

Datatypes

Each value in Red is of a specific datatype. Red values are strongly typed, but variables (words used to refer to values), are not. Values are the foundation of Red. Where most languages have 6-8 base datatypes, Red has almost 50. Many of these have unique, literal forms. There are some implementation details that, for the most part, you don't need to worry about. For example, `[native! action! function! routine!]` are all types of functions, and you can use them without knowing the type.

To use Red effectively, you should become familiar with all the datatypes available.

Datatypes Table

- - indicates no literal form for a value. These values must be created with `make` or with construction syntax (which is TBD for many types).

Type	Literal Form	Description
<code>action!</code>	-	Datatype "methods" in the Red internal OO model.
<code>binary!</code>	<code>2#{11111111}, 16#{2A}, 64#{2AQQ}</code>	Series of bytes.
<code>bitset!</code>	-	Array of bits used to store boolean values.
<code>block!</code>	<code>[], [one 2 "three"], [print 1.23]</code>	Collections of data or code that can be evaluated at any point in time.
<code>char!</code>	<code>"z", "^C", #"^(esc)"</code>	Unicode code points.
<code>datatype!</code>	<code>bitset!, string!, time!, word!</code>	Valid types that are predefined for Red.
<code>date!</code>	<code>1999-10-5, 5/10/1999, 5-October-1999</code>	Calendar dates, relying on the Gregorian calendar.
<code>email!</code>	<code>adams@fourty-two.net, gregg@red-lang.org</code>	Directly expressed email addresses.
<code>error!</code>	-	Specialized <code>object!</code> values used for error conditions.
<code>event!</code>	-	Object-like values that represent external activity. Effectively read-only.
<code>file!</code>	<code>%foo, %alongfilename, %/C/</code>	File names, or directory names and paths.
<code>float!</code>	<code>+123.4, -123.4, 0042.0, 60'000'12'3.4</code>	64-bit positive or negative number that contains a decimal point.
<code>function!</code>	-	User-defined functions.
<code>get-path!</code>	<code>:foo/bar/baz</code>	Unevaluated <code>path!</code> value.
<code>get-word!</code>	<code>:foo</code>	Unevaluated <code>word!</code> value.

handle!	-	Opaque integers used for interacting with the operating system.
hash!	-	Block-like interface with fast lookups.
image!	-	Series of RGBA values that represent pixels in 2D images.
integer!	1234, +1234, -1234, 60'000'000	32-bit numbers with no decimal point.
issue!	#555-555-5555, #ABC-123-xyx, #ID	Hashtags, keywords, and identifiers that do not contain spaces.
lit-path!	'foo/bar/baz	Unevaluated (quoted) path! value.
lit-word!	'foo	Unevaluated word! value.
logic!	true false, yes no, on off	Boolean values.
map!	#(), #(a: 1 b: 2)	Associative array of key/value pairs.
money!	\$123, -USD\$1'234.56789	Amount of a specific (or generic) currency.
native!	-	External functions written in Red/System.
none!	-	Value that does not belong to any other datatype. Analogous to nil or nothing.
object!	-	Named or unnamed contexts that contain word: value pairs.
op!	-	Infix function of two arguments.
pair!	42x42	Two-dimensional coordinates.
paren!	(one 2 "three" 4:00:00)	Immediately evaluated block! .
path!	foo/bar/baz	Series of values delimited by slashes /. Limited in the types of values that they can contain.
percent!	100%, 0.76%	Float values expressed as percentages.
ref!	@foo	Denotes a reference by name.
refinement!	/bar	Symbolic values used as modifiers to functions or extensions to objects, files, urls, or paths.
routine!	-	Function with a Red spec and Red/System body.
set-path!	foo/bar/baz:	Sets a "contextual reference" (word in a context, indexes slot in a series, or keyed reference in a series or map).
set-word!	foo:	Sets a reference to a value.
string!	"foo"	Sequence of Unicode code points (char! values) wrapped in quotes.
tag!	<foo>	Markup language tags and elements.
time!	0:01:00, 3:02:01, 0:05:00.45	Directly expressed time values. Hours, minutes, seconds, and milliseconds.

tuple!	1.2.3, 255.255.255.0, 1.2.3.4.5.6.7.8.9.10.11.12	RGB and RGBA color values, ip addresses, and version numbers. Three to twelve positive integers separated by decimal points.
typeset!	series! , number!	Set, or collection, of predefined datatype! values.
unset!	-	Single value that represents no usable value.
url!	https://www.red-lang.org, ftp://localhost	Directly expressed Uniform Resource Locators.
vector!	-	Ordered sequence of values of identical type (char! , integer! , percent! , float!).
word!	foo	Symbolic value that can be used like a variable. Identifier that refers to a value in a given context.