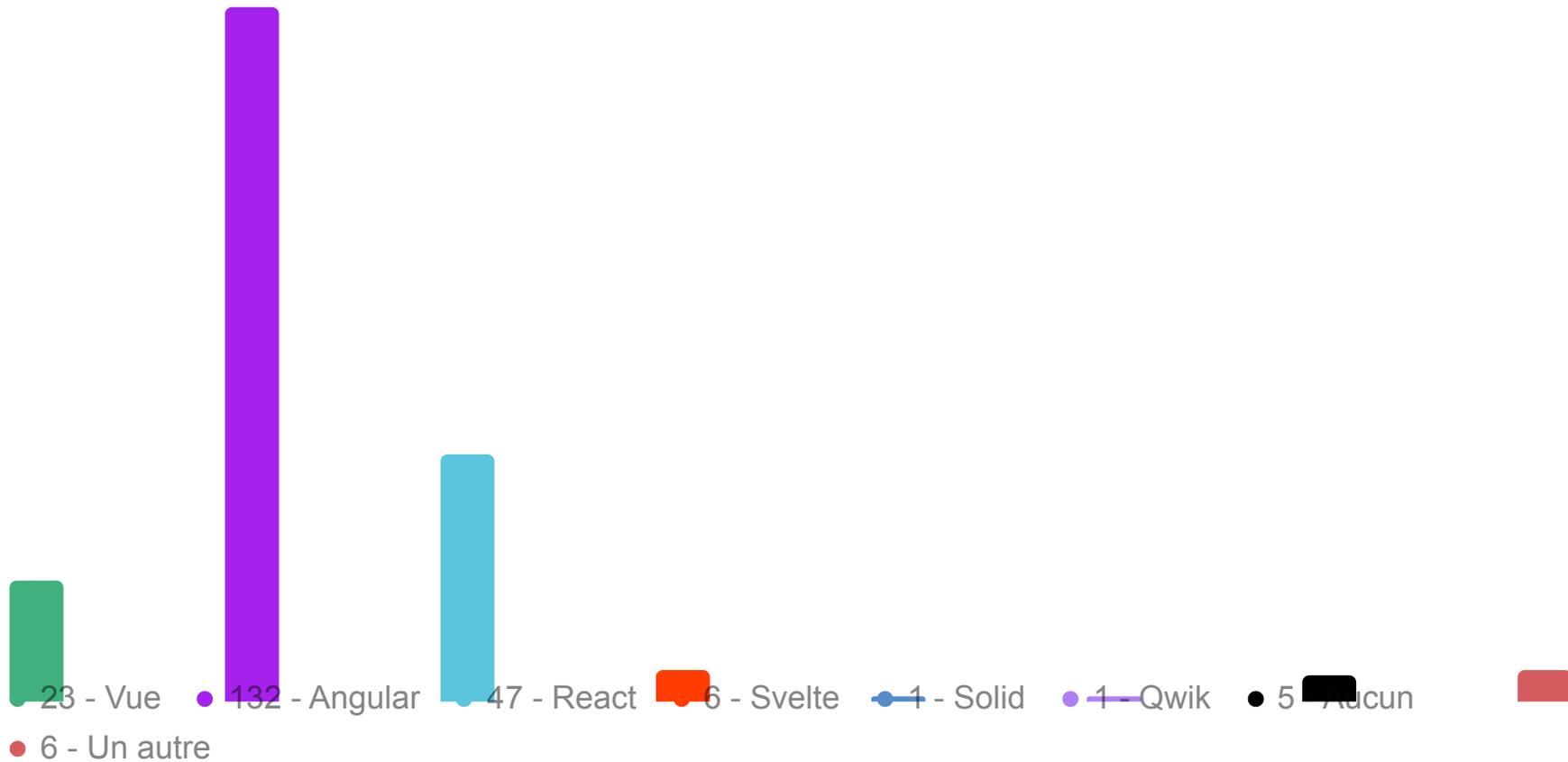


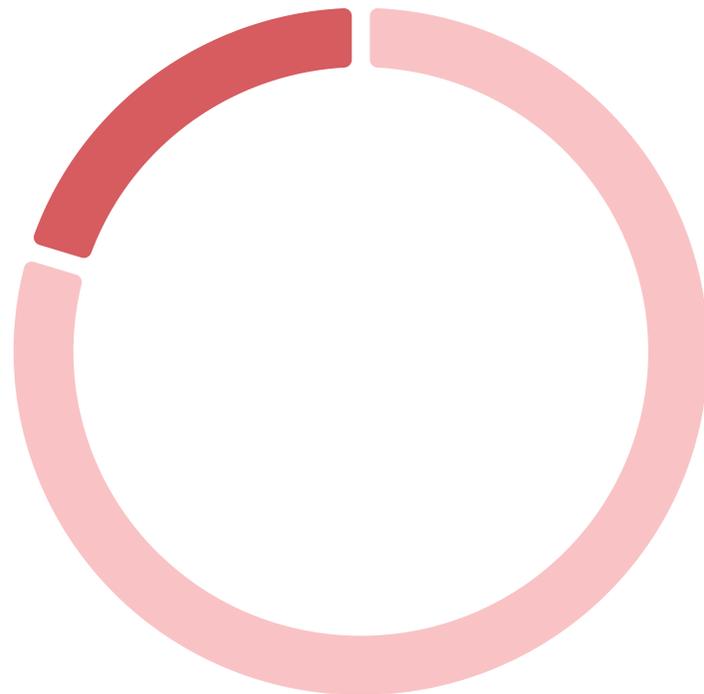
La réactivité et les signaux : démystifions la magie du frontend



Quel framework pour le frontend ?



Qui s'est déjà questionné sur le fonctionnement profond de la réactivité de son framework ?



● 153 - Oui ● 38 - Non

`ui = fn(state)`

	A	B	C
1	Quantité	0	
2	Prix unitaire	15	
3	Total	0	



```
import { computed, effect, signal } from 'alien-signals' ▶
```

```
const quantity = signal(0)
```

```
const price = signal(15)
```

```
const total = computed(() => quantity() * price())
```

```
effect(() => {  
  console.log(`Total: ${total()}`)  
})
```

```
//
```

```
Total: 0
```



Estéban Soubiran

Ingénieur logiciel Avionique chez  Maiaspace

 Laravel

 Vite

 Vue

 Nuxt

Using Pinia Colada in Modals Without Spoiling the UX

Pint, Rector, Larastan, and Pest: Essentials for Success

Setting Up Your Laravel Project Structure and Essentials

A Model Context Protocol (MCP) Server for My Website

```
import { computed, effect, signal } from 'alien-signals' ▶
```

```
const quantity = signal(0)
```

```
const price = signal(15)
```

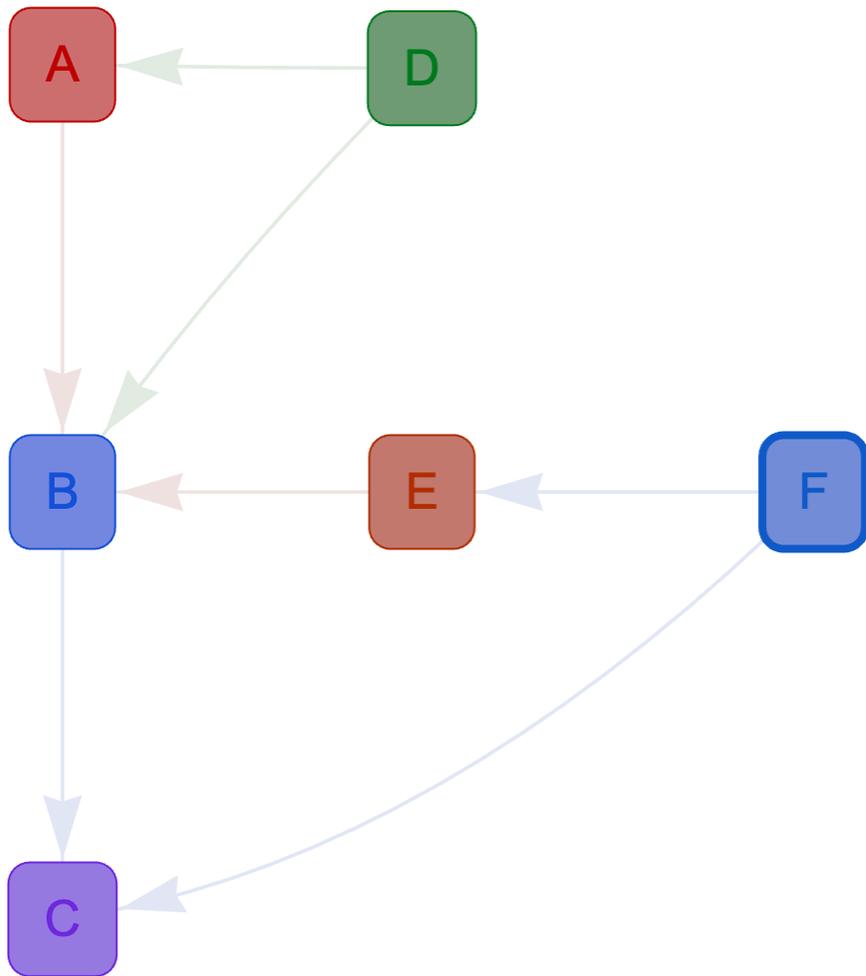
```
const total = computed(() => quantity() * price())
```

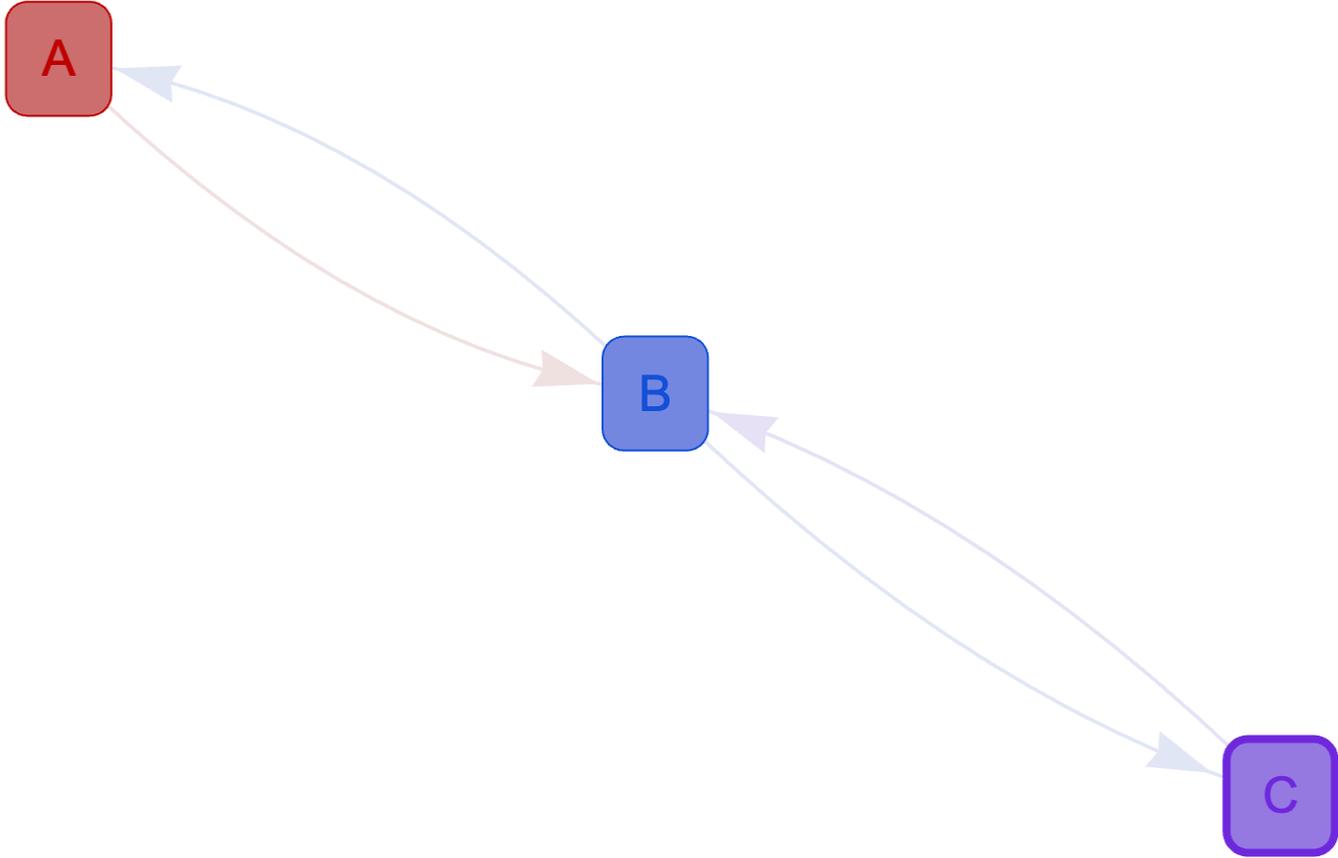
```
effect(() => {  
  console.log(`Total: ${total()}`)  
})
```

```
quantity(3)
```

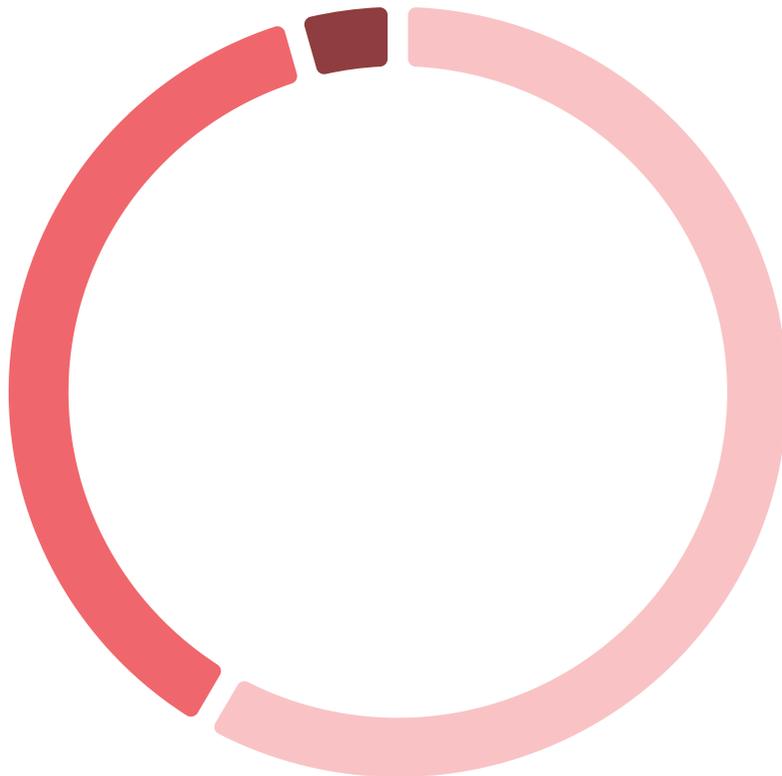
```
Total: 0
```

```
Total: 45
```





Vous suivez encore ?



● 93 - Oui, évidemment ● 59 - Oui, enfin je crois ● 6 - Non, là c'est trop 🤔

```
import { computed, effect, signal } from 'alien-signals'
```

```
const quantity = signal(0)
```

```
const price = signal(15)
```

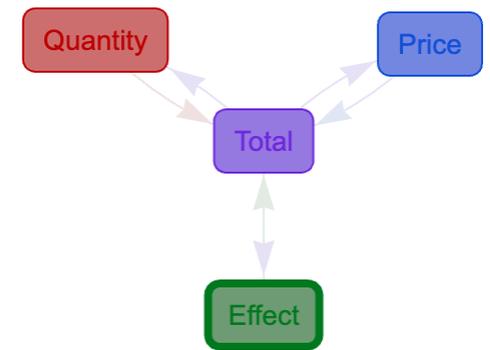
```
const total = computed(() => quantity() * price())
```

```
effect(() => {  
  console.log(`Total: ${total()}`)  
})
```

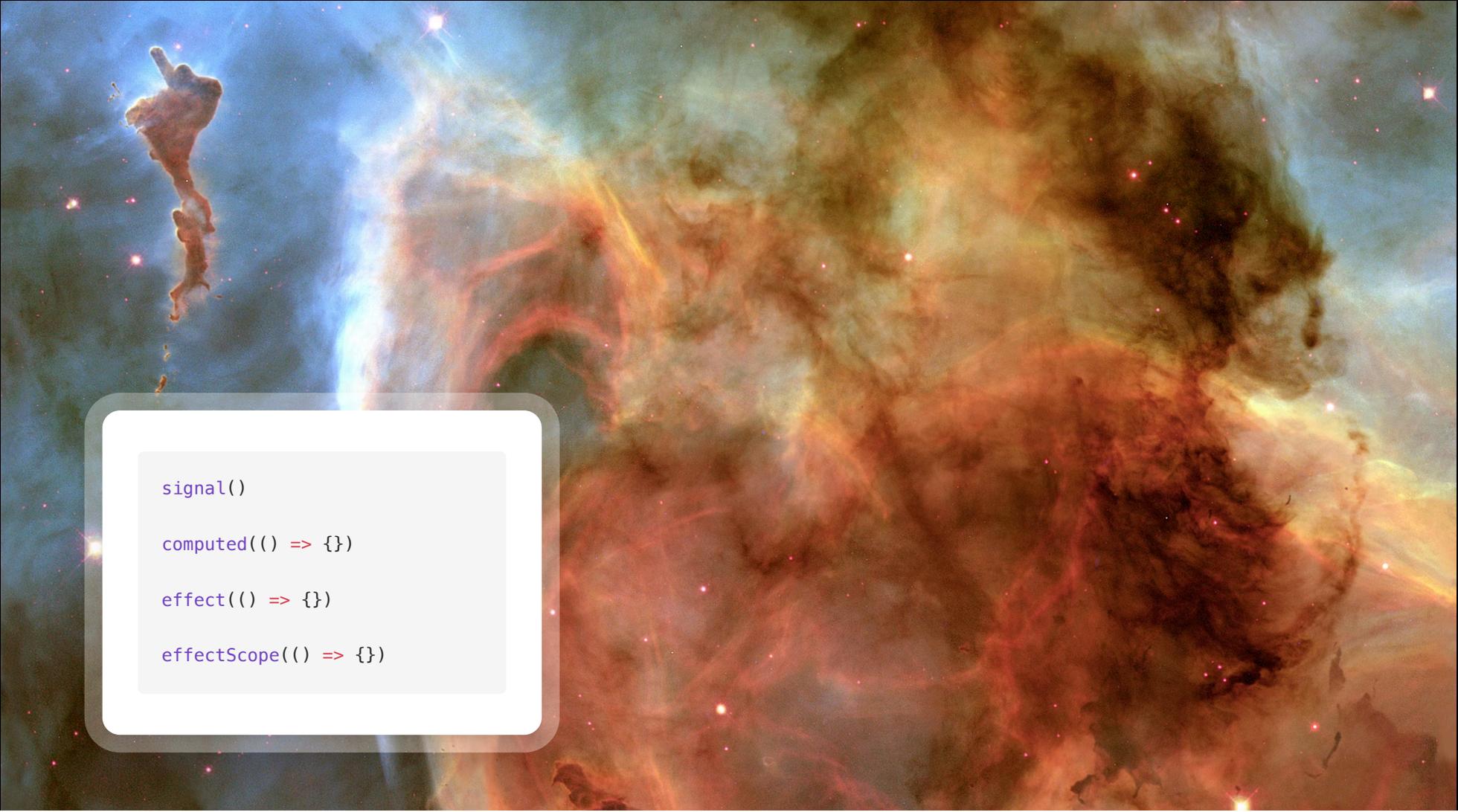
U[V]	0	0,4	0,6	0,8	0,9	1,0
I[mA]	0	-0,4	-0,76	-1,12	-1,5	-1,9
U[V]	0	-1	-2	-3	-4	-5
I[mA]	0	1,4	2,8	4,2	5,6	7,1
U[V]	0	1	2	3	4	5

$y) \begin{pmatrix} -t & y \\ z & -x \end{pmatrix} = \begin{pmatrix} yz - xt & 0 \\ 0 & yz - tx \end{pmatrix} =$

$yz - xt) I_0 = -(xt - yz) I_0$



$$-Q_{41} = vCT_1(1 - \epsilon^{1/2}) + vC_V T_1(\mathcal{K} - 1),$$
$$-Q_{34} = vC_V T_2(\mathcal{K} - 1) + vCT_4(1 - \epsilon^{1/2}),$$



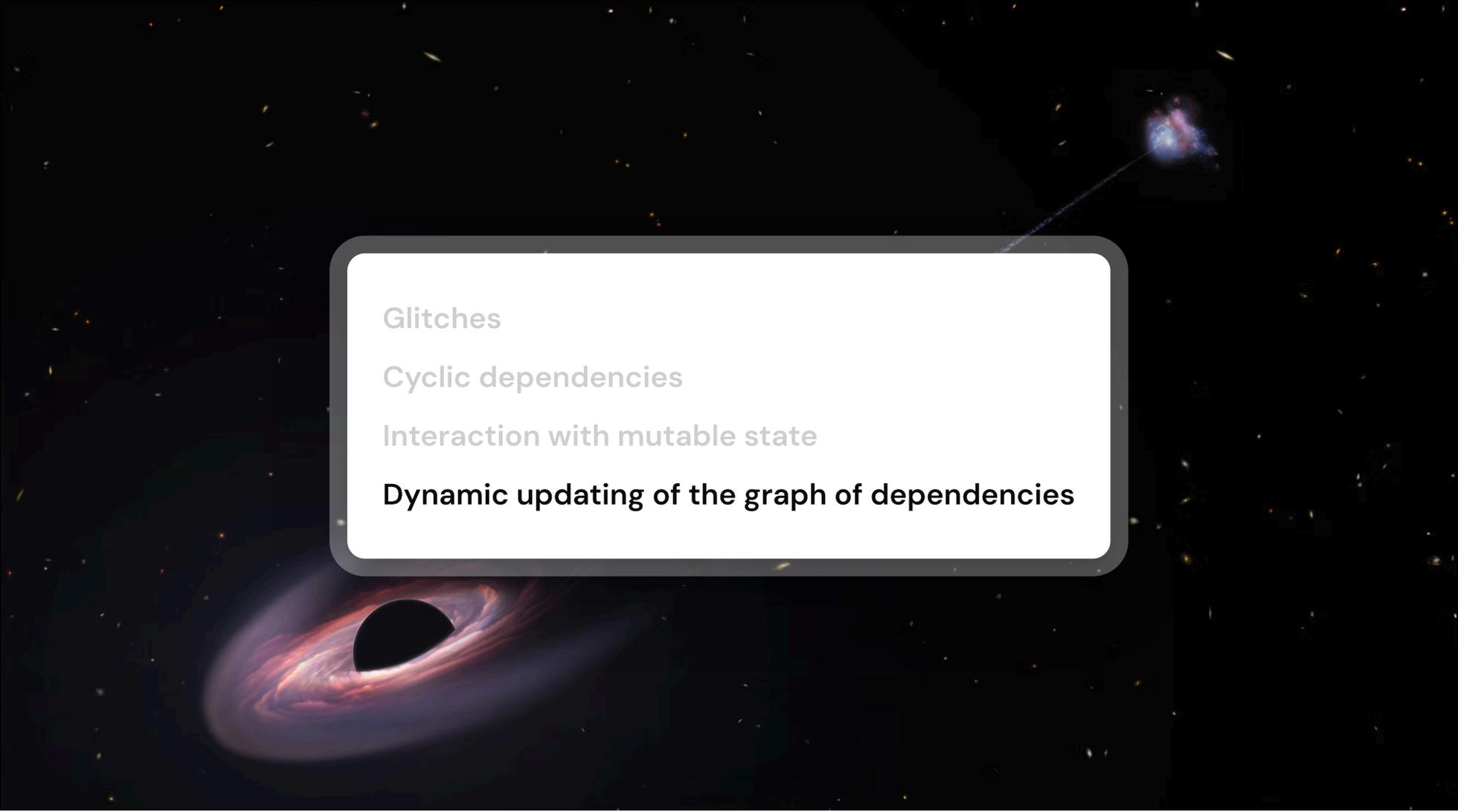
```
signal()
```

```
computed(() => {})
```

```
effect(() => {})
```

```
effectScope(() => {})
```

```
1 export function signal<T>(): WriteableSignal<T | undefined>
2 export function signal<T>(oldValue: T): WriteableSignal<T>
3 export function signal<T>(oldValue?: T): WriteableSignal<T | undefined> {
4   return signalGetterSetter.bind({
5     currentValue: oldValue,
6     subs: undefined,
7     subsTail: undefined,
8   }) as WriteableSignal<T | undefined>
9 }
```

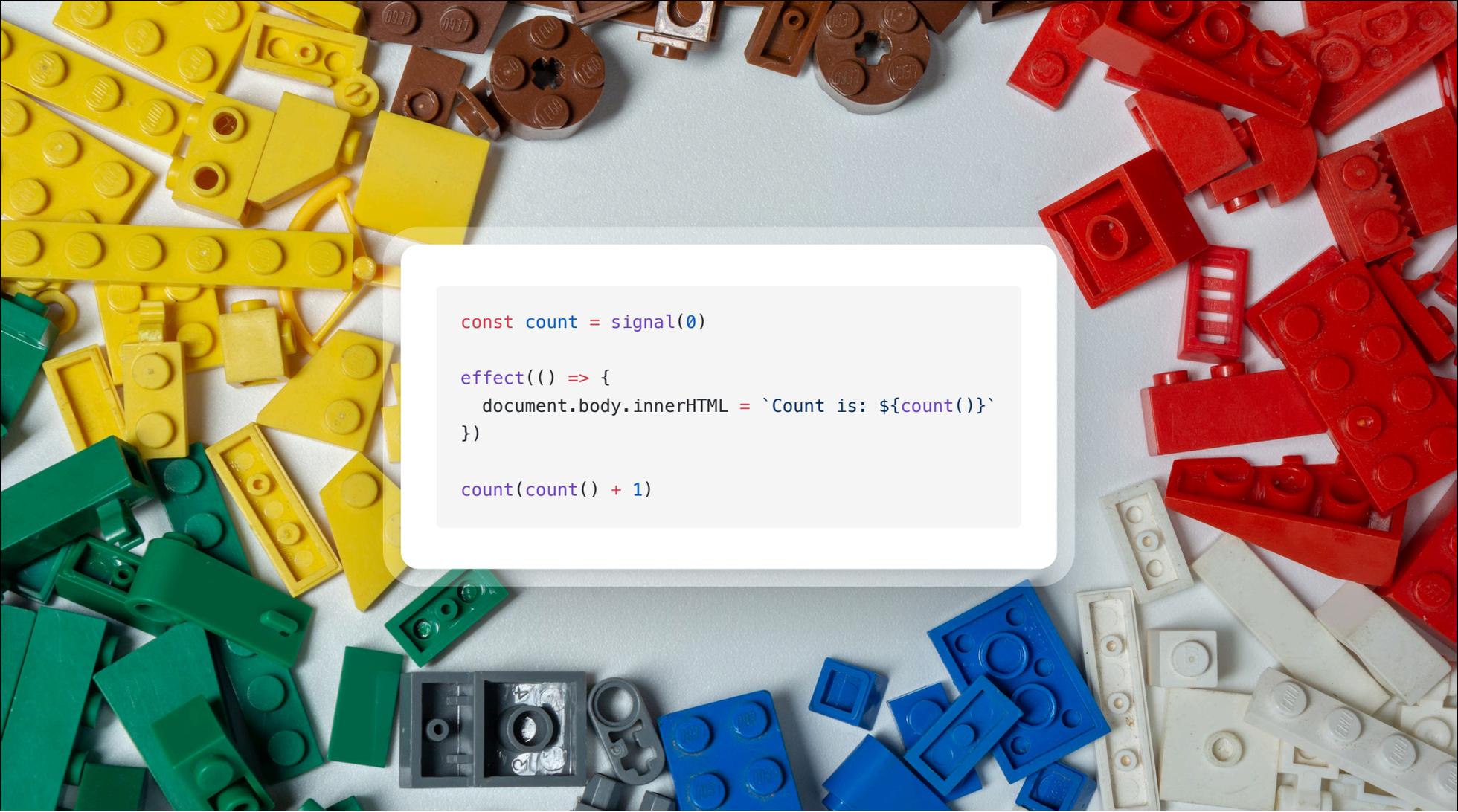
The background of the slide is a dark space filled with numerous small, distant stars. In the upper right corner, there is a bright, multi-colored galaxy with blue and red hues. In the lower left corner, there is a black hole with a glowing accretion disk in shades of purple and red. A white rounded rectangle with a dark border is centered on the slide, containing a list of topics.

Glitches

Cyclic dependencies

Interaction with mutable state

Dynamic updating of the graph of dependencies



```
const count = signal(0)
```

```
effect(() => {  
  document.body.innerHTML = `Count is: ${count()}`  
})
```

```
count(count() + 1)
```

```
instance.scope.on()  
const effect = (instance.effect = new ReactiveEffect(componentUpdateFn))  
instance.scope.off()
```

Looking for more?

- Explore the **Alien Signals** source code
- Watch **Reactivity across frameworks**
- Read **Unveiling the Magic: Exploring Reactivity Across Various Frameworks**
- Stay curious and keep learning!