

Red/View□□□□□□□□□□

□□

1. □□□□□□	2
2. □□□□□□□□	2
2.1. □□□□□□□□	4
3. □□□□□□□□	4
4. Para □□□□□□	5
5. □□□□□□	5
6. □□□□□□□□	5
6.1. Base	5
6.2. Text	6
6.3. Button	6
6.4. Check	7
6.5. Radio	7
6.6. Field	7
6.7. Area	8
6.8. Text-list	9
6.9. Drop-list	9
6.10. Drop-down	10
6.11. Progress	10
6.12. Slider	10
6.13. Camera	11
6.14. Panel	11
6.15. Tab-panel	11
6.16. Window	12
6.17. Screen	13
6.18. Group-box	13
7. □□□□□□□□□□	13
8. SHOW□□	13
9. □□□□□□ VS □□□□□□□□	14
10. □□□□□□□□□□	15
11. □□□□	15
11.1. □□□□□□	15
11.2. Event! □□□□	17
11.3. Actors	18
11.4. □□□□□□□□	19
11.5. □□□□□□□□□□□□	20
11.5.1. insert-event-func	20

11.5.2. remove-event-func	20
12. System/view object	21
13. View.....	21
14.	21

1.

Red/View.....View.....Red.....

-API
-
-
-
-
- OS.....
-

.....

2.

..... **face!** *(facet)*

.....:

.....
type	word! (.....)
offset	pair!
size	pair!
text	string!
image	image!
color	tuple!	R.G.B.....R.G.B.A.....
menu	block!
data	any-type!
enable?	logic!
visible?	logic!
selected	integer!

変数名	型	初期値	制約	説明
flags	block!, word!	0	0	フラグのビット列。初期値は0。
options	block!	0	0	[name: value]の辞書。初期値は空辞書。
parent	object!	0	0	親オブジェクトの参照。初期値は0。
pane	block!	0	0	パネの参照。初期値は0。
state	block	0	0	状態の辞書。初期値は空辞書。
rate	integer!, time!	0	0	整数型integerと時間型timeの辞書。初期値はnone。
edge	object!	0	0	エッジの参照。初期値は0。
para	object!	0	0	パラメータの参照。初期値は0。
font	object!	0	0	フォントの参照。初期値は0。
actors	object!	0	0	アクターの参照。初期値は0。
extra	any-type!	0	0	任意の型の辞書。初期値は空辞書。
draw	block!	0	0	描画の辞書。初期値はDraw。

flags の初期値:

変数名	型
all-over	辞書 over 辞書

辞書の初期値:

Notes:

- 辞書の初期値は none
- offset と size の辞書
- offset と size の辞書の初期値は none (辞書の初期値は0) の辞書
- 辞書の初期値は color、image、text、draw の辞書

辞書の初期値face!の辞書の初期値は 辞書の 辞書 type の辞書

```
button: make face! [type: 'button']
```

辞書の初期値 type の辞書

2.1. 文字列の操作

文字列の操作は、主に以下の方法で行われます。

操作	説明
drag-on	文字列の操作は、主に以下の方法で行われます。 'down' 'mid-down' 'alt-down' 'aux-down' などのキーを押下したときに実行される。

3. 文字列のフォーマット

文字列のフォーマットは、

font

文字列のフォーマットは、主に以下の方法で行われます。文字列のフォーマットは、主に以下の方法で行われます。文字列のフォーマットは、主に以下の方法で行われます。

操作	説明	操作	説明
name	string!	OS	OSのバージョン
size	integer!	OS	OSのバージョン
style	word!, block!	OS	OSのバージョン
angle	integer!	OS	OSのバージョン
color	tuple!	OS	OSのバージョン
anti-alias?	logic!, word!	OS	OSのバージョン
shadow	(reserved)	OS	OSのバージョン
state	block!	OS	OSのバージョン
parent	block!	OS	OSのバージョン

Notes:

- 文字列のフォーマットは、**none** の場合、文字列のフォーマットは、
- angle** の場合、文字列のフォーマットは、
- 文字列のフォーマットは、文字列のフォーマットは、

文字列のフォーマットは、

- bold**
- italic**
- underline**
- strike**

文字列のフォーマットは、

- 文字列 (anti-alias?: yes/no)
- 文字列 (anti-alias?: 'ClearType')

4. Para 属性

Para属性

para!

Para属性用于设置HTML元素的默认样式。1. 设置元素的默认样式。2. 设置元素的默认样式。

属性	值	说明
origin	属性	设置元素的默认样式
padding	属性	设置元素的默认样式
scroll	属性	设置元素的默认样式
align	word!	设置元素的默认样式: left, center, right
v-align	属性	设置元素的默认样式: top, middle, bottom
wrap?	logic!	设置元素的默认样式
parent	block!	设置元素的默认样式

Notes:

- Para属性: none 设置元素的默认样式

5. Pane 属性

Pane属性用于设置HTML元素的默认样式。

- pane 属性: 设置元素的默认样式
- parent 属性: 设置元素的默认样式

pane 属性: 设置元素的默认样式。Z 属性: 设置元素的默认样式。

pane

screen 属性: 设置元素的默认样式。pane 属性: 设置元素的默认样式。

screen 属性: 设置元素的默认样式。screen 属性: 设置元素的默认样式。

[Face tree] | [face-tree.png](#)

6. Base 属性

6.1. Base

base 属性: 设置元素的默认样式。base 属性: 设置元素的默认样式。

Facet	值
type	'base'

options	default
flags	block!

options

- **no-border** GUI

data text

- **text** data load text none options/default
- data text data form

options

- **default** text none data text

NOTE:

- **selected**

enter	on-enter	Enter
change	on-change	
key	on-key	

6.7. Area

options

Facet	
type	'area'
text	
flags	area block!

options

- **no-border** GUI

Notes:

- **selected**
- A vertical scroll-bar can appear if all lines of text cannot be visible in the area (might be controlled by a **flag** option in the future).
- area flag

Property	Event	Value
change	on-change	String
key	on-key	String

6.8. Text-list

Text-list is a list of text items. It is a simple list of text items that can be selected or not.

Facet	Value
type	'text-list'
data	String! hash!
selected	Boolean! none

Property	Event	Value
select	on-select	String! selected Boolean
change	on-change	select String! selected Boolean

Notes:

- Text-list is a simple list of text items that can be selected or not.

6.9. Drop-list

Drop-list is a list of text items that can be selected or not. It is a simple list of text items that can be selected or not.

Facet	Value
type	'drop-list'
data	String! hash!
selected	Boolean! none

data String! hash!
selected Boolean! 1 Boolean! data String! hash!

Property	Event	Value
select	on-select	String! selected Boolean
change	on-change	select String! selected Boolean

Notes:

- `selected` 属性用于指定当前选中的选项

6.10. Drop-down

下拉菜单（Drop-down）是一种常见的UI元素，用于展示一组选项，并允许用户从中选择一个值。

Facet	属性
<code>type</code>	<code>'drop-down'</code>
<code>data</code>	选项列表，格式为 <code>block! hash!</code>
<code>selected</code>	当前选中的选项，格式为 <code>none</code> 或选项的 <code>hash!</code>

`data` 属性用于指定下拉菜单的选项列表。选项列表的格式为 `block! hash!`，其中 `block!` 表示选项的块，`hash!` 表示选项的哈希值。例如，`data` 属性可以设置为 `block! hash!`，表示选项列表。

属性	事件	说明
<code>select</code>	<code>on-select</code>	当用户选择了一个选项时触发。该事件的 <code>selected</code> 属性将返回所选选项的 <code>hash!</code> 。
<code>change</code>	<code>on-change</code>	当用户更改了选项时触发。该事件的 <code>selected</code> 属性将返回所选选项的 <code>hash!</code> 。

Notes:

- `selected` 属性用于指定当前选中的选项

6.11. Progress

进度条（Progress）是一种常见的UI元素，用于显示任务的完成进度。

Facet	属性
<code>type</code>	<code>'progress'</code>
<code>data</code>	进度值，格式为 <code>percent!float!</code>

Notes:

- `float` 属性 `data` 用于指定进度值，范围从 `0.0` 到 `1.0`。

6.12. Slider

滑块（Slider）是一种常见的UI元素，用于允许用户通过拖动来调整数值。

Facet	属性
<code>type</code>	<code>'slider'</code>
<code>data</code>	滑块值，格式为 <code>percent!float!</code>

Notes:

- float **data** 0.0 1.0

6.13. Camera

Facet	
type	'camera'
data	
selected	data none

Notes:

- data none show
- to-image

6.14. Panel

Facet	
type	'panel'
pane	Z

Notes:

- offset
-

6.15. Tab-panel

Facet	
type	'tab-panel'
data	
pane	block!
selected	noneinteger!

6.17. Screen

[illegible]

6.18. Group-box

[illegible]

Facet	
type	'group-box
pane	000000000000000000000000Z00000000000

Notes:

- `offset` 偏移量
- 偏移量

7.

- [illegible]

Notes:

- [illegible]

8. SHOW□□

show <face>

11

state
pane

```

state
OS API

```

Notes:

- show **changes** **deferred**
- OS **handle!**

9. 〇〇〇〇〇〇 VS 〇〇〇〇〇〇〇〇〇

[illegible][illegible][illegible]

- [illegible]

10. □□□□□□□□□□

11

11. □□□□

11.1. □□□□□

15

名前	タイプ	説明
drop	オブジェクト	ドロップされたオブジェクト
click	オブジェクト	クリックされたオブジェクト
dbl-click	オブジェクト	ダブルクリックされたオブジェクト
over	オブジェクト	オブジェクトがマウスオーバーされたときに発生するイベント。2つのオブジェクト flags オブジェクト all-over オブジェクトがマウスオーバーされたときに発生するイベント
move	オブジェクト	移動されたオブジェクト
resize	オブジェクト	リサイズされたオブジェクト
moving	オブジェクト	移動中のオブジェクト
resizing	オブジェクト	リサイズ中のオブジェクト
wheel	オブジェクト	ホイールイベント
zoom	オブジェクト	ズームイベント
pan	オブジェクト	パンイベント
rotate	オブジェクト	回転イベント
two-tap	オブジェクト	ダブルタップイベント
press-tap	オブジェクト	プレスタップイベント
key-down	キーコード	キーが押されたとき
key	キーコード	キーが押されたときに発生するイベント。control、shift、menu キーコード
key-up	キーコード	キーが押されたとき
enter	キーコード	Enter キーが押されたとき
focus	オブジェクト	フォーカスが移ったオブジェクト
unfocus	オブジェクト	フォーカスが移ったオブジェクト
select	オブジェクト	選択されたオブジェクト
change	オブジェクト	変更されたオブジェクト。変更されたオブジェクトの値が変更されたときに発生するイベント
menu	オブジェクト	メニューが開かれたオブジェクト
close	オブジェクト	メニューが閉じられたオブジェクト
time	時間	イベントが発生した時間 rate オブジェクト

Notes:

- オブジェクトはWindows XPでサポートされています
- 1つの **moving** オブジェクト **move** オブジェクト
- 1つの **resizing** オブジェクト **resize** オブジェクト

11.2. Event! いろいろ

いろいろなイベントが用意されています。ここでは、いくつかのイベントについて説明します。

イベント名	説明
<code>type</code>	イベントの種類 (word!)
<code>face</code>	イベントの顔 (object!)
<code>window</code>	イベントのウィンドウ (object!)
<code>offset</code>	イベントのオフセット (pair!) 2つの整数のペア
<code>key</code>	イベントのキー (char! word!)
<code>picked</code>	イベントのピッキング (integer! percent! wheel! menu! ID! word! Windows! GESTUREINFO! fullArguments!)
<code>flags</code>	イベントのフラグ (block!)
<code>away?</code>	イベントが遠ざかっているかどうか (true logic! over)
<code>down?</code>	イベントが押されているかどうか (true logic!)
<code>mid-down?</code>	イベントが中クリックされているかどうか (true logic!)
<code>alt-down?</code>	イベントがAltキーが押されているかどうか (true logic!)
<code>ctrl?</code>	イベントがCtrlキーが押されているかどうか (true logic!)
<code>shift?</code>	イベントがShiftキーが押されているかどうか (true logic!)

`event/flags` イベントのフラグ

- `away`
- `down`
- `mid-down`
- `alt-down`
- `aux-down`
- `control`
- `shift`

Notes:

- `type` イベントの種類 (type) イベントの種類 (type)

`event/key` イベントのキー (word!)

- `page-up`
- `page-down`

- end
- home
- left
- up
- right
- down
- insert
- delete
- F1
- F2
- F3
- F4
- F5
- F6
- F7
- F8
- F9
- F10
- F11
- F12

key-down □ key-up □□□□□□□□□□□□□□□□ event/key □□□□□□□□

- left-control
- right-control
- left-shift
- right-shift
- left-menu
- right-menu

11.3. Actors

actors

[illegible]

11

```
on-<event>: func [face [object!] event [event!]]
```

```
<event> : []
```

```
face      : ████████████████████████████████████
```

```
event : []
```

GUI on-create on-create
create face

103

```
'stop' : []
```

```
'done' : []
```

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

11.4. □□□□□□

- [illegible]

[Event flow] | *event-flow.png*

□ □

1. `onmouseover` 事件在鼠标指针移到元素上时触发。
2. `onmouseout` 事件在鼠标指针移出元素时触发。
 - a. `onmouseover` 事件在鼠标指针移到元素上时触发 `on-detect` 事件。
 - b. `onmouseout` 事件在鼠标指针移出元素时触发 `on-detect` 事件。
 - c. `onmouseover` 事件在鼠标指针移到元素上时触发 `on-detect` 事件。
3. `onmousedown` 事件在鼠标按钮被按下时触发。
 - a. `onmousedown` 事件在鼠标按钮被按下时触发 `on-click` 事件。
 - b. `onmousedown` 事件在鼠标按钮被按下时触发 ``on-click`` 事件。
 - c. `onmousedown` 事件在鼠标按钮被按下时触发 `on-click` 事件。

Notes:

- `done word`
- `system/view/capturing?: yes`

11.5. 事件管理API

このセクションでは、イベント管理APIについて説明します。

11.5.1. insert-event-func

概要

```
insert-event-func <handler>

<handler> : 関数

func [face [object!] event [event!]]
```

説明

```
func [face [object!] event [event!]]
```

例

この関数は、イベント管理APIの一部です。関数は、face、object、eventの3つの引数を受け取ります。faceは、イベントが発生するオブジェクトの顔（face）を表します。objectは、イベントが発生するオブジェクトを表します。eventは、イベントの種類を表します。functionは、イベントが発生したときに実行される関数を表します。

関数は、以下のような値を返します。

- none : イベントが発生しなかったときに返す値。
- 'done : イベントが発生したときに返す値。
- 'stop : イベントが発生したときに返す値。

関数は、以下のような値を返します。

11.5.2. remove-event-func

概要

```
remove-event-func <handler>

<handler> : 関数
```

説明

この関数は、イベント管理APIの一部です。関数は、face、object、eventの3つの引数を受け取ります。faceは、イベントが発生するオブジェクトの顔（face）を表します。objectは、イベントが発生するオブジェクトを表します。eventは、イベントの種類を表します。functionは、イベントが発生したときに実行される関数を表します。

12. System/view object

Word	説明
screens	画面オブジェクトのリスト
event-port	イベントポート
metrics	メトリクス
platform	プラットフォーム
VID	VIDオブジェクト
handlers	ハンドラー
reactors	リアクター
evt-names	イベント名
init	初期化
awake	目覚め
capturing?	yes = 検出 detect している no していない
auto-sync?	yes = 自動同期 no = 手動同期
debug?	yes = デバッグ on no していない
silent?	yes = VIDDraw しない no する

13. Viewオブジェクト

Viewオブジェクトは、プラットフォーム、イベントポート、メトリクス、Redオブジェクト、Needsオブジェクトを持つ。

```
Red [
  Needs: 'View'
]
```

NOTE: red オブジェクトは、Viewオブジェクトを持つ。Needs

14. 関数

Function	説明
view	VIDオブジェクトを /no-wait で表示する
unview	表示を解除する
layout	VIDオブジェクトのレイアウト

Function	関数
center-face	顔の中心に配置する
dump-face	顔のデータをダンプする
do-actor	アクターの動作を実行する
do-events	イベントを実行する
draw	Draw関数でimageを描画する
to-image	faceをimageに変換する
size-text	テキストのサイズを設定する

関数の一覧

- 関数の一覧
- image!関数