

□□□□□□□□□□□□□□□□VID□

□□

1. □□	2
2. □□□□□	3
3. □□□□□□	3
3.1. title	3
3.2. size	3
3.3. backdrop	4
4. □□□□□	4
4.1. across	4
4.2. below	5
4.3. return	5
4.4. space	5
4.5. origin	5
4.6. at	6
4.7. pad	6
4.8. do	6
5. □□□□□□□	6
5.1. h1	6
5.2. h2	7
5.3. h3	7
5.4. h4	7
5.5. h5	7
5.6. box	7
5.7. image	7
6. □□□□□□□	7
6.1. □□□□□	8
6.1.1. left	8
6.1.2. center	8
6.1.3. right	8
6.1.4. top	8
6.1.5. middle	9
6.1.6. bottom	9
6.1.7. bold	9
6.1.8. italic	9
6.1.9. underline	9
6.1.10. extra	10
6.1.11. data	10

6.1.12. draw	10
6.1.13. font	10
6.1.14. para	11
6.1.15. wrap	11
6.1.16. no-wrap	11
6.1.17. font-size	11
6.1.18. font-color	12
6.1.19. font-name	12
6.1.20. react	12
6.1.21. loose	13
6.1.22. all-over	13
6.1.23. hidden	13
6.1.24. disabled	13
6.1.25. select	13
6.1.26. focus	14
6.1.27. hint	14
6.1.28. rate	14
6.1.29. default	14
6.2. 窗口	15
6.3. 对话框	15
7. 容器	16
7.1. panel	16
7.2. group-box	16
7.3. tab-panel	17
8. 样式	17
8.1. style	17

1. 简介

VID 是 Visual Interface Dialect 的缩写，是 Red 语言的一个 DSL。

VID 是一个用于描述用户界面的 DSL。

- 描述用户界面
- 描述布局
- 描述样式

VID 是一个用于描述用户界面的 DSL，它使用 `window` 来描述窗口。

VID 是一个用于描述用户界面的 DSL，它使用 `layout` 来描述布局，使用 `view` 来描述视图。

NOTE

VID는 Redox의 help view와 help layout를 사용합니다

2. 구조

VID의 구조는 다음과 같습니다

```
[  
  <view>  
  <layout>  
]
```

- view: view의 구조를 정의합니다
- layout: layout의 구조를 정의합니다

NOTE

VID는 Redox의 help view와 help layout를 사용합니다

3. 사용법

NOTE

react는 Redox의 help view와 help layout를 사용합니다

3.1. title

예

```
title <text>  
  
<text> : string!
```

예

Redox의 help view와 help layout를 사용합니다

3.2. size

예

```
size <value>  
  
<value> : pair!
```

예

Redox의 help view와 help layout를 사용합니다

3.3. backdrop

00

backdrop <color>

```
<color> : color word! tuple! issue!
```

11

□ □

4.

VID

- 
- 

□ □ □ □ □ □ □ □ □ □ □ □

- 10x10
- 10x10
- across
- top

across

[across] | *across.png*

below left

[below] | *below.png*

4.1. across

11

across <alignment>

```
<alignment> : 0000000000000000 top | middle | bottom
```

11

1topalignment

4.2. below

11

below <alignment>

```
<alignment> : 000000000000 left | center | right
```

11

[illegible]

4.3. return

11

```
return <alignment>
```

```
<alignment> : 000000000000 left | center | right | top | middle | bottom
```

11

```

alignment

```

4.4. space

11

space <offset>

```
<offset> : 00000000pair!0
```

11

[illegible]

4.5. origin

00

origin <offset>

```
<offset> : 00000000pair!0
```

11

[illegible]

4.6. at

例

```
at <offset>
```

```
<offset> : 0からpair!の
```

例

0から20までの偶数と、20から100までの奇数の総和を計算する。20は偶数で、100は奇数である。

4.7. pad

例

```
pad <offset>
```

```
<offset> : 0からpair!の
```

例

0から100までの偶数の総和を計算する。

4.8. do

例

```
do <body>
```

```
<body> : code to evaluate (block!).
```

例

Redのselfメソッドは、selfを返す。

5. 変数

変数は、変数宣言で宣言される。VIDは、変数宣言のキーワードである。styleは、変数のスタイルである。

5.1. h1

H1は、32ビットのtext変数。

[illegible]

NOTE

window □□□□□□□□□□□□□□□□□□□□

6.1. □□□□□

6.1.1. left

11

left

11

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

6.1.2. center

11

center

11

□ □

6.1.3. right

11

right

11

□□□□□□□□□□□□□□□□□□□□

6.1.4. top

00

top

11

□□□□□□□□□□□□□□□□□□□□ top □□□□□

6.1.5. middle

□□

middle

□□

□□□□□□□□□□□□□□□□ middle □□□□

6.1.6. bottom

□□

bottom

□□

□□□□□□□□□□□□□□□□ bottom □□□□

6.1.7. bold

□□

bold

□□

□□□□□□□□□□□□□□□□ bold □□□□

6.1.8. italic

□□

italic

□□

□□□□□□□□□□□□□□□□ italic □□□□

6.1.9. underline

□□

underline

□□

□□□□□□□□□□ underline □□□□

6.1.10. extra

11

```
extra <value>
```

```
<value> : any-type!
```

11

extra

6.1.11. data

11

```
data <list>
```

```
<list> : [] [] [] [] [] [] [] [] block! []
```

11

data [REDACTED]

6.1.12. draw

11

```
draw <commands>
```

```
<commands> : [][][][]block![]
```

11

draw Draw Draw

6.1.13. font

00

font <spec>

```
<spec> : ██████████block! object! word!
```

00

font font! font! font!

NOTE

font VID

6.1.14. para

11

para <spec>

```
<spec> :   para      block!  object!  word!
```

11

para para! para! para!

NOTE

¶ para ¶ para ¶ VID ¶

6.1.15. wrap

11

wrap

11

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

6.1.16. no-wrap

11

no-wrap

11

□ □

6.1.17. font-size

11

font-size <pt>

```
<pt> : integer! word!
```

11

□ □

6.1.18. font-color

11

font-color <value>

```
<value> :         tuple! word! issue! 
```

11

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

6.1.19. font-name

11

font-name <name>

```
<name> : []string! word!
```

11

□ □

6.1.20. react

react

11

```
react [<body>]
```

```
react later [<body>]
```

```
<body> :   Redblock!
```

00

react face
word react
later body

NOTE:

View `set-path`

[illegible]

11

11

11

11

11

11

11

default <value>

```
<value>: 'data' []any-type![]
```

11

text none data options

NOTE □□□□□ text □ field □□□□□□□□□□□□□□□□

6.2. □□□□

[illegible]

名前	説明
integer!	整数値を返す
pair!	2つの値を組にしたペアを返す
tuple!	複数の値を組にしたタプルを返す
issue!	16進数のカラーコードを返す。#rgb, #rrggbb, #rrggbbaa
string!	文字列を返す
percent!	data をパーセントで表示。progress を slider を表示
logic!	data をチェックボックス、radio を表示
image!	画像を返す
url!	URL を返す
block!	ブロックを返す
get-word!	単語を返す
char!	文字を返す

6.3. □□□□

[illegible]

□ □

```
<actor>
```

on-<event> <actor>

```
<actor> : []block! get-word!
```

```
<event> : 00000000word!0
```

[illegible]

```
func [face [object!] event [event! none!]][...body...]
```

get-word

View [VID](#)

11

00

- `across()`
- `below()`

00

16

11

VID

- **across** $\frac{1}{n}$
- **below** $\frac{1}{n}$

NOTE `string!`

7.3. tab-panel

11

```
tab-panel <options> [<name> <body>...]
```

```
<options> : 0000000000000000
<name> : 00000000string!
<body> : 00000000VID0000block!
```

11

8.

8.1. style

00

```
style <new> <old> <options>
```

```
<new>      : 0000000000set-word!0
<old>      : 0000000000word!0
<options>  : 00000000000000000000
```

11

[illegible][illegible]