

# 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	状態の参照。初期値は0。
rate	integer!, time!	0	0	整数型integerと時間型timeの参照。初期値は0。
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	任意の型の参照。初期値は0。
draw	block!	0	0	描画の参照。初期値は0。

flags の初期値は0:

変数名	型
all-over	0 over 初期値は0

初期値は0

Notes:

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

初期値は0face!初期値は0初期値は0 初期値は0 type 初期値は0

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

初期値は0 type 初期値は0



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

Notes:

- Para属性: none 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: 元素的默认样式
- parent 属性: 元素的默认样式

pane 属性: 元素的默认样式。Z元素的默认样式。

pane

screen 属性: screen 元素的默认样式 pane 属性: window 元素的默认样式

screen(window属性)元素的默认样式(screen 属性: screen 元素的默认样式)

[Face tree] | *face-tree.png*

## 6. Base 属性

### 6.1. Base

base 属性: 元素的默认样式 base 属性: 128.128.128 元素的默认样式

Facet	值
type	'base





options	options
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	options
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! block! 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! block! hash!
selected	Boolean! none

data String! block! hash!  
selected Boolean! 1 Boolean! data String! block! 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。

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]

<b>Facet</b>	□□
<b>type</b>	'screen'
<b>size</b>	□□□□□□□□□□□□□□□□□□□□□□□□□□ <b>pair!</b> □
<b>pane</b>	□□□□□□□□□□□□□□□□□□□□ <b>block!</b> □

## 6.18. Group-box

edge

Facet	
type	'group-box
pane	000000000000000000000000Z00000000000

Notes:

- `offset` 偏移量
- 偏移量

7.

1. face! []
2. []
3. show []
  - a. []
  - b. face/state []
4. []
5. []

Notes:

- free

## 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	説明	備考
<b>drop</b>	ドロップ	ドロップイベント
<b>click</b>	クリック	クリックイベント
<b>dbl-click</b>	ダブルクリック	ダブルクリックイベント
<b>over</b>	オーバー	マウスが要素の上に移動したときに発生するイベント。2つの イベント <b>flags</b> イベント <b>all-over</b> イベント
<b>move</b>	移動	移動イベント
<b>resize</b>	リサイズ	リサイズイベント
<b>moving</b>	移動中	移動中イベント
<b>resizing</b>	リサイズ中	リサイズ中イベント
<b>wheel</b>	ホイール	ホイールイベント
<b>zoom</b>	ズーム	ズームイベント
<b>pan</b>	パン	パンイベント
<b>rotate</b>	回転	回転イベント
<b>two-tap</b>	二重タップ	二重タップイベント
<b>press-tap</b>	プレスタップ	プレスタップイベント
<b>key-down</b>	キーダウン	キーダウンイベント
<b>key</b>	キー	キーイベント。control、shift、menu
<b>key-up</b>	キーアップ	キーアップイベント
<b>enter</b>	エンター	Enter
<b>focus</b>	フォーカス	フォーカスイベント
<b>unfocus</b>	アンフォーカス	アンフォーカスイベント
<b>select</b>	選択	選択イベント
<b>change</b>	変更	変更イベント
<b>menu</b>	メニュー	メニューイベント
<b>close</b>	クローズ	クローズイベント
<b>time</b>	時間	時間イベント <b>rate</b> イベント

Notes:

- Windows XP
- 1 **moving** イベント **move** イベント
- 1 **resizing** イベント **resize** イベント



## 11.2. Event!

type	(word!)
face	object!
window	object!
offset	pair!
key	char! word!
picked	integer! percent!wheel menu IDword! WindowsGESTUREINFOullArguments
flags	block!
away?	true logic! over
down?	true logic!
mid-down?	true logic!
alt-down?	true logic!
ctrl?	CTRLtrue logic!
shift?	SHIFTtrue logic!

event/flags

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

Notes:

- type

event/key 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

111

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

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

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

## 11.4. □□□□□□

[illegible]

- [illegible]

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

[illegible]

1. `onmouseover` 鼠标悬停时触发的事件
2. `onmouseout` 鼠标移出时触发的事件
  - a. `onmouseover` 和 `onmouseout` 事件在元素上同时绑定，当鼠标悬停时，`onmouseover` 事件触发，当鼠标移出时，`onmouseout` 事件触发。
  - b. `onmouseover` 和 `onmouseout` 事件在元素上同时绑定，当鼠标悬停时，`onmouseover` 事件触发，当鼠标移出时，`onmouseout` 事件触发。
  - c. `onmouseover` 和 `onmouseout` 事件在元素上同时绑定，当鼠标悬停时，`onmouseover` 事件触发，当鼠标移出时，`onmouseout` 事件触发。
3. `onmousedown` 鼠标按下时触发的事件
  - a. `onmousedown` 和 `onmouseup` 事件在元素上同时绑定，当鼠标按下时，`onmousedown` 事件触发，当鼠标松开时，`onmouseup` 事件触发。
  - b. `onmousedown` 和 `onmouseup` 事件在元素上同时绑定，当鼠标按下时，`onmousedown` 事件触发，当鼠标松开时，`onmouseup` 事件触发。
  - c. `onmousedown` 和 `onmouseup` 事件在元素上同时绑定，当鼠标按下时，`onmousedown` 事件触发，当鼠标松开时，`onmouseup` 事件触发。

Notes:

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

## 11.5. 事件管理

このセクションでは、イベント管理のAPIについて説明します。

### 11.5.1. insert-event-func

例

```
insert-event-func <handler>

<handler> : 関数

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

例

```
;; 関数
function!
```

例

この関数は、イベント管理のAPIの一部です。関数は、イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。関数は、イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。

関数は、イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。

- **none** : イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。
- **'done** : イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。
- **'stop** : イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。

関数は、イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。

### 11.5.2. remove-event-func

例

```
remove-event-func <handler>

<handler> : 関数
```

例

関数は、イベント管理のAPIの一部です。関数は、イベントの発生を監視し、特定のイベントが発生したときに実行される関数を登録します。

## 12. System/view object

Word	説明
screens	画面のリスト
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 で呼び出す
unview	VIDオブジェクトを呼び出す
layout	VIDオブジェクトを呼び出す

<b>Function</b>	関数
<b>center-face</b>	顔の中心に配置する
<b>dump-face</b>	顔のデータをダンプする
<b>do-actor</b>	アクターを実行する
<b>do-events</b>	イベントを実行する
<b>draw</b>	Draw関数でimageを描画する
<b>to-image</b>	faceをimageに変換する
<b>size-text</b>	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