Typesety

Table of Contents

1. Úvodem . . . 1
2. Aktuálně existující typesety: . . . 1
   2.1. all-word! . . . 1
   2.2. any-block! . . . 2
   2.3. any-function! . . . 2
   2.4. any-list! . . . 2
   2.5. any-object! . . . 2
   2.6. any-path! . . . 2
   2.7. any-string! . . . 2
   2.8. any-type! . . . 2
   2.9. any-word! . . . 2
   2.10. default! . . . 2
   2.11. external! . . . 3
   2.12. immediate! . . . 3
   2.13. internal! . . . 3
   2.14. number! . . . 3
   2.15. scalar! . . . 3
   2.16. series! . . . 3

1. Úvodem

Typesety jsou sady hodnot typu `datatype!`, uložených v kompaktní řadě (array) bitů (max 96 bitů), umožňujících podporu vysokorychlostní kontroly typů při runtime.

Datové typy v typesetu mohou mít společné rysy nebo chování ale není to podmínkou. Typeset lze vytvořit v závislosti na kriteriích, vyhovujících potřebám uživatele.

Viz: typeset!

2. Aktuálně existující typesety:

2.1. all-word!

- make typeset! [word! set-word! lit-word! get-word! refinement! issue!]

2.2. any-block!

- make typeset! [block! paren! path! lit-path! set-path! get-path! hash!]

2.3. any-function!

- make typeset! [native! action! op! function! routine!]

2.4. any-list!

- make typeset! [block! paren! hash!]

2.5. any-object!

- make typeset! [object! error!]

2.6. any-path!

- make typeset! [path! lit-path! set-path! get-path!]

2.7. any-string!

- make typeset! [string! file! url! tag! email! ref!]

2.8. any-type!

- make typeset! [datatype! unset! none! logic! block! paren! string! file! url! char! integer! float! word! set-word! lit-word! get-word! refinement! issue! native! action! op! function! path! lit-path! set-path! get-path! routine! bitset! object! typeset! error! vector! hash! pair! percent! tuple! map! binary! time! tag! email! handle! date! image! event!]

2.9. any-word!

- make typeset! [word! set-word! lit-word! get-word!]

2.10. default!

- make typeset! [datatype! none! logic! block! paren! string! file! url! char! integer! float! word! set-word! lit-word! get-word! refinement! issue! native! action! op! function! path! lit-path! set-path! get-path! routine! bitset! object! typeset! error! vector! hash! pair! percent! tuple! map! binary! time! tag! email! handle! date! image! event!]

2.11. external!

- make typeset! [event!]

2.12. immediate!

- make typeset! [datatype! none! logic! char! integer! float! word! set-word! lit-word! get-word! refinement! issue! typeset! pair! percent! tuple! time! handle! date!]

2.13. internal!

- make typeset! [unset! float! percent!]

2.14. number!

- make typeset! [integer! float! percent!]

2.15. scalar!

- make typeset! [char! integer! float! pair! percent! tuple! time! date! money!]

2.16. series!

- make typeset! [block! paren! string! file! url! path! lit-path! set-path! get-path! vector! hash! binary! tag! email! image!]

  Slovo `series` má v prostředí jazyka Red dva významy. Jednak to je (s vykřičníkem) označení typesetu (viz výše), jednak to je (bez vykřičníku) označení druhu objektu. Řada (series) je v Redu definována jako sekvence prvků, jež má počáteční pozici, která může být posunována z první pozice (`head`) k poslední pozici (`tail`). Počátek prázdné řady je na poslední pozici (`tail`). Jednotlivé pozice jsou označeny pořadovým číslem - indexem.

```
>> a: "hello"
== "hello"

>> b: next a
== "ello"

>> index? a
== 1

>> index? b
== 2

>> same? a b
== false

>> same? a head b
== true

>> append a " world"
== "hello world"

>> b
== "ello world"
```