



Adieu RxJS ! Vive les
Signals ! Oh wait...

Anthony Pena



Anthony Pena

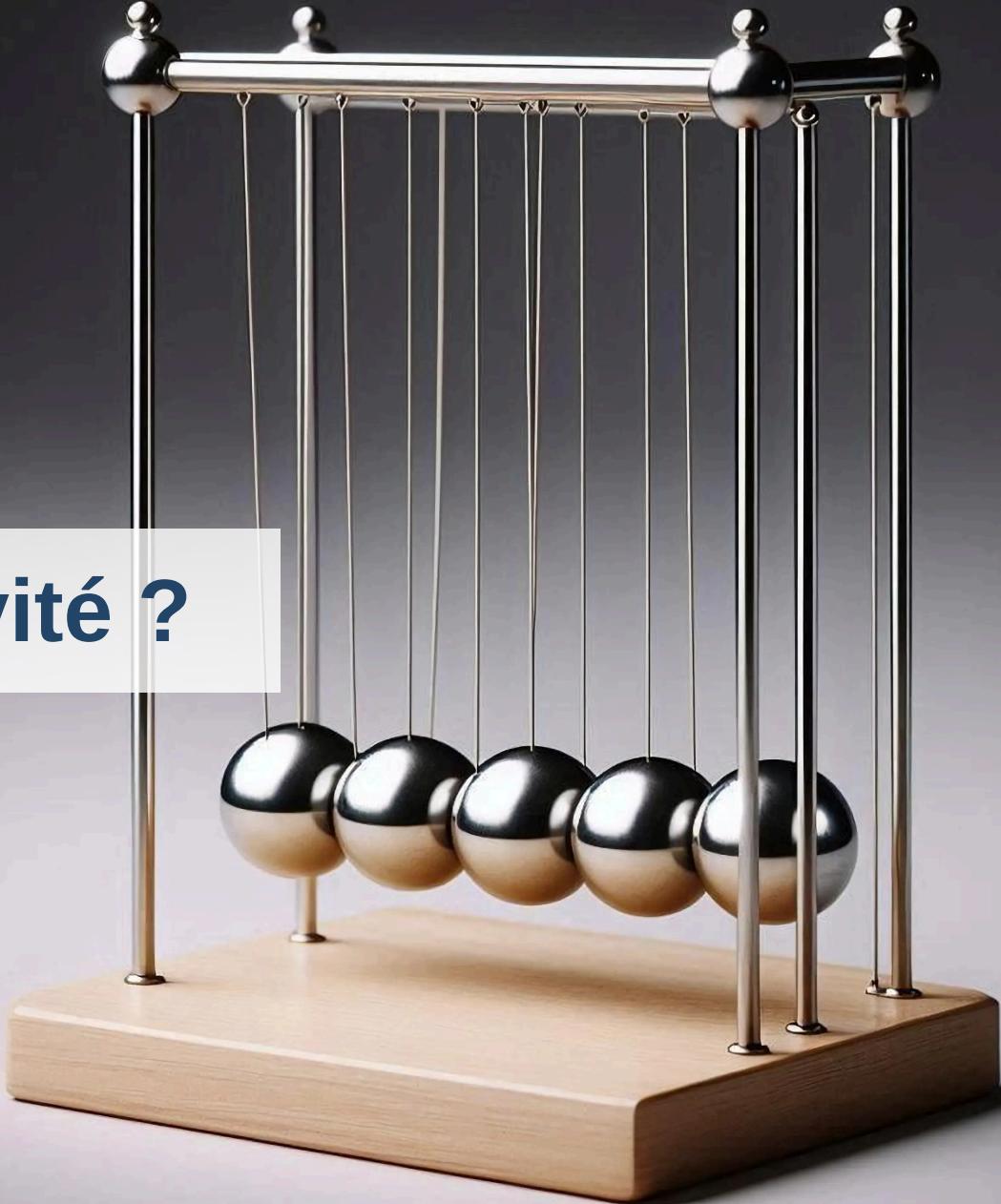
Développeur Web Fullstack @ **[sf=ir]**



A detailed painting of a caveman in a dense forest. He is wearing a fur vest and is focused on chiseling a large rock with a wooden hammer. In the background, another caveman sits on the ground, and a third stands behind him. The scene is set in a lush, green environment with tall trees and ferns.

Avant les frameworks

La Réactivité ?



"A declarative programming model for updating based on changes to state."

Kristen / pzuraq

<https://www.pzuraq.com/blog/what-is-reactivity>



Angular.JS et ses watchers



Angular 2 et Zone.js

```
@Component({
  template: `
    <p>{{ text }}</p>
  `,
})
export class PlaygroundComponent {
  text = "";

  ngOnInit() {
    setInterval(() => this.text += '!', 1_000)
  }
}
```

```
@Component({
  template: `
    <input (change)="setText($event)"/>
    <p>{{ text }}</p>
  `,
})
export class PlaygroundComponent {
  text = "";

  setText(event: Event) {
    this.text = (event.target as HTMLInputElement).value;
  }
}
```

```
@Component({
  template: `
    <input [(ngModel)]="text"/>
    <p>{{ text }}</p>
  `
})
export class PlaygroundComponent {
  text = "";

  ngOnInit() {
    setInterval(() => this.text += '!', 1_000)
  }
}
```

```
@Component({
  selector: 'app-root',
  standalone: true,
  template: `
    <p>{{ text }}</p>
  `,
})
export class PlaygroundComponent {
  text = "waiting...";

  ngOnInit() {
    this.asyncHello().then(text => this.text = text);
  }

  private asyncHello(): Promise<string> {
    // ...
  }
}
```

```
@Component({
  template: `
    <p>{{ text | async }}</p>
  `,
})
export class PlaygroundComponent {
  text = this.asyncHello();

  private asyncHello(): Promise<string> {
    // ...
  }
}
```

C'est cool tout ça non ?

Oui mais Zone.js

Angular 17 et les Signals

CODE SLIDE : reprendre la demo edition de variable mais en Signal

**CODE SLIDE : reprendre la demo edition de
champ de texte mais en Signal**

**CODE SLIDE : reprendre la demo promesse
mais en Signal**

Signals <3

Bientôt un standard

<https://github.com/tc39/proposal-signals>

Zone.js c'est fini ?

CODE SLIDE : demo Signal input()

CODE SLIDE : demo Signal output()

CODE SLIDE : demo Signal viewchild()

Oups j'ai oublié RxJS dans tous ça 🙄 (non)

Mais au fait... C'est quoi RxJS ?

**RxJS = ReactiveX for JavaScript = librairie
pour traiter des flux d'évènements**

Parlons Observable

**CODE SLIDE : demo Signal + Observable
toSignal() toObservable()**

Montrer les limitations des Signals

Observables

Composable functions with guarantees.

Use observables for cancellation and event coordination.

- + Generally stateless
process event and clean up

- + Multiple values
a collection of events or values over time

- + Push N values

- + Convert to Signal? Yes

<https://twitter.com/BenLesh/status/1775207971410039230>

Signals

A dependency graph of values and logic around notification, invalidation, computation, and memoization.

Use signals for state management.

- + Maintain state
retain things in memory

- + Single value
a value that changes over time

- + Read 1 value

- + Convert to Observable? Avoidable
the event that set the signal can also notify the observable



CityJS
Cloudyleet
co

**Prendre le cas du HttpClient et des
interceptors qui sont entièrement basés sur
RxJS**

**Montrer par l'exemple de quand utiliser les
Signals et quand utiliser RxJS**

Les Signals c'est pour gérer les états et la réactivité dans les composants

RxJS est là pour gérer tous vos flux de données



Anthony Pena

Développeur Web Fullstack @

[sf≡ir]

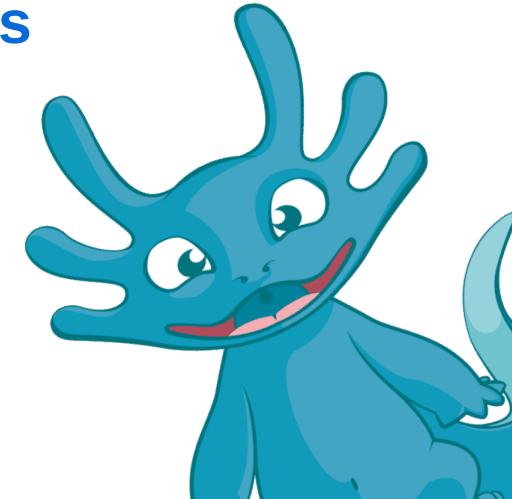
 @_Anthony_Pena_

 @kuroidoruido

 @penaanthon

<https://k49.fr.nf>

<https://github.com/kuroidoruido/talks>



Thank you!