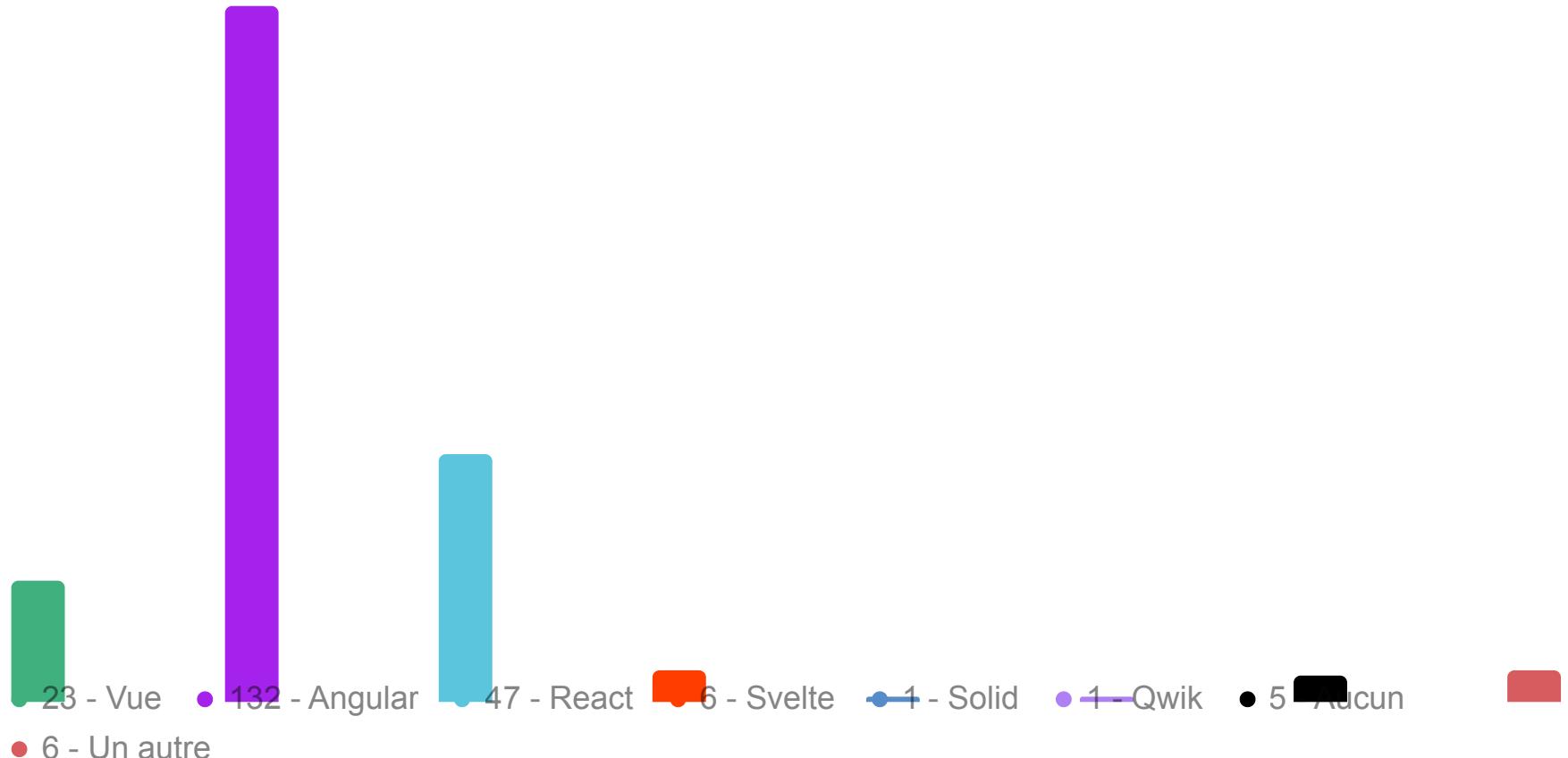


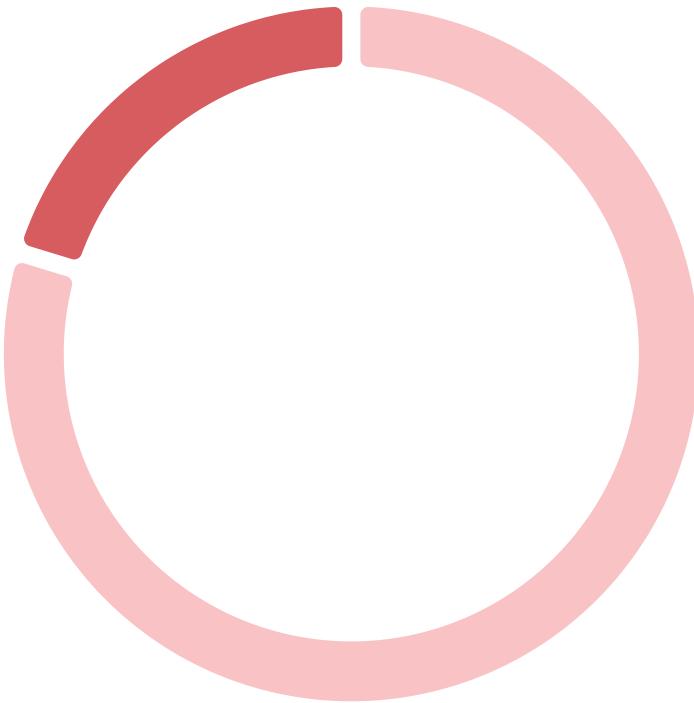
La réactivité et les signaux : démystifions la magie du frontend



Quel framework pour le frontend ?



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

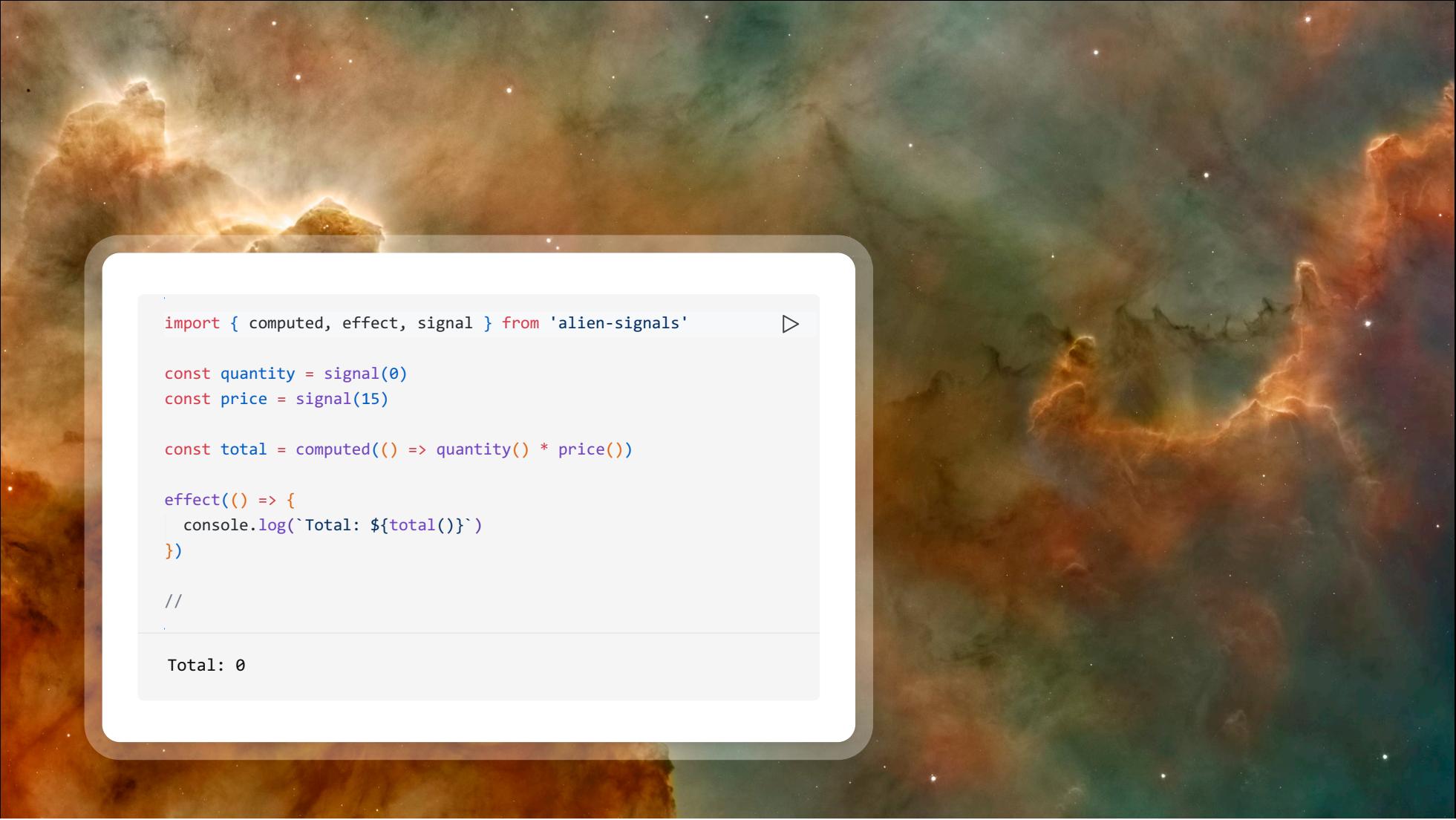


- 153 - Oui
- 38 - Non

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

A Journey to Craft Interactive UI Experiences with Dialogs

Fusion or the Art of Writing PHP Into a Vue SFC Components

Laravel and Vite: A Love Story Ruined with Cross-Origin

2024: Reflections on a Year of Change and What Comes Next



soubiran.dev



@soubiran_



@soubiran.dev



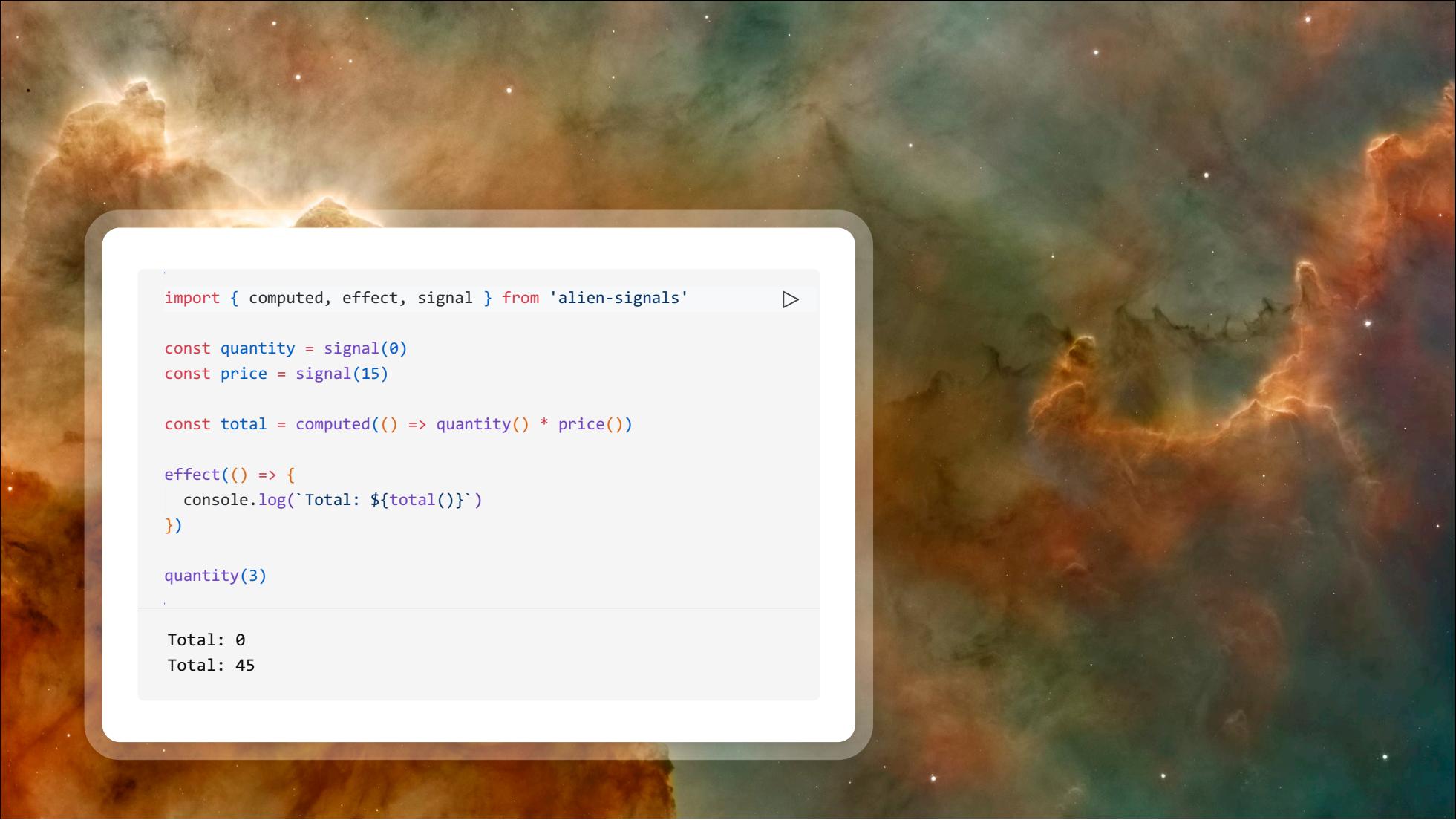
Estéban S



Barbapapazes



Estéban's Open Source



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

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

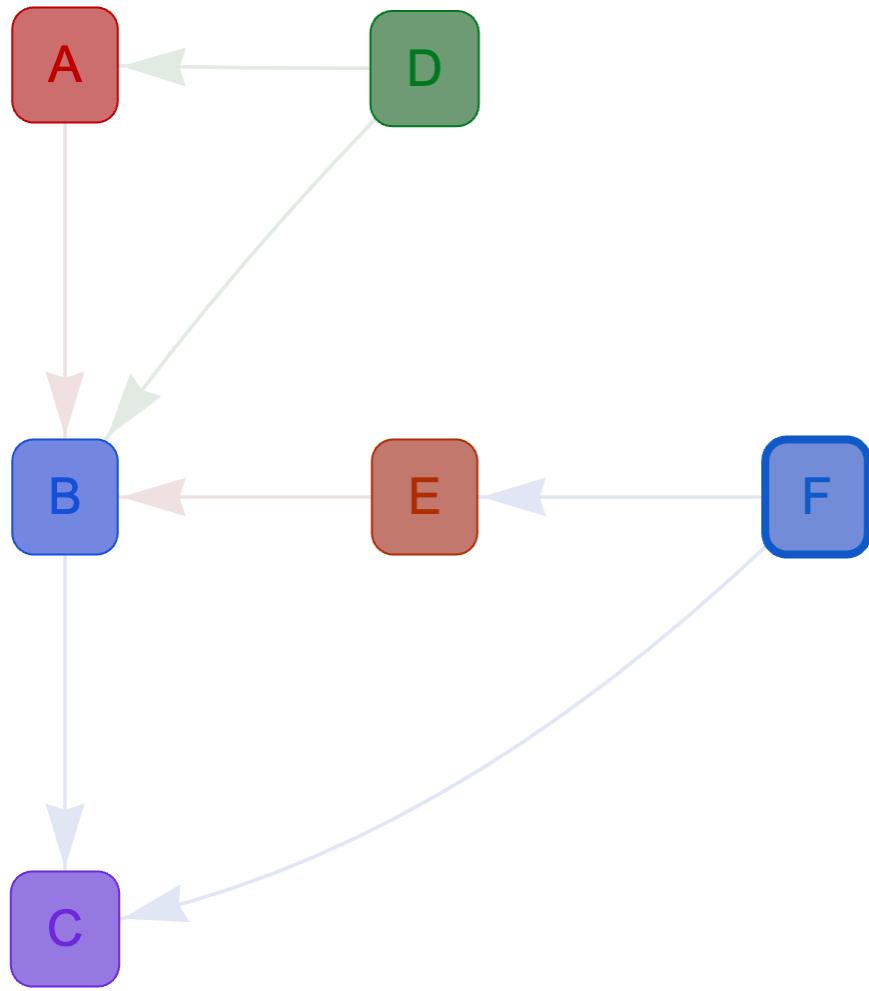
const total = computed(() => quantity() * price())

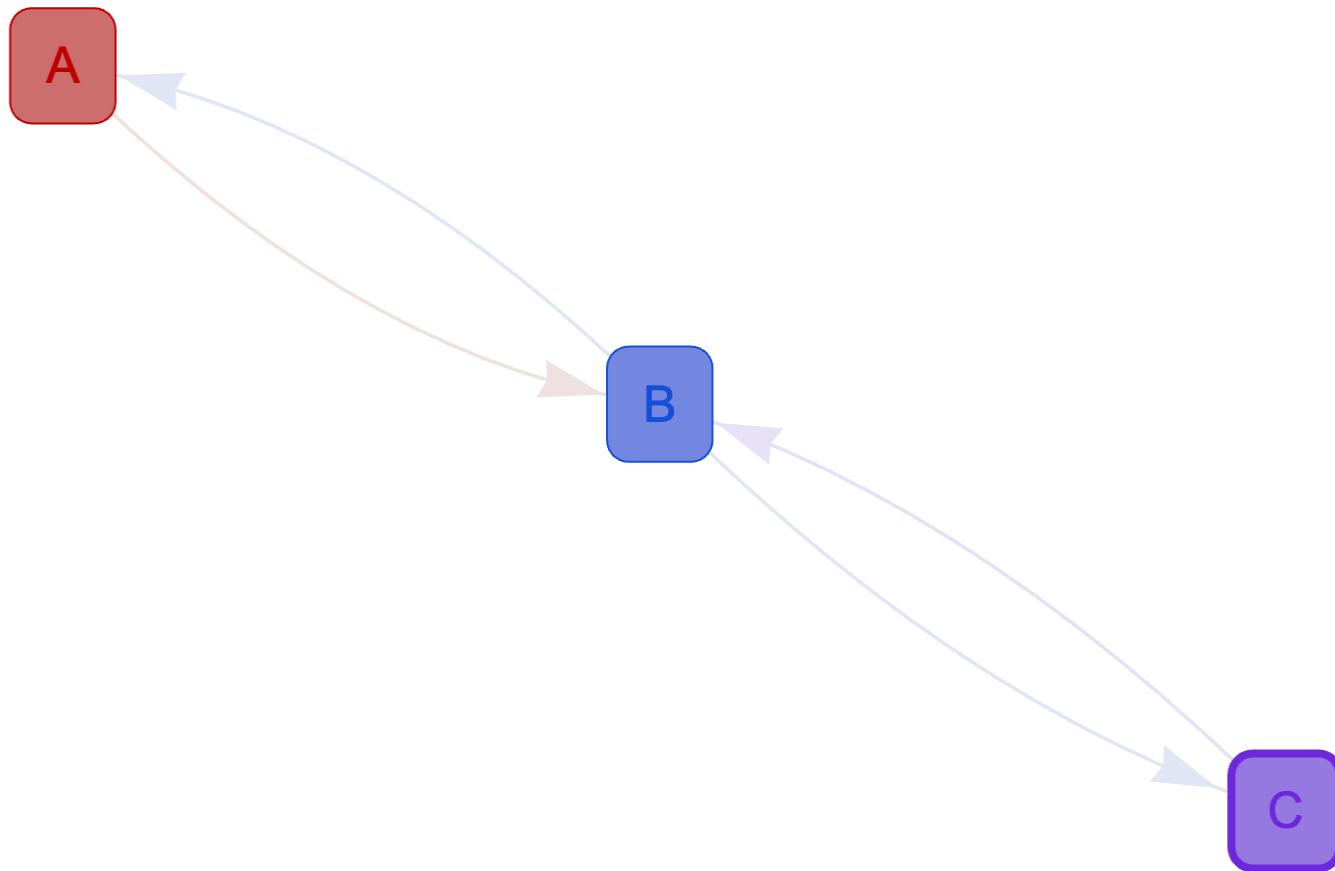
effect(() => {
  console.log(`Total: ${total()}`)
})

quantity(3)

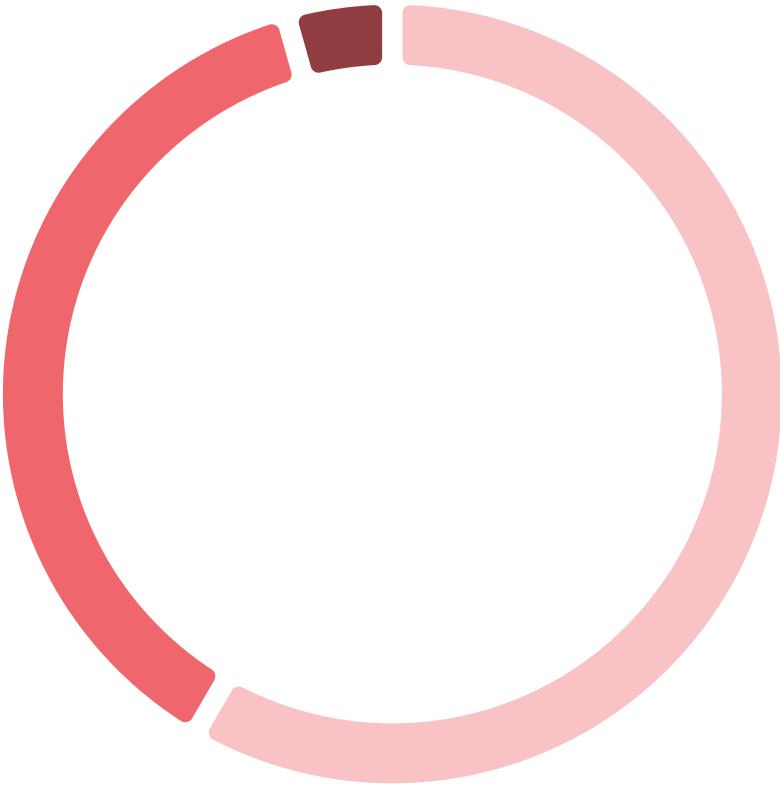

```

Total: 0
Total: 45

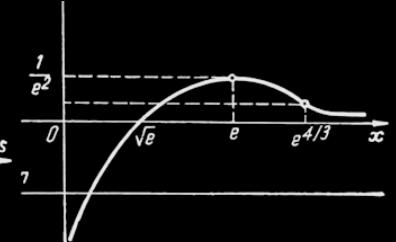
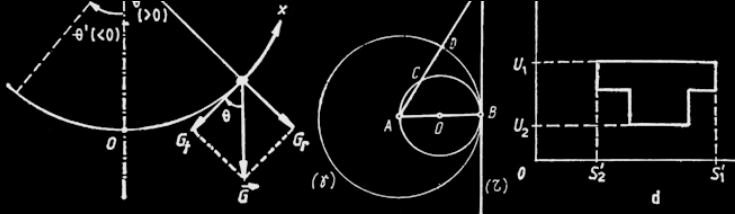




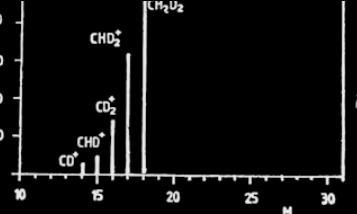
Vous suivez encore ? 🎅



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



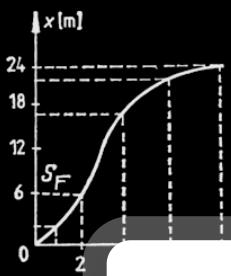
$$E_c = E_{c_{\max}} \Rightarrow \cos^2\left(3t_c + \frac{\pi}{3}\right) = 1 \Rightarrow \cos\left(3t_p + \frac{\pi}{3}\right) = \pm 1 = \cos(n\pi) \Rightarrow t_p = \frac{\pi}{3} \left(n - \frac{1}{3}\right)$$



$$\sqrt{\frac{k}{m}} = \sqrt{\frac{4\pi m_1 K_p}{3m_1}} = \sqrt{\frac{4\pi K_p}{3}}$$

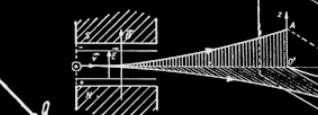
$$\omega = \sqrt{\frac{g_0}{B_0}},$$

$$= \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{R_0}{\gamma}} = 5,03 \cdot 10^3 \text{ s.}$$



$$\frac{1 - \left(-\frac{1}{n+2}\right)^{n+1}}{1 + \frac{1}{n+2}} + \frac{1}{n+1} \cdot \frac{1 - \left(-\frac{1}{n+1}\right)^{n+1}}{1 + \frac{1}{n+1}} = \int_{-a}^0 x^2 e^{ax} dx = \frac{1}{a} (x^2 e^{ax}) \Big|_{-a}^0 - \frac{2}{a} \int_{-a}^0 e^{ax} dx$$

$$-\frac{\frac{1}{n+1}^{n+1}}{n+2} - \frac{1 - \left(-\frac{1}{n+2}\right)^{n+1}}{n+3} = + \frac{2}{a^2} \left[\frac{1}{a} (e^{ax}) \Big|_{-a}^0 \right] = -ae^{-a^2} - \frac{2}{a} e^{-a^2}$$



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

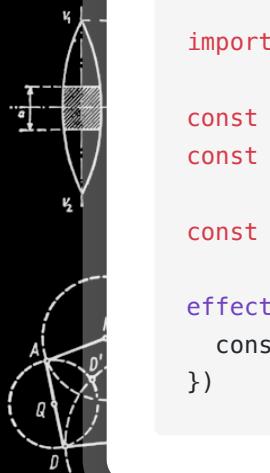
const quantity = signal(0)
const price = signal(15)

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

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

Quantity

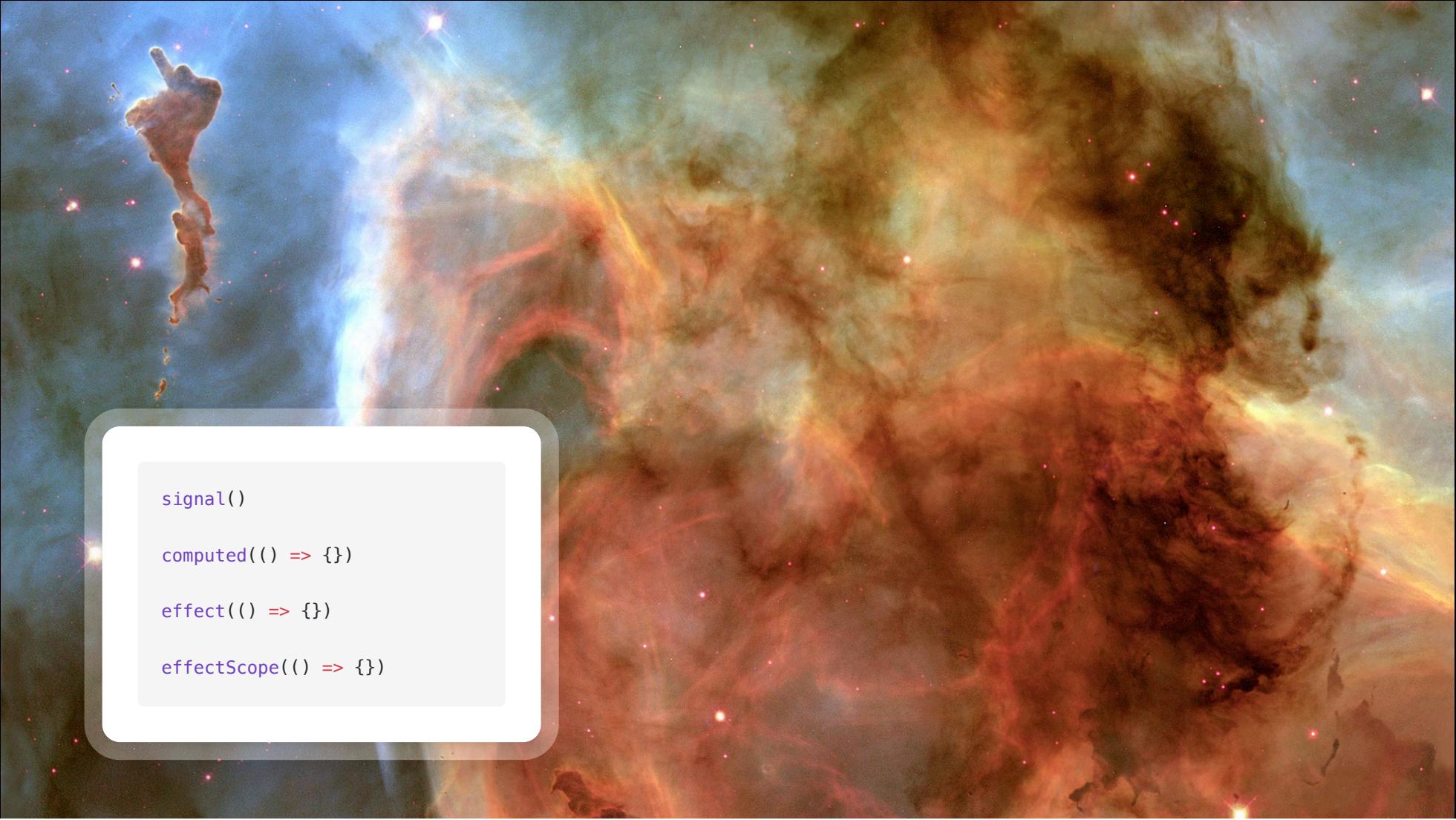
Price



$U[V]$	0	0,4	0,6	0,8	0,9
$I[mA]$	0	-0,4	-0,76	-1,12	-1,5
$U[V]$	0	-1	-2	-3	-4
$I[mA]$	0	1,4	2,8	4,2	5,6
$\frac{U}{I} [V/A]$	0	-1	-2	-3	-4

$$\begin{array}{c|cc} 0 & y & -t \\ \hline 4.9 & t \\ 5 & z \\ 1 & -x \end{array} = \begin{pmatrix} yz - xt & 0 \\ 0 & yz - tx \end{pmatrix} =$$

$$\begin{aligned} Q_{41} &= vCT_1(1 - \varepsilon^{1/2}) + vC_V T_1(\mathcal{H} - 1), \\ Q_{34} &= vC_V T_2(\mathcal{H} - 1) + vCT_4(1 - \varepsilon^{1/2}), \end{aligned}$$

The background of the image is a vibrant, multi-colored nebula, likely the Eagle Nebula, showing intricate patterns of gas and dust in shades of blue, green, yellow, and red. It serves as a dramatic and visually appealing backdrop for the text overlay.

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

The background of the slide features a stunning image of a nebula, likely the Horsehead Nebula, with its characteristic dark silhouette and surrounding colorful gas clouds in shades of blue, orange, and yellow. A white rectangular box containing the code is centered in front of this celestial backdrop.

```
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 }
```

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 [Align Signals](#) source code
- Watch [Reactivity across frameworks](#)
- Read [Unveiling the Magic: Exploring Reactivity Across Various Frameworks](#)
- Stay curious and keep learning!