

# □□□□□□□□□□□□□□□□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 的方言。它提供了一种 DSL 来描述 UI。

VID 提供了一种简单的方式来描述 UI 的布局、样式和行为。

- 简单易学
- 易于集成
- 易于扩展

VID 提供了一种简单的方式来描述 UI 的布局、样式和行为。window 是 VID 中的一个基本概念。

VID 提供了一种简单的方式来描述 UI 的布局、样式和行为。layout 是 VID 中的一个基本概念。view 是 VID 中的一个基本概念。

## 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.

XXXXXXXXXXVIDXX

- 
- 

□ □ □ □ □ □ □ □ □ □ □ □

- □□□□ 10x10
- □□□□ 10x10
- □□□ across
- □□□□□□□ top

across

[across] | *across.png*

below left

[below] | *below.png*

## 4.1. across

11

across <alignment>

```
<alignment> : 000000000000 top | middle | bottom
```

11

1topalignment

## 4.2. below

11

below <alignment>

```
<alignment> : [] [] [] [] [] [] [] [] 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pair!の
```

例

0から20までの偶数と、21から40までの奇数のペアを生成する。20は偶数で、40は奇数で、ペアの数は20個。

## 4.7. pad

例

```
pad <offset>
```

```
<offset> : 0からpair!の
```

例

0から20までの偶数と、21から40までの奇数のペアを生成する。20は偶数で、40は奇数で、ペアの数は20個。

## 4.8. do

例

```
do <body>
```

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

例

Redのselfメソッドは、selfを返す。selfはselfを返す。

## 5. 5.1. h1

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



## 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!`

## 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

00

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

11

[illegible]

### 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`

### 6.1.21. loose

□ □

loose

11

□ □

### 6.1.22. all-over

00

all-over

11

all-over

### 6.1.23. hidden

11

hidden

11

□ □

### 6.1.24. disabled

11

disabled

11

[illegible]

### 6.1.25. select

11

```
select <index>
```

```
<index> : 整数型integer!
```

[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 値を progress スライダーで表示
logic!	data 値を check ボックス、radio ボタンで表示
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]

actor

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

□□□□□□□□□□ □□ □□□□□□□□

get-word

7.

[illegible]View [VID](#)

### 7.1. panel

00

```
panel <options> [<content>]

<options> : 0000000000000000
<content> : 00000000VID000block![]
```

11

VIDE

- `\across`
- `\below`

## 7.2. group-box

00

```
group-box <divider> <options> [<body>]
```

**<divider>** : 10個のinteger!

**<options>** : 10個の

**<body>** : 1個のVID1個のblock!



VID

- NOTE**    `string!`

□□

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

[illegible]

11

[illegible]

17