2,"bc","ab",3,null].slice(1,3) | [ 2, "bc" ] |
| [4,2,"bc","ab",3,null].reverse() | [ null, 3, "ab", "bc", 2, 4 ] |
| [4,2,"bc","ab",3,null].sort() | erreur |
| [4,2,"bc","ab",3,null].join("") | 42bcab3 |
| [4,2,"bc","ab",3,null].join(":") | 4:2:bc:ab:3 |
| [4,2,3].sort() | [ 2, 3, 4 ] |
| ["bc","ab","cde"].sort() | [ "ab", "bc", "cde" ] |
| ["Bc","ab","cde"].sort() | [ "Bc", "ab", "cde" ] |
| ["bc","àb","cde"].sort() | [ "bc", "cde", "àb" ] |
| ["A","b","a","B"].uniques() | [ "A", "a", "b", "B" ] |
| ["a","b","a","b"].uniques() | [ "a", "b" ] |
| filter (["ab","c","def"],  v,length(v) > 1) | [ "ab", "def" ] |

## Boucles et conditions

|  |  |
| --- | --- |
| forEach(  ["A","b","a","B"],x,x) | [ "A", "b", "a", "B" ] |
| forEach(  ["A","b","a","B"],x,toLowercase(x)) | [ "a", "b", "a", "b" ] |
| if (1==2,"a","b") | b |