

11

1.	□□□□□	1
1.1.	□□□	2
1.2.	□□□□□□□□□□	2
1.3.	□□□□□□□□□□	3
2.	API	4
2.1.	react	4
2.2.	is	4
2.3.	react?	5
2.4.	clear-reactions	5
2.5.	dump-reactions	6
3.	□□□□□□□□□□	6
3.1.	reactor!	6
3.2.	deep-reactor!	6

1.

□□□□□

0.6.0 Red Red Red
Red object-oriented
reactive programming

API

[react-simple] | *react-simple.png*[illegible][react-graph] | *react-graphs.png*

CDE

react/unlink clear-reaction

NOTE:

- [illegible]

- Red FRP
- Red/View GUI face!

1.1. □□□

名前	氏名
住所	〒住所
電話番号	電話番号
メールアドレス	メールアドレス
職業	職業
趣味	趣味
備考	備考
署名	署名
捺印	捺印

1.2. □□□□□□□□□□

[illegible]

00

```
view [
  s: slider return
  b: base react [b/color/1: to integer! 255 * s/data]
]
```

slider base Red/View

□ □

```
vec: make reactor! [x: 0 y: 10]
box: object [length: is [square-root (vec/x ** 2) + (vec/y ** 2)]]
```

is is word

GUI
vec/x
vec/y
vector
vec

```
a: make reactor! [x: 1 y: 2 total: is [x + y]]
```

00

```
a: make reactor! [x: 1 y: 2]
total: is [a/x + a/y]
```

NOTE: The total number of respondents is 100.

react/link

```

;--
win: layout [
  size 400x500
  across
  style ball: base 30x30 transparent draw [fill-pen blue circle 15x15 14]
  ball ball ball ball ball ball ball b: ball loose
  do [b/draw/2: red]
]

follow: func [left right][left/offset/y: to integer! right/offset/y * 108%]

faces: win/pane
while [not tail? next faces][
  react/link :follow [faces/1 faces/2]
  faces: next faces
]
view win

```

3

2. API

2.1. react

11

```
react <code>  
react/unlink <code> <source>
```

```
react/link <func> <objects>
react/unlink <func> <source>
```

```
react/late <code>
```

```
<code>      : []  
<func>     : []  
<objects>  : []  
<source>   : ['allwordobject!block!']
```

Returns : `code` `func`

00

```

import React from 'react';
import { Link } from 'react-router-dom';
import './App.css';

function App() {
  return (
    <div>
      <h1>React Router</h1>
      <div>
        <Link to="/link">Link</Link>
        <Link to="/path">Path</Link>
      </div>
    </div>
  );
}

export default App;

```

react

/later

Red

[link](#) react

```
../../../../unlink../../../../<source>../../../../
```

- 'all'word
-
-

[/unlink](#)

2.2. is

11

```
<word>: is <code>
```

```
<word> : wordset-word!  
<code> : 1block!
```

11

is word <code>
2

NOTE: Excel

10

```
a: make reactor! [x: 1 y: 2 total: is [x + y]]

a/total
== 3

a/x: 100

a/total
== 102
```

```
a: make reactor! [x: 1 y: 2 total: is [x + y]]

a/total
== 3

a/x: 100

a/total
== 102
```

2.3. react?

□ □

```

react? <obj> <field>
react?/target <obj> <field>

<obj>    : 0000000000000000object!
<field>  : 0000000000000000000000word!

Returns : 00000000block!000function!  none!  00

```

```

react? <obj> <field>
react?/target <obj> <field>

<obj>      : 0000000000000000object!
<field>    : 0000000000000000word!

Returns : 000000block!000function!  none!  00

```

```

react? <obj> <field>
react?/target <obj> <field>

<obj>      : 0000000000000000object!
<field>    : 0000000000000000word!

Returns : 000000block!000function!  none!  00

```

11

```

react?
none
/target
none

```

2.4. clear-reactions

00

```
clear-reactions
```

11

□□□□□□□□□□□□□□□□□□□□

2.5. dump-reactions

11

dump-reactions

11

3. □□□□□□□□□□

Red

3.1. reactor!

11

```
make reactor! <body>

<body> : []block![]

[]
```

11

[illegible]

NOTE: $\frac{1}{2}$ is $\frac{1}{2}$

3.2. deep-reactor!

11

```
make deep-reactor! <body>

<body> : []block![]

[]
```

00

```

series
series

```

NOTE: _____ is _____

1

series

NOTE:deep-reactor!isseries

```
r: make deep-reactor! [
  x: [1 2 3]
  y: [[a b] [c d]]
  total: is [append copy x copy y]
]
append r/y/2 'e
print mold r/total
```