

Types de données

Chaque valeur en Red est d'un type donné. Les valeurs Red sont fortement typées, mais les variables (les mots utilisés pour se référer aux valeurs) ne le sont pas. Les valeurs sont le fondement de Red. Tandis que la plupart des langages ont 6 à 8 types de données de base, Red en a presque 50. Beaucoup d'entre eux ont des formes littérales uniques. Il y a certains détails d'implémentation dont, dans la plupart des cas, vous n'avez pas à vous préoccuper. Par exemple, `[native! action! function! routine!]` sont tous des types de fonctions, et vous pouvez les utiliser sans connaître leur type.

Pour utiliser Red efficacement, vous devriez devenir familier avec tous les types de données disponibles.

Table des types de données

- - indique qu'une valeur n'a pas de forme littérale. Ces valeurs doivent être créées avec `make` ou avec une syntaxe de construction (qui reste à définir pour de nombreux types).

| Type | Forme littérale | Description |
|------------------------|---|---|
| <code>action!</code> | - | "Méthodes" du modèle orienté objet interne de Red. |
| <code>binary!</code> | <code>2#{11111111}, 16#{2A}, 64#{2AQQ}</code> | Séries d'octets. |
| <code>bitset!</code> | - | Tableaux de bits utilisé pour stocker des valeurs booléennes. |
| <code>block!</code> | <code>[], [one 2 "three"], [print 1.23]</code> | Collections de données ou de code qui peuvent être évaluées à n'importe quel moment. |
| <code>char!</code> | <code>#"z", #"^C", #"^(esc)"</code> | Points de code (caractères) Unicode. |
| <code>datatype!</code> | <code>bitset!, string!, time!, word!</code> | Types valides qui sont prédéfinis pour Red. |
| <code>date!</code> | <code>1999-10-5, 5/10/1999, 5-October-1999</code> | Dates calendaires, basées sur le calendrier Grégorien. |
| <code>email!</code> | <code>adams@fourty-two.net, gregg@red-lang.org</code> | Adresses email brutes. |
| <code>error!</code> | - | Valeurs de type <code>object!</code> spéciales utilisées en cas d'erreur. |
| <code>event!</code> | - | Valeurs similaires à des objets, qui représentent une activité externe. En lecture seule. |
| <code>file!</code> | <code>%foo, %alongfilename, %/C/</code> | Noms de fichiers, ou noms de répertoires et de chemins d'accès. |
| <code>float!</code> | <code>+123.4, -123.4, 0042.0, 60'000'12'3.4</code> | Nombres à virgule positifs ou négatifs codés sur 64 bits. |

| | | |
|-------------|----------------------------------|---|
| function! | - | Fonctions définies par l'utilisateur. |
| get-path! | :foo/bar/baz | Valeurs de type path! non évaluées. |
| get-word! | :foo | Valeurs de type word! non évaluées. |
| handle! | - | Entiers cachés utilisés pour l'interaction avec le système d'exploitation. |
| hash! | - | Interface de type bloc permettant des recherches rapides. |
| image! | - | Séries de valeurs RGBA qui représentent des pixels dans les images 2D. |
| integer! | 1234, +1234, -1234, 60'000'000 | Nombres sans virgule codés sur 32 bits. |
| issue! | #555-555-5555, #ABC-123-xyx, #ID | Hashtags, mots-clés, et identifiants qui ne contiennent pas d'espace. |
| lit-path! | 'foo/bar/baz | Valeurs de type path! précédées d'une apostrophe, non évaluées. |
| lit-word! | 'foo | Valeurs de type word! précédées d'une apostrophe, non évaluées. |
| logic! | true false, yes no, on off | Valeurs booléennes. |
| map! | #(), #(a: 1 b: 2) | Tableau associatif de paires clé/valeur. |
| native! | - | Fonctions externes écrites en Red/System. |
| none! | - | Valeurs qui n'appartiennent à aucun autre type de données. Analogue à nil ou nothing. |
| object! | - | Contextes nommés ou non, qui contiennent des paires mot: valeur (word: value). |
| op! | - | Fonctions infixe à deux arguments. |
| pair! | 42x42 | Coordonnées bidimensionnelles. |
| paren! | (one 2 "three" 4:00:00) | block! immédiatement évalués. |
| path! | foo/bar/baz | Séries de valeurs délimitées par des "slash" /. Les types de valeurs qu'elles peuvent contenir sont limités. |
| percent! | 100%, 0.76% | Nombres à virgule exprimés sous forme de pourcentages. |
| refinement! | /foo | Valeurs symboliques utilisées comme modificateurs de fonctions ou extensions d'objets, fichiers, urls ou chemins. |
| routine! | - | Fonctions ayant une spécification en Red et un corps en Red/System. |

| | | |
|-----------|---|---|
| set-path! | foo/bar/baz: | Définit une "référence contextuelle" (mot dans un contexte, position d'index dans une série, ou référence correspondant à une clé dans une série ou un tableau de type <code>map!</code>). |
| set-word! | foo: | Définit une référence à une valeur. |
| string! | "foo" | Séquences de points de code Unicode (valeurs de type <code>char!</code>) encadrées de guillemets. |
| tag! | <foo> | Tags et éléments de langages à balises. |
| time! | 0:01:00, 3:02:01, 0:05:00.45 | Valeurs horaires brutes. Heures, minutes, secondes et millisecondes. |
| tuple! | 1.2.3, 255.255.255.0, 1.2.3.4.5.6.7.8.9.10.11.12 | Valeurs de couleurs RGB et RGBA adresses IP, et numéros de versions. Trois à douze entiers positifs séparés par des points décimaux. |
| typeset! | series!, number! | Ensembles, ou collections, de valeurs <code>datatype!</code> prédéfinies. |
| unset! | - | Valeur unique qui représente l'absence de valeur utilisable. |
| url! | https://www.red-lang.org, ftp://localhost | URLs (adresses de ressource) brutes. |
| vector! | - | Séquences ordonnées de valeurs d'un type identique (<code>char!</code> , <code>integer!</code> , <code>percent!</code> , <code>float!</code>). |
| word! | foo | Valeurs symboliques qui peuvent être utilisées comme des variables. Identifiant qui fait référence à une valeur dans un contexte donné. |