

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:

変数名	初期値
all-over	0 over の初期値

初期値の定義:

Notes:

- 初期値の定義は none の初期値
- offset と size の初期値
- offset と size の初期値は none の初期値(初期値の初期値) の初期値
- 初期値の定義は color、image、text、draw の初期値

初期値の定義はface!の初期値の初期値の初期値 type の初期値

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

初期値の定義は type の初期値

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	文字列の先頭に 'down'、'mid-down'、'alt-down'、'aux-down' を追加する。& で区切る。

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、文字列の先頭に 'down'、'mid-down'、'alt-down'、'aux-down' を追加する。& で区切る。

文字列のフォーマット

操作	説明	操作	説明
name	string!	OS	OSの種類を指定する。
size	integer!	サイズ	文字のサイズを指定する。
style	word!, block!	スタイル	文字のスタイルを指定する。
angle	integer!	角度	文字の角度を指定する。
color	tuple!	色	R.G.B.またはR.G.B.Aで色を指定する。
anti-alias?	logic!, word!	アンチエイリアス	アンチエイリアスを有効にするかどうかを指定する。
shadow	(reserved)	影	文字に影を付けるかどうかを指定する。
state	block!	状態	文字の状態を指定する。
parent	block!	親	文字の親要素を指定する。

Notes:

- 文字列のフォーマットは **none** で指定する。
- angle** は文字の角度を指定する。
- 文字列のフォーマットは文字列の先頭に 'down'、'mid-down'、'alt-down'、'aux-down' を追加する。& で区切る。

文字列のフォーマット:

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

文字列のフォーマット:

- 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元素的显示方式。Para属性1个属性，用于设置HTML元素的显示方式。Para属性用于设置HTML元素的显示方式。

属性	值	说明
origin	属性	设置元素的显示方式
padding	属性	设置元素的显示方式
scroll	属性	设置元素的显示方式
align	word!	设置元素的显示方式: left, center, right
v-align	属性	设置元素的显示方式: top, middle, bottom
wrap?	logic!	设置元素的显示方式
parent	block!	设置元素的显示方式: 设置元素的显示方式

Notes:

- Para属性: none 设置元素的显示方式

5. Pane 属性

Pane属性用于设置HTML元素的显示方式。Pane属性1个属性，用于设置HTML元素的显示方式。

- pane 属性: 设置元素的显示方式
- parent 属性: 设置元素的显示方式

pane 属性: 设置元素的显示方式。pane 属性: 设置元素的显示方式。

pane

screen 属性: 设置元素的显示方式。screen 属性: 设置元素的显示方式。

screen 属性: 设置元素的显示方式。screen 属性: 设置元素的显示方式。

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

6. Base 属性

6.1. Base

base 属性: 设置元素的显示方式。base 属性: 设置元素的显示方式。

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

Facet	Property	Value
change	on-change	on-change
key	on-key	on-key

6.8. Text-list

Text-list is a list of text items. It is a simple list of text items, each with a label and a value. The list is rendered as a single line of text, with each item separated by a space. The list is rendered as a single line of text, with each item separated by a space.

Facet	Property
type	'text-list'
data	block! hash!
selected	selected

Facet	Property	Value
select	on-select	on-select
change	on-change	on-change

Notes:

- Text-list is a simple list of text items, each with a label and a value. The list is rendered as a single line of text, with each item separated by a space. The list is rendered as a single line of text, with each item separated by a space.

6.9. Drop-list

Drop-list is a list of text items. It is a simple list of text items, each with a label and a value. The list is rendered as a single line of text, with each item separated by a space. The list is rendered as a single line of text, with each item separated by a space.

Facet	Property
type	'drop-list'
data	block! hash!
selected	selected

data selected 1 data selected

Facet	Property	Value
select	on-select	on-select
change	on-change	on-change

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中的选项列表。每个选项由 `block!` 和 `hash!` 组成。例如，`data = block! hash!` 表示一个选项，其 `block!` 部分为 `data`，`hash!` 部分为 `1`。当 `selected` 属性设置为 `data` 时，表示该选项被选中。

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。此时 <code>selected</code> 属性会更新为所选选项的 <code>hash!</code> 。
<code>change</code>	<code>on-change</code>	当用户更改了选中的选项时触发。此时 <code>selected</code> 属性会更新为所选选项的 <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

Facet	□□
type	'screen'
size	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□pair!□
pane	□□□□□□□□□□□□□□□□□□□□block!□

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

- `getenv()`
- `setenv()` `show`

[illegible][illegible]

- [illegible]

- `Rebol/View`

series Red

API series

```
view [
  list: text-list data ["John" "Bob" "Alice"]
  button "Add" [append list/data "Sue"]
  button "Change" [lowercase pick list/data list/selected]
]
```

11.1. □□□□□

15

名前	説明	備考
drop	ドロップ	ドラッグ＆ドロップ
click	クリック	単発のクリック
dbl-click	ダブルクリック	ダブルクリック
over	オーバー	マウスが要素の上にある状態。2つの要素が重なっている場合 flags は all-over となる
move	移動	移動
resize	リサイズ	リサイズ
moving	移動中	移動中
resizing	リサイズ中	リサイズ中
wheel	ホイール	ホイール
zoom	ズーム	ズーム
pan	パン	パン
rotate	回転	回転
two-tap	二重タップ	二重タップ
press-tap	押しタップ	押しタップ
key-down	キーダウン	キーダウン
key	キー	control、shift、menu キー
key-up	キーアップ	キーアップ
enter	エンター	Enter キー
focus	フォーカス	フォーカス
unfocus	アンフォーカス	アンフォーカス
select	選択	選択
change	変更	変更
menu	メニュー	メニュー
close	クローズ	クローズ
time	時間	時間 rate は速度

Notes:

- Windows XP 以降
- 1回の **moving** は **move** のみ
- 1回の **resizing** は **resize** のみ

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

```
insert-event-func function!
```

例

この関数は、イベント管理APIの一部です。イベント管理APIは、イベントを登録、削除、実行するためのAPIです。

関数の引数は、次のとおりです。

- **none** : イベント管理APIの一部です。
- **'done** : イベント管理APIの一部です。
- **'stop** : イベント管理APIの一部です。

関数の戻り値は、イベント管理APIの一部です。

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。

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

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	オブジェクトを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