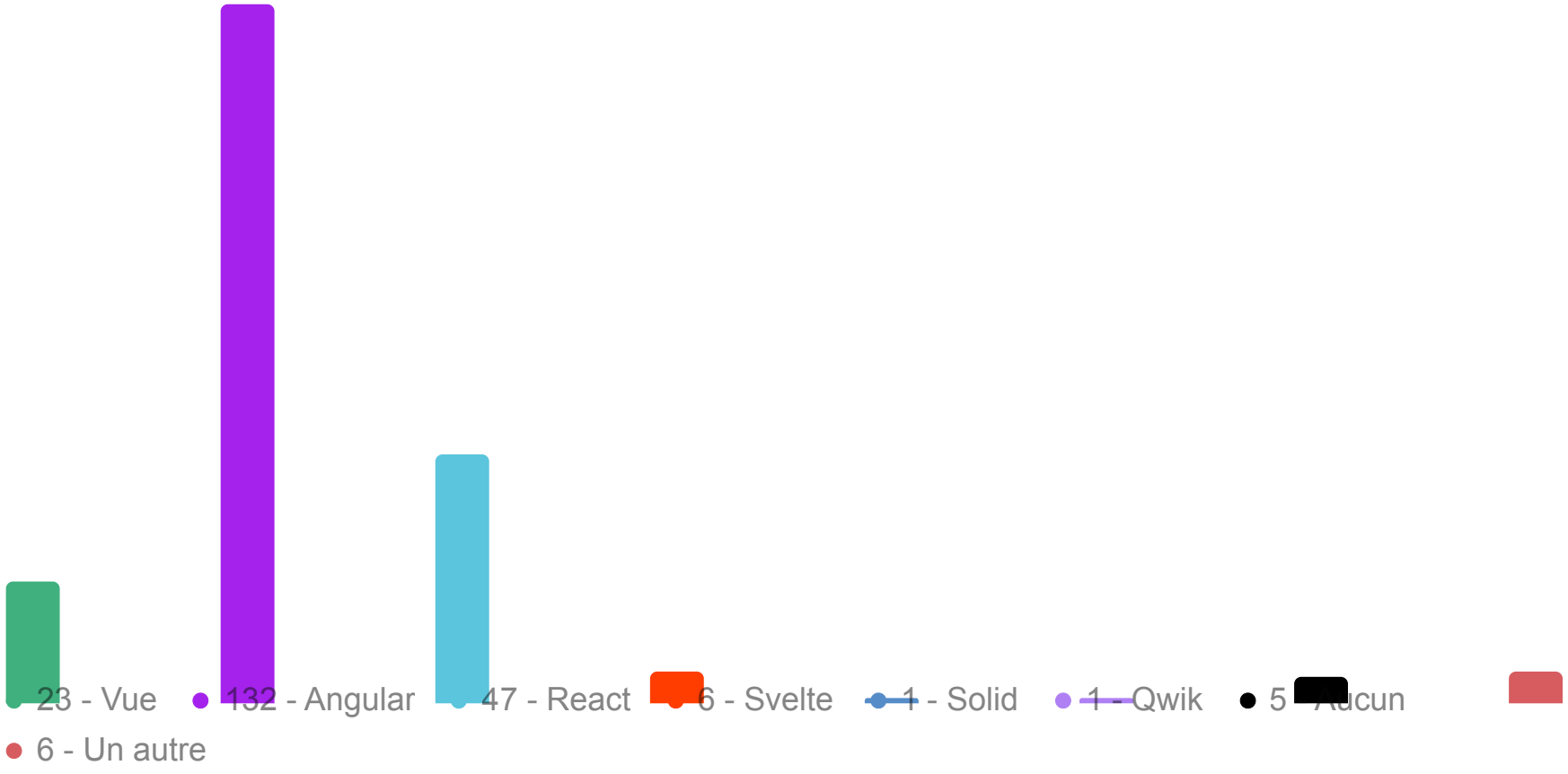
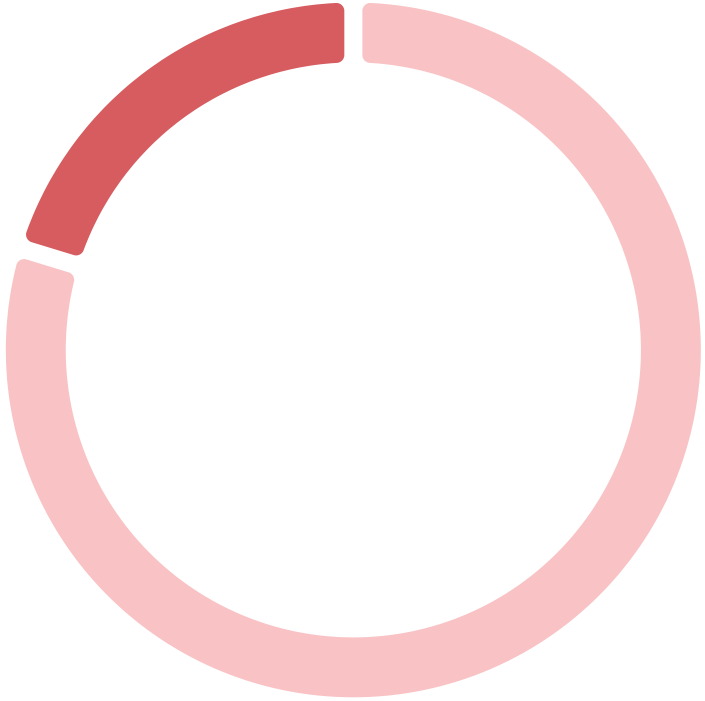
La réactivité et les signaux : démystifions la magie du frontend

Quel framework pour le frontend ?



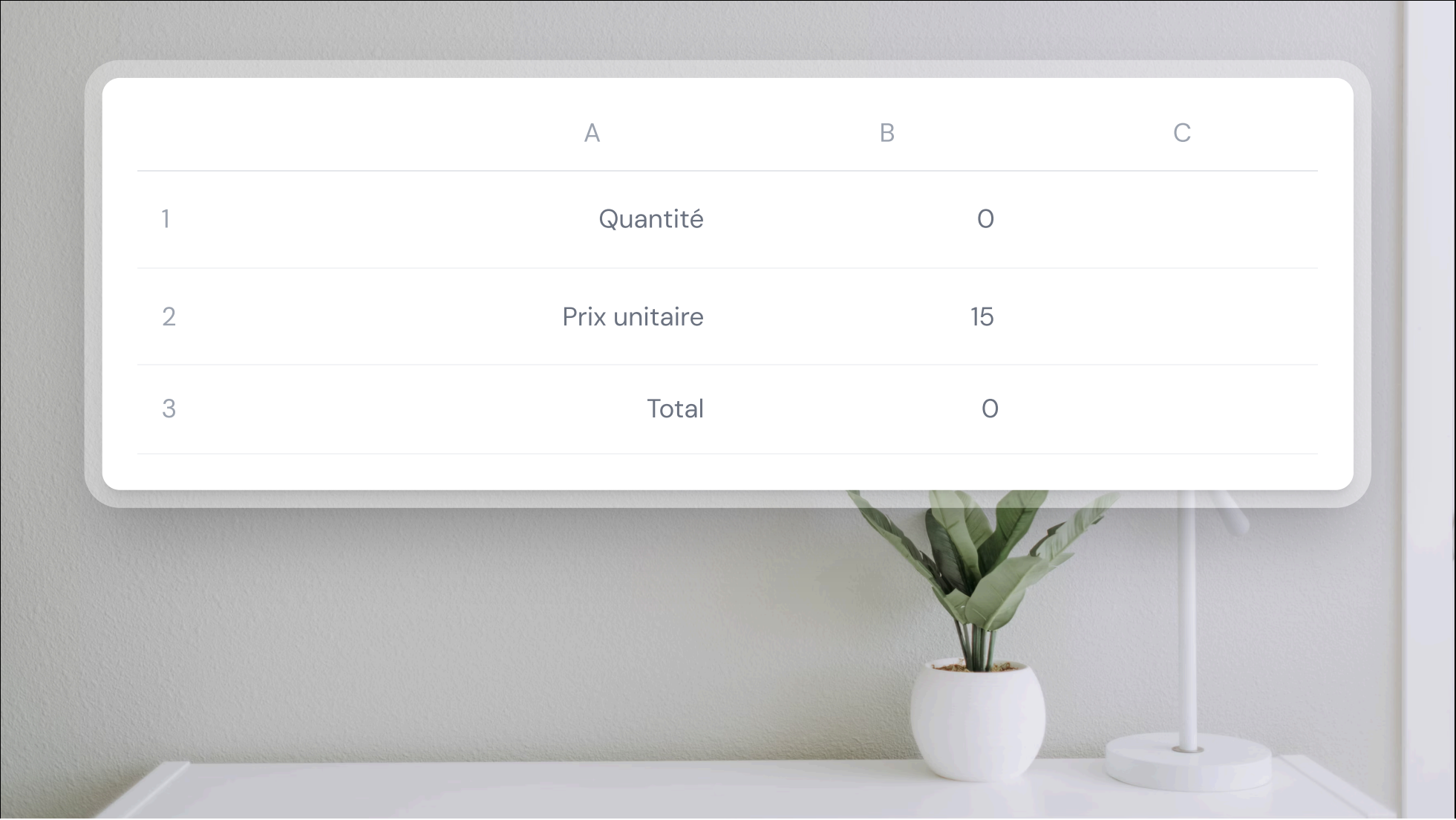
Qui s'est déjà questionné sur le fonctionnement profond de la réactivité de son framework ?



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

Laravel Vite Vue Nuxt

Ingénieur logiciel Avionique chez Maiaspace

Using Pinia Colada in Modals Without Spoiling the UX

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

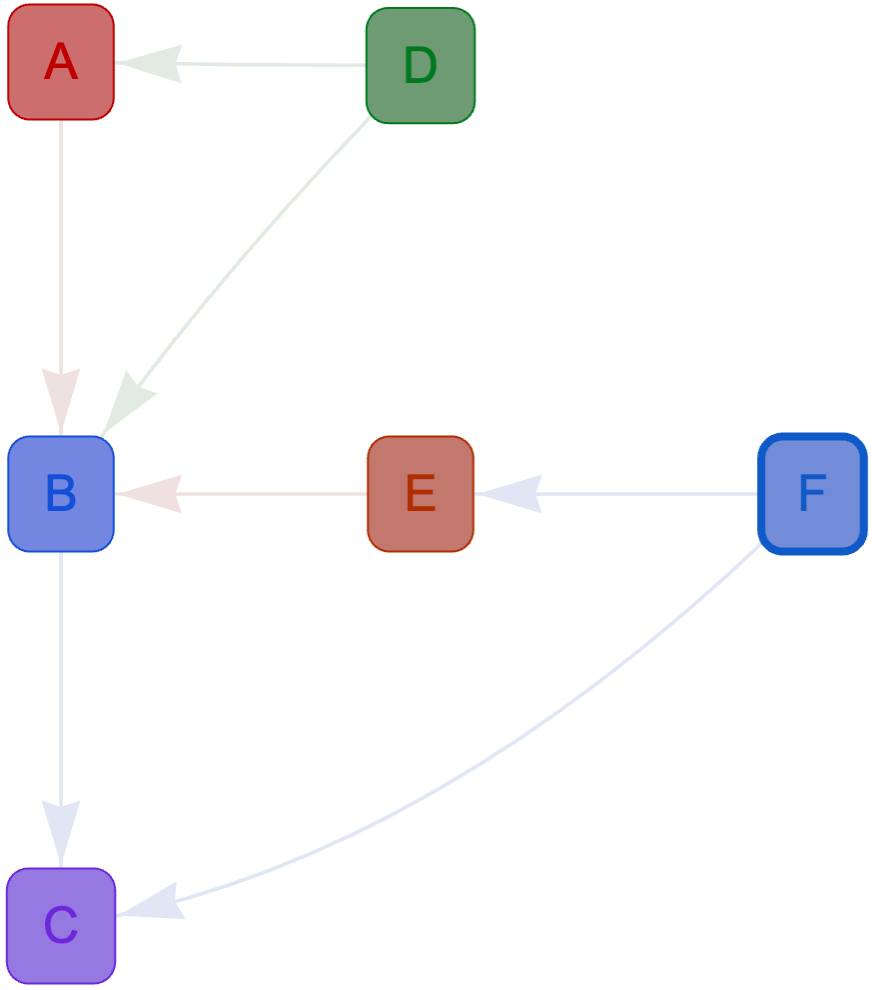
Setting Up Your Laravel Project Structure and Essentials

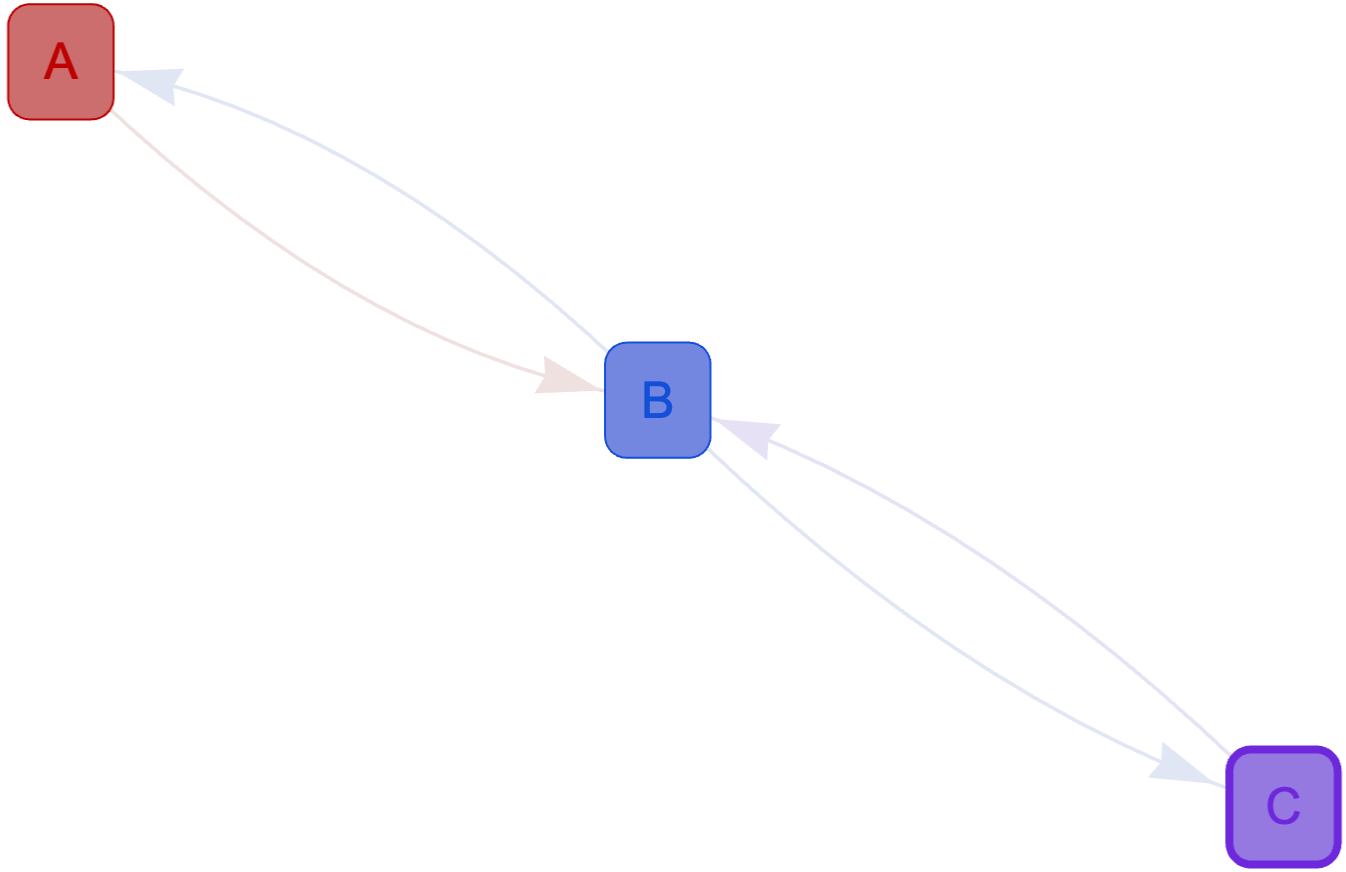
A Model Context Protocol (MCP) Server for My Website

soubiran.dev @soubiran_ @soubiran.dev Estéban S Barbapapazes Estéban's Open Source

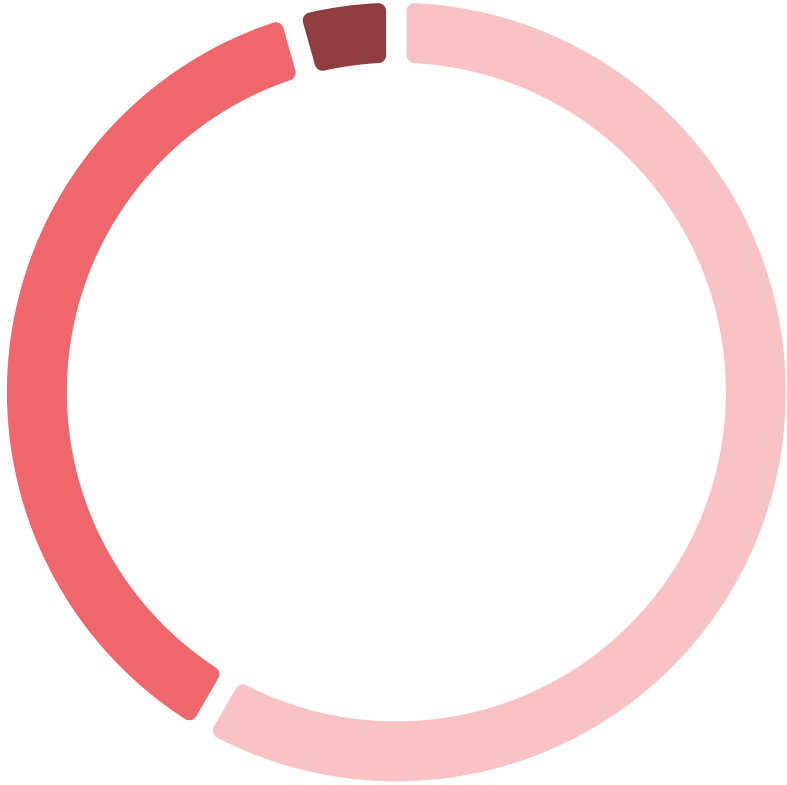
```
import { computed, effect, signal } from 'alien-signals'

const quantity = signal(0)
const price = signal(15)

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

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

quantity(3)
```

```
Total: 0
Total: 45
```


A
D
B
E
F
C

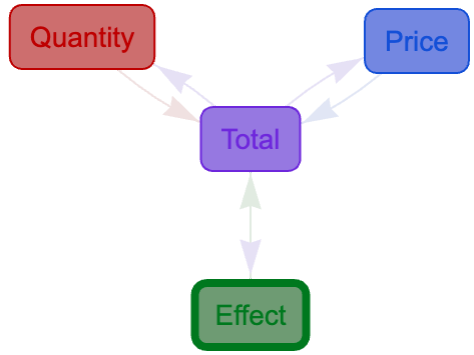

A
B
C

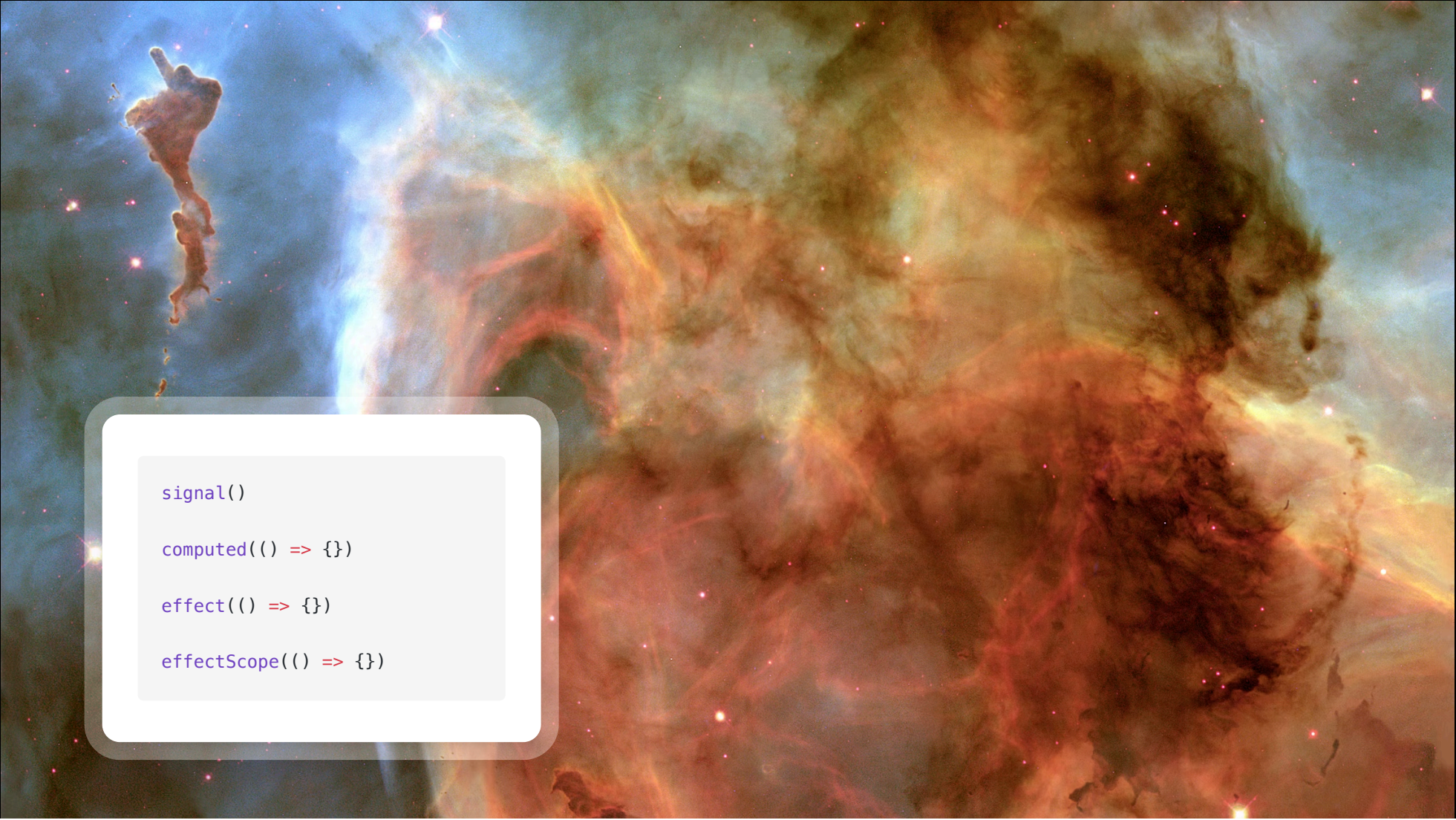
Vous suivez encore ?



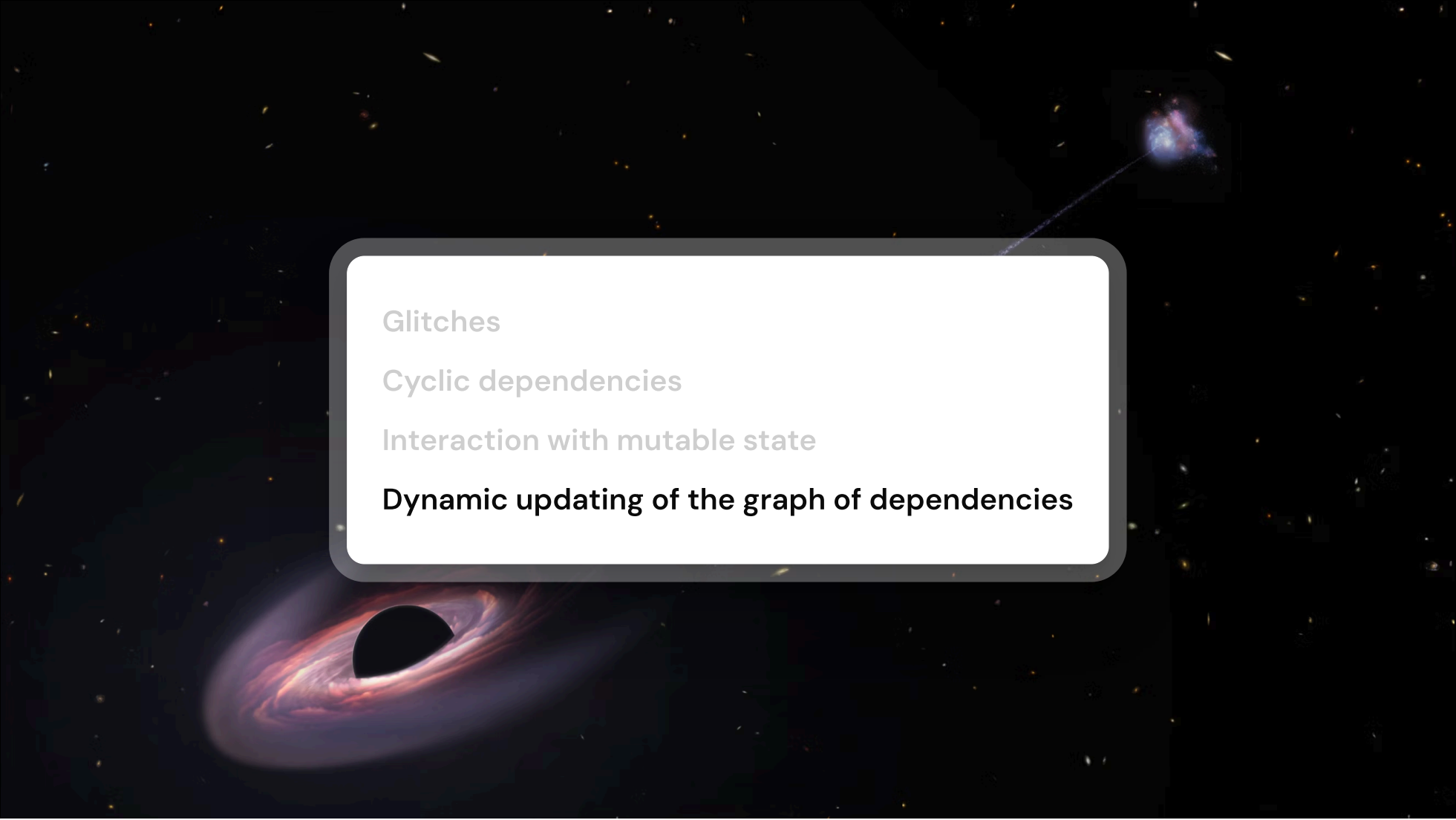
93 - Oui, évidemment
59 - Oui, enfin je crois
6 - Non, là c'est trop 🤯

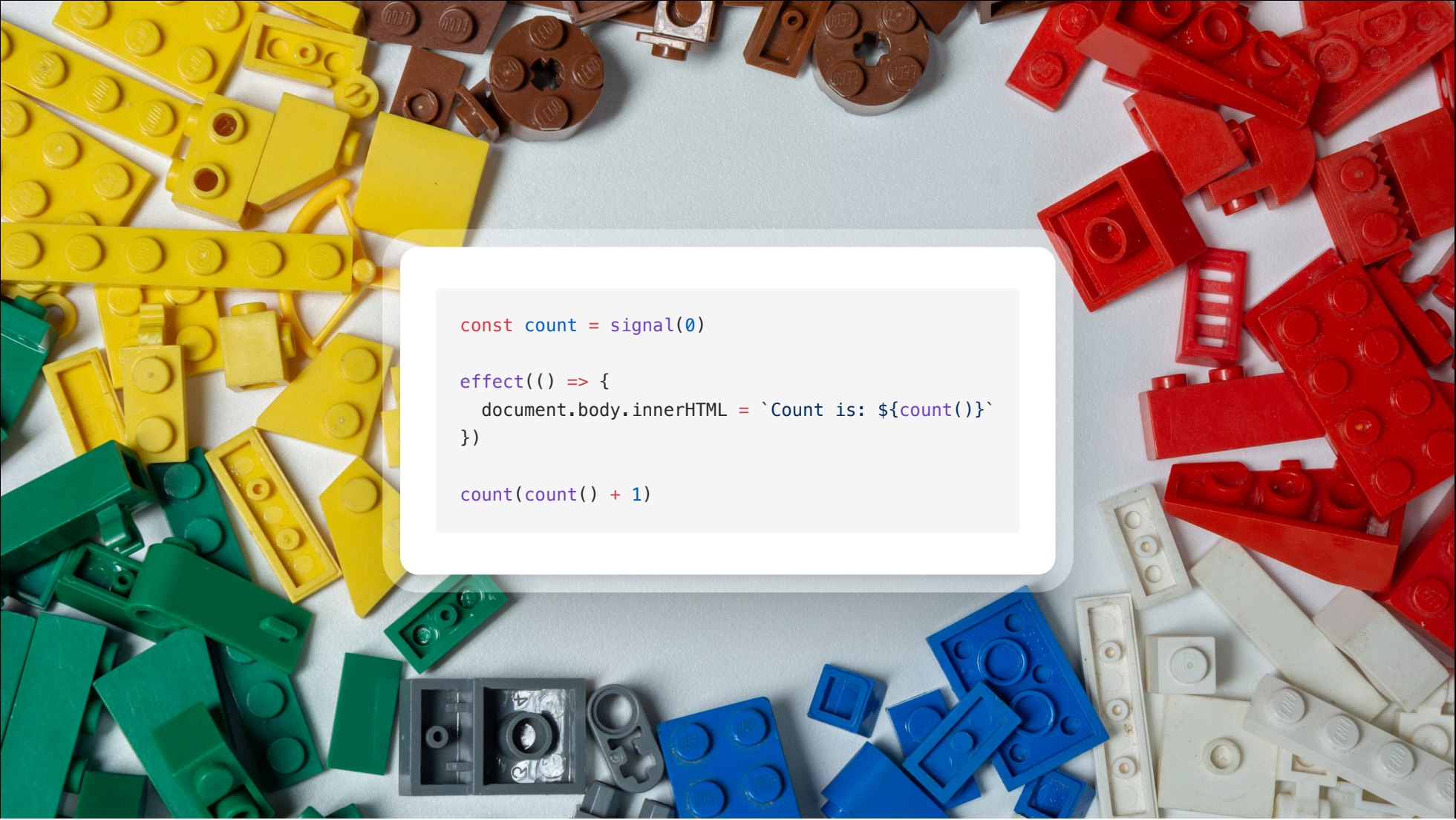
```
import { computed, effect, signal } from 'alien-signals'

const quantity = signal(0)
const price = signal(15)

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

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




signal()
computed(() => {})
effect(() => {})
effectScope(() => {})

```ts
export function signal<T>(): WriteableSignal<T | undefined>
export function signal<T>(oldValue: T): WriteableSignal<T>
export function signal<T>(oldValue?: T): WriteableSignal<T | undefined> {
  return signalGetterSetter.bind({
    currentValue: oldValue,
    subs: undefined,
    subsTail: undefined,
  }) as WriteableSignal<T | undefined>
}
```


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()

Estéban Soubiran
esteban@soubiran.dev

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!