

□□□□□□□□□□□□□□□□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. 列表	16
7.1. panel	16
7.2. group-box	16
7.3. tab-panel	17
8. 菜单	17
8.1. style	17

1. 简介

VID 是 Visual Interface Dialect 的缩写，是 Red 的方言，用于描述 DSL 窗口。

VID 是 Visual Interface Dialect 的缩写，是 Red 的方言，用于描述 DSL 窗口。

- 窗口
- 对话框
- 列表

VID 是 Visual Interface Dialect 的缩写，是 Red 的方言，用于描述 DSL 窗口。

VID 是 Visual Interface Dialect 的缩写，是 Red 的方言，用于描述 DSL 窗口。

NOTE

VID는 Redox의 help view와 help layout를 포함합니다

2. 구조

VID의 구조는 다음과 같습니다

```
[  
  <div>  
  <div>  
]
```

- `div`: `div` 태그를 사용하여 구조를 정의합니다
- `div`: `div` 태그를 사용하여 구조를 정의합니다

NOTE

VID는 Redox의 help view와 help layout를 포함합니다

3. 사용법

NOTE

`react`는 `react` 태그를 사용하여 구조를 정의합니다

3.1. title

예

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

예

구조를 정의합니다

3.2. size

예

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

예

구조를 정의합니다

3.3. backdrop

00

backdrop <color>

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

11

[illegible]

4.

VID

- [illegible]

□ □ □ □ □ □ □ □ □ □ □ □

- □□□□ 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>` : 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

□ □

4.6. at

例

```
at <offset>
```

```
<offset> : 0から255までのpair!
```

例

0から255までのpairを2回繰り返す。つまり、0から255までのpairを1回繰り返す。

4.7. pad

例

```
pad <offset>
```

```
<offset> : 0から255までのpair!
```

例

0から255までのpairを2回繰り返す。つまり、0から255までのpairを1回繰り返す。

4.8. do

例

```
do <body>
```

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

例

Redのselfメソッドは、selfを返す。つまり、selfを返す。

5. 5.1. h1

5.1. h1は、h1のスタイルを指定する。つまり、h1のスタイルを指定する。

5.1. h1

H1のスタイルは、32ビットのtext。

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

11

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

6.1.12. draw

11

```
draw <commands>
```

```
<commands> : 00000000b!lock!
```

11

draw Draw Draw

6.1.13. font

00

font <spec>

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

□ □

font font! font! font!

NOTE

font VID

6.1.14. para

11

para <spec>

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

11

para para! para! para para

NOTE

para para VID

6.1.15. wrap

11

wrap

11

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

6.1.16. no-wrap

□ □

no-wrap

11

□ □

6.1.17. font-size

11

font-size <pt>

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

□ □

□ □

6.1.18. font-color

11

font-color <value>

```
<value> : 00000000tuple! word! issue!
```

11

□ □

6.1.19. font-name

11

font-name <name>

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

11

[illegible]

6.1.20. react

react

11

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

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

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

00

react face
 word react
 later body

NOTE:

View `set-path`

6.1.21. loose

11

loose

11

□ □

6.1.22. all-over

11

all-over

11

all-over

6.1.23. hidden

□□

hidden

11

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

6.1.24. disabled

11

disabled

11

[illegible]

6.1.25. select

11

```
select <index>
```

```
<index> : 0000000000000000integer!
```

[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つの値の組を返す
tuple!	複数の値の組を返す
issue!	16進数のRGB値を返す#rgb, #rrggbb, #rrggbbaa
string!	文字列を返す
percent!	data 値 progress slider 値
logic!	data 値 check radio 値
image!	画像データを返す
url!	URLを返す
block!	ブロックデータを返す
get-word!	単語データを返す
char!	文字データを返す

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 [this video](#) on YouTube

11

00

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

00

16

11

VID

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

NOTE `string!`

7.3. tab-panel

00

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

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

00

[illegible]

8.

8.1. style

00

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

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

□ □

[illegible][illegible]