

# GUI Frameworks, Libraries & toolkits



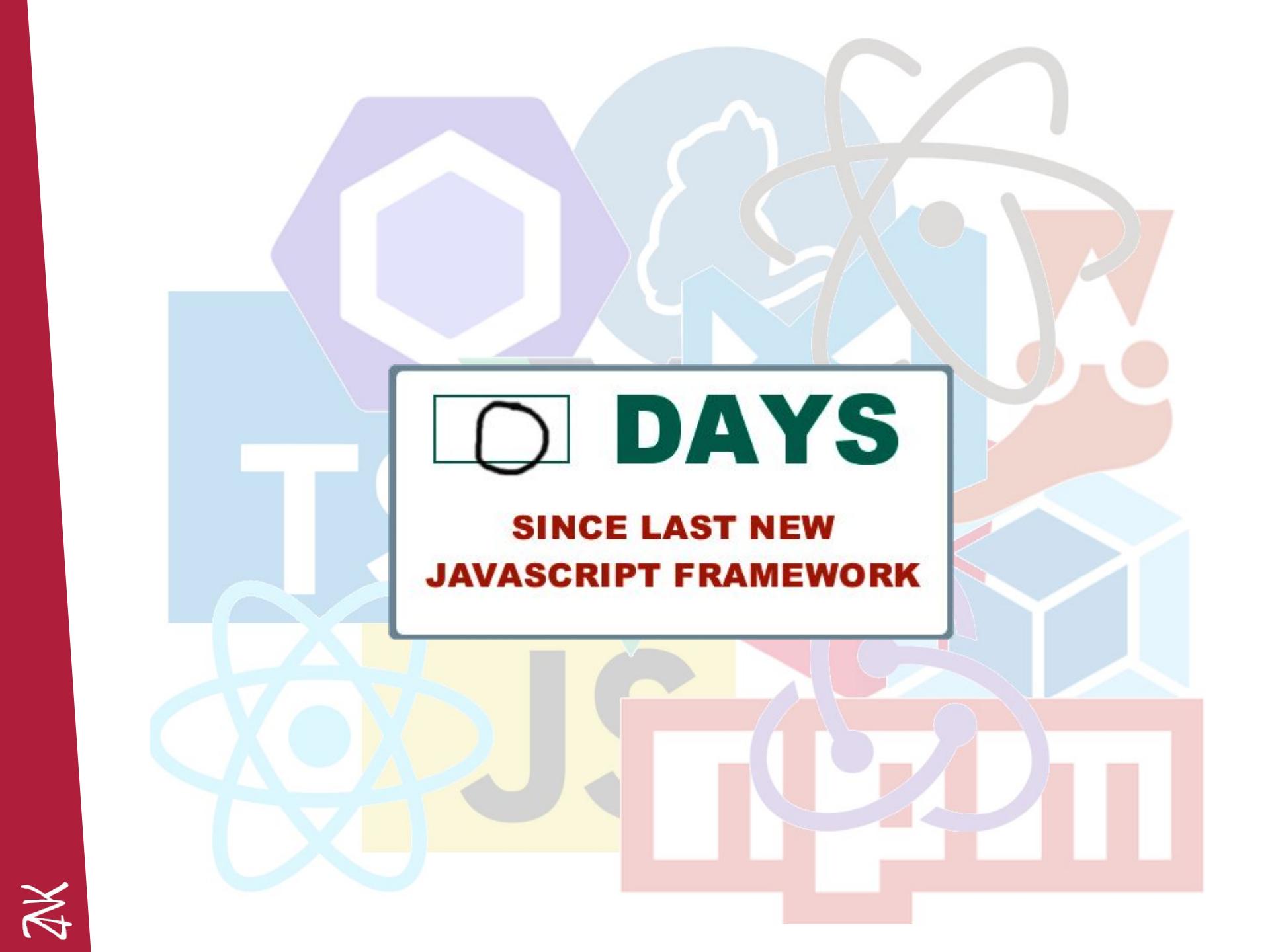
# **Fullstack Dev & Formateur Zenika**

## Fondateur du GDG Nantes & DevFest

[benjamin.petetot@gmail.com](mailto:benjamin.petetot@gmail.com)  
[@bpetetot](https://twitter.com/bpetetot)

# SOMMAIRE

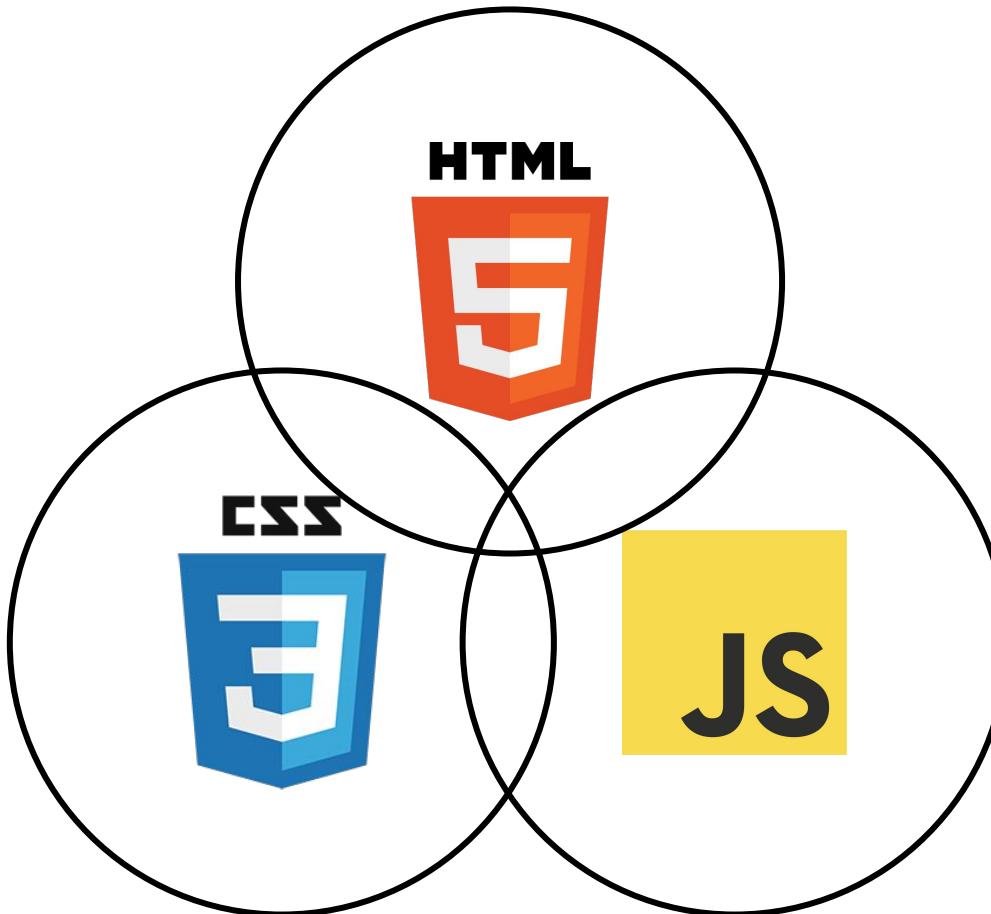
1. Évolution des langages web
2. Outils de développement
3. Approche orientée composants
4. Librairies & Frameworks
  - a. Angular 
  - b. React 
  - c. Vuejs 
5. On code...
6. Testing
7. Mobile



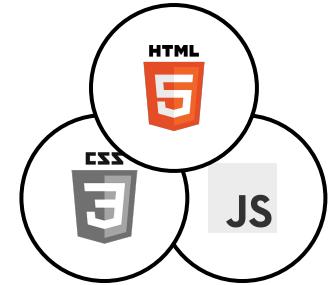
**0** DAYS  
**SINCE LAST NEW  
JAVASCRIPT FRAMEWORK**

# Évolution des langages web

# ÉVOLUTION DES LANGAGES WEB



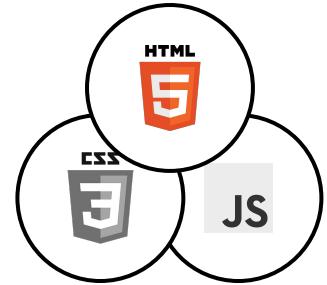
# ÉVOLUTION DES LANGAGES WEB



## HTML

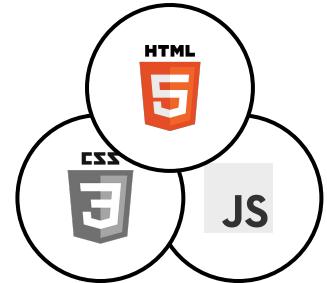


# ÉVOLUTION DES LANGAGES WEB



SEMANTIQUES

# ÉVOLUTION DES LANGAGES WEB

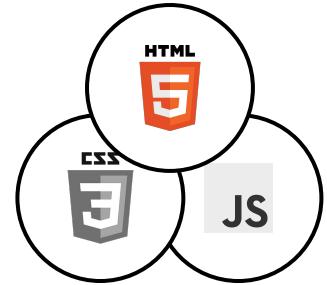


SEMANTIQUES



CONNECTIVITÉ

# ÉVOLUTION DES LANGAGES WEB



SEMANTIQUES

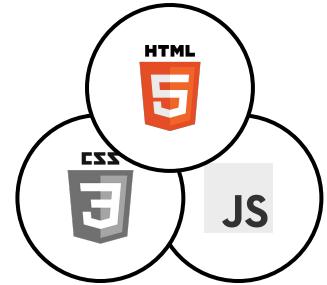


CONNECTIVITÉ



PERFORMANCE & INTEGRATION

# ÉVOLUTION DES LANGAGES WEB



SEMANTIQUES



CONNECTIVITÉ

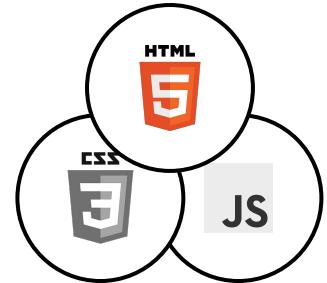


PERFORMANCE & INTEGRATION



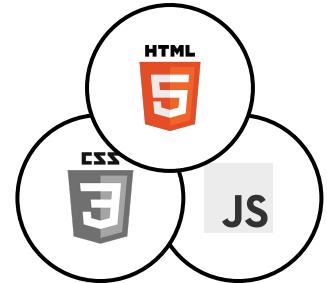
STOCKAGE & HORS-LIGNE

# ÉVOLUTION DES LANGAGES WEB



MULTIMÉDIA

# ÉVOLUTION DES LANGAGES WEB

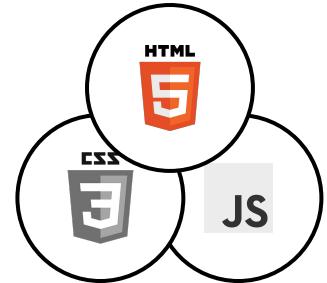


MULTIMÉDIA



ACCÈS DEVICE

# ÉVOLUTION DES LANGAGES WEB

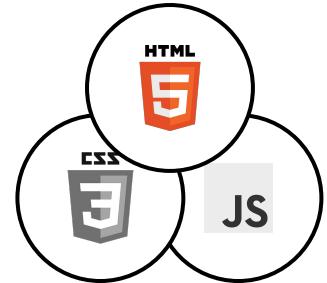


 MULTIMÉDIA

 ACCÈS DEVICE

 3D & EFFETS

# ÉVOLUTION DES LANGAGES WEB



MULTIMÉDIA



ACCÈS DEVICE

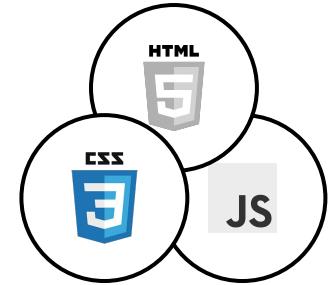


3D & EFFETS

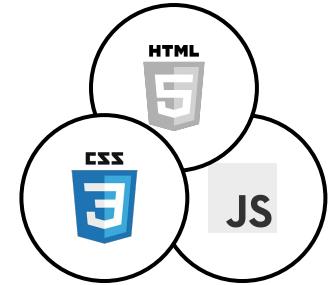


STYLE

# ÉVOLUTION DES LANGAGES WEB

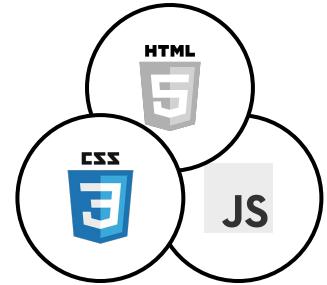


# ÉVOLUTION DES LANGAGES WEB



Sass

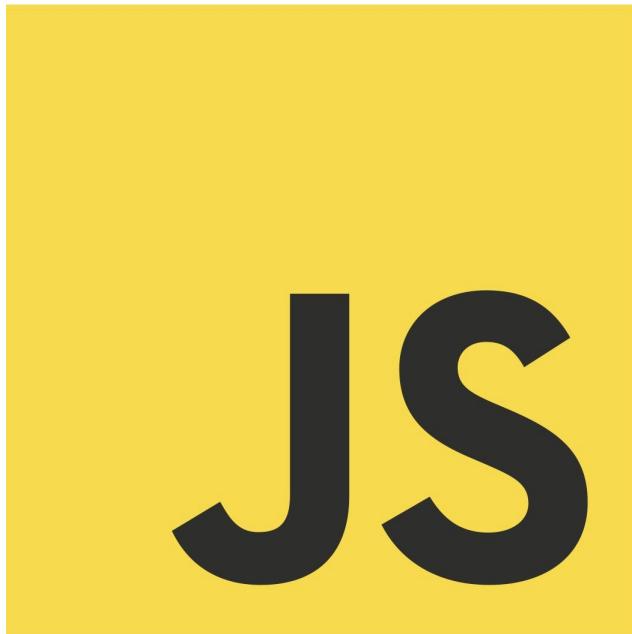
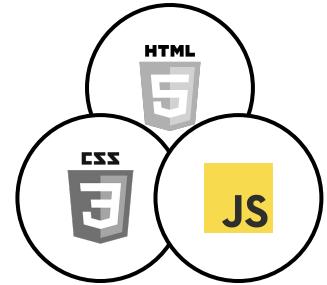
# ÉVOLUTION DES LANGAGES WEB



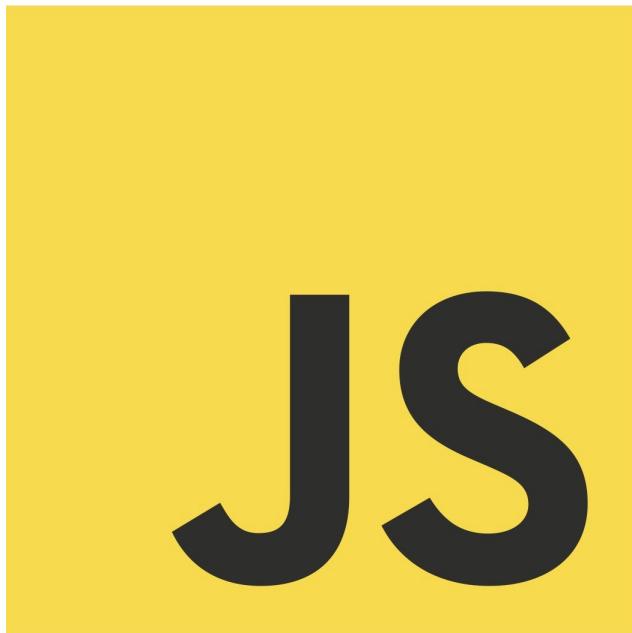
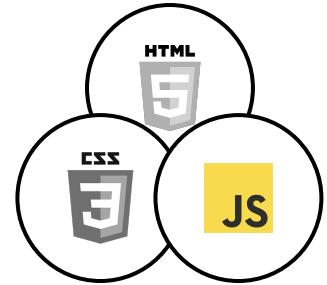
Sass

{less}

# ÉVOLUTION DES LANGAGES WEB



# ÉVOLUTION DES LANGAGES WEB

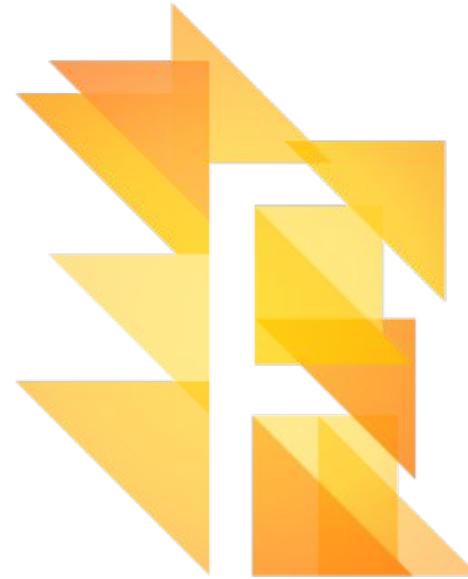
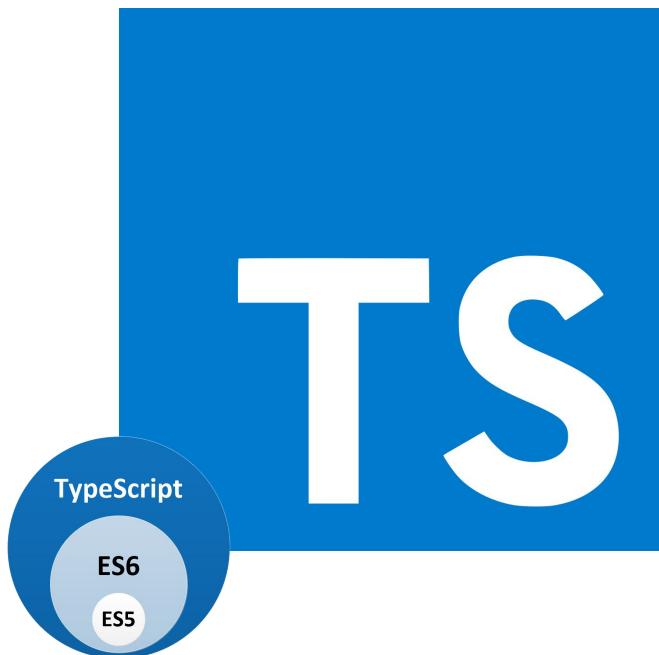
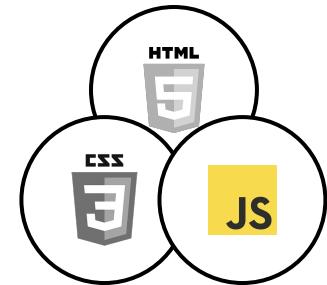


BABEL



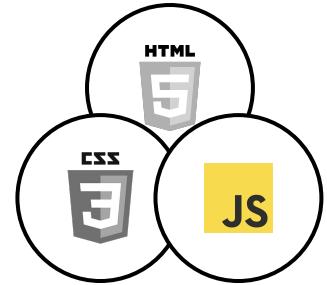
# ÉVOLUTION DES LANGAGES WEB

## Alternatives



# ÉVOLUTION DES LANGAGES WEB

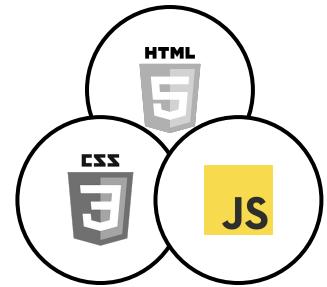
## Indispensables Javascript



LO (✓)

# ÉVOLUTION DES LANGAGES WEB

## Indispensables Javascript

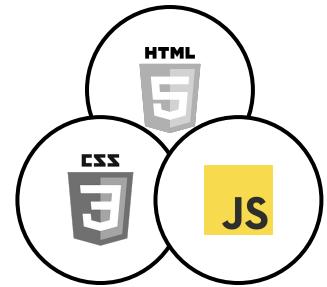


L  
O  
\_\_\_\_\_

```
_debounce(func, [wait=0], [options={}])  
_deburr([string=''])
```

# ÉVOLUTION DES LANGAGES WEB

## Indispensables Javascript



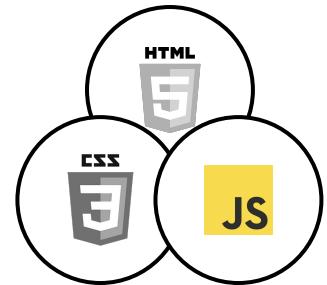
L  
O  
  
\_\_\_\_\_

```
_.memoize(func, [resolver])
_.pick(object, [paths])
_.omit(object, [paths])
```

```
_.debounce(func, [wait=0], [options={}])
_.deburr([string=''])
```

# ÉVOLUTION DES LANGAGES WEB

## Indispensables Javascript



L  
O  
\_\_\_\_\_

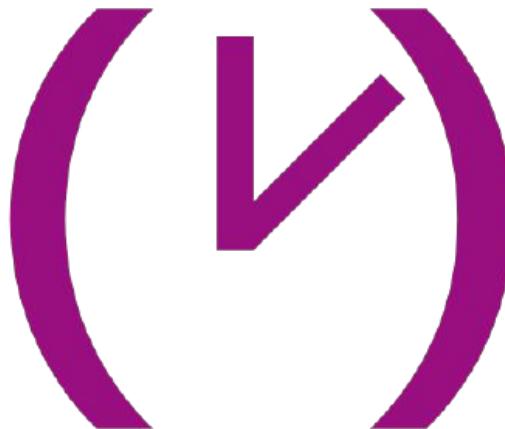
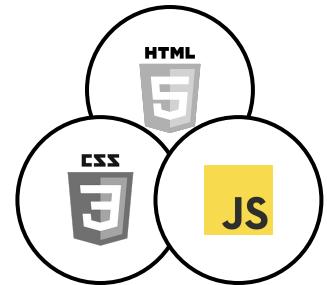
```
_.memoize(func, [resolver])  
_.pick(object, [paths])  
_.omit(object, [paths])
```

```
_.debounce(func, [wait=0], [options={}])  
_.deburr([string=''])
```

```
_.filter(collection, [predicate=_identity])  
_.reduce(collection, [iteratee=_identity], [accumulator])  
_.map(collection, [iteratee=_identity])
```

# ÉVOLUTION DES LANGAGES WEB

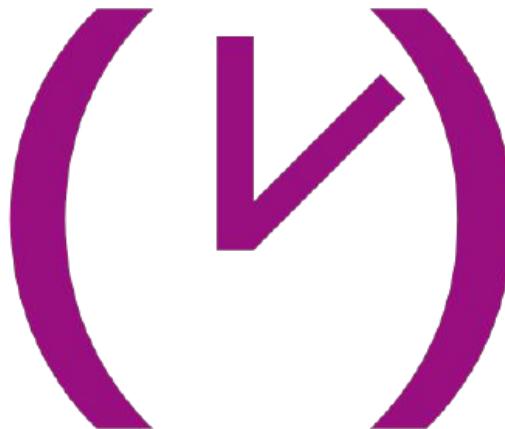
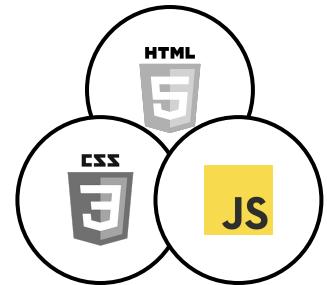
## Indispensables Javascript



```
// Which of these dates is the latest?  
var result = max(  
    new Date(1989, 6, 10),  
    new Date(1987, 1, 11),  
    new Date(1995, 6, 2),  
    new Date(1990, 0, 1)  
)  
//=> Sun Jul 02 1995 00:00:00
```

# ÉVOLUTION DES LANGAGES WEB

## Indispensables Javascript

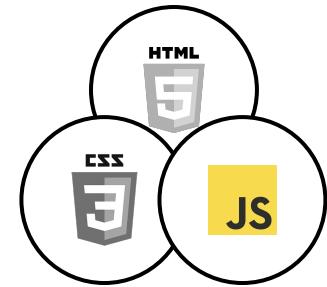


```
// Which of these dates is the latest?  
var result = max(  
  new Date(1989, 6, 10),  
  new Date(1987, 1, 11),  
  new Date(1995, 6, 2),  
  new Date(1990, 0, 1)  
)  
//=> Sun Jul 02 1995 00:00:00
```

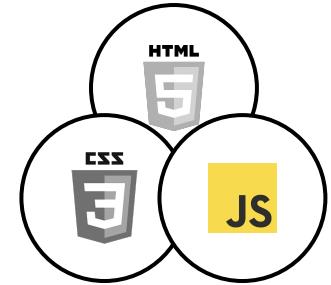
```
// If week starts with Monday,  
// are 31 August 2014 and 4 September 2014 in the same week?  
var result = isSameWeek(  
  new Date(2014, 7, 31),  
  new Date(2014, 8, 4),  
  {weekStartsOn: 1})  
//=> false
```

# ÉVOLUTION DES LANGAGES WEB

JS  
EVERYWHERE



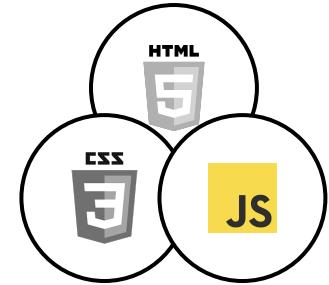
# ÉVOLUTION DES LANGAGES WEB



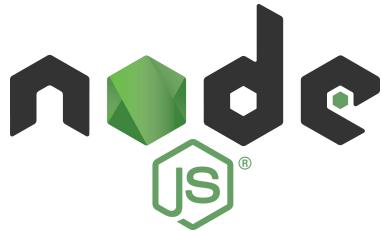
JS  
EVERYWHERE



# ÉVOLUTION DES LANGAGES WEB

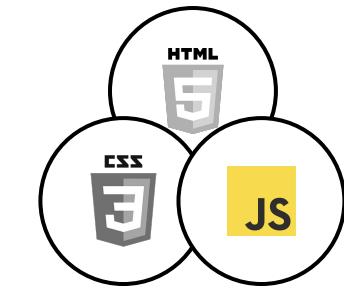
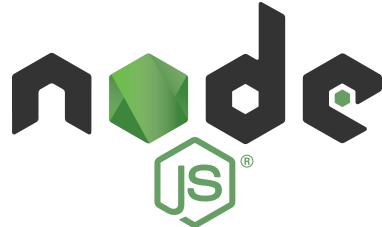


JS  
EVERYWHERE



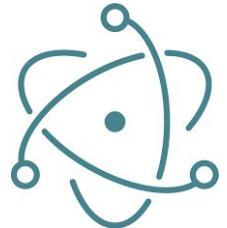
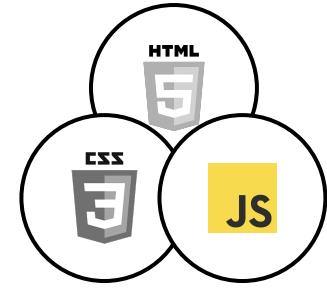
# ÉVOLUTION DES LANGAGES WEB

JS  
EVERYWHERE



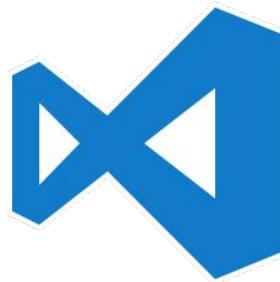
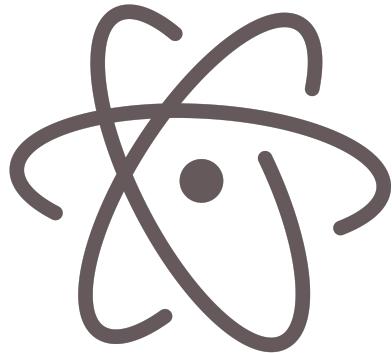
# ÉVOLUTION DES LANGAGES WEB

JS  
EVERYWHERE

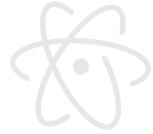


# Outils de développement

# EDITEURS



# EDITEURS



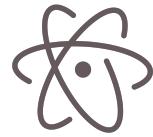
## Sublime Text



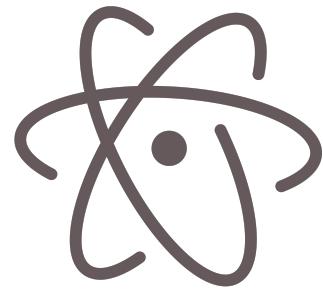
70 \$ par utilisateur

Jon Skinner

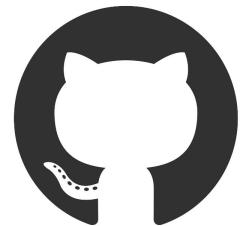
# EDITEURS



## Atom



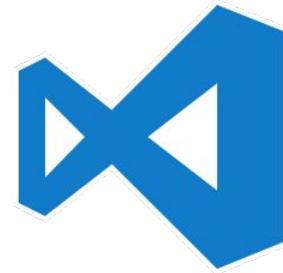
Gratuit



# EDITEURS



## Visual Studio Code



Gratuit



# EDITEURS



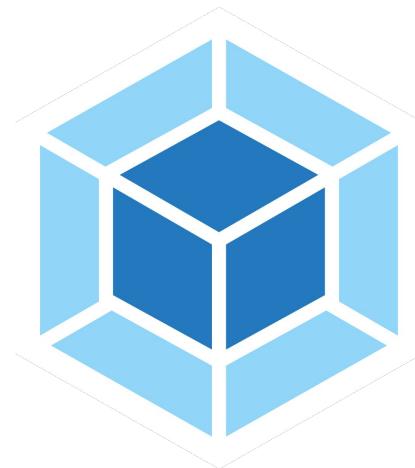
## WebStorm



129 € / util. / an

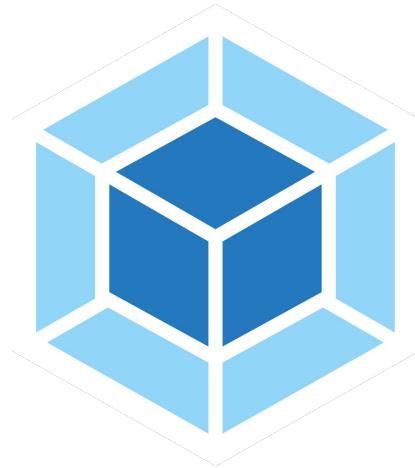


# PACKAGERS D'APPLICATIONS

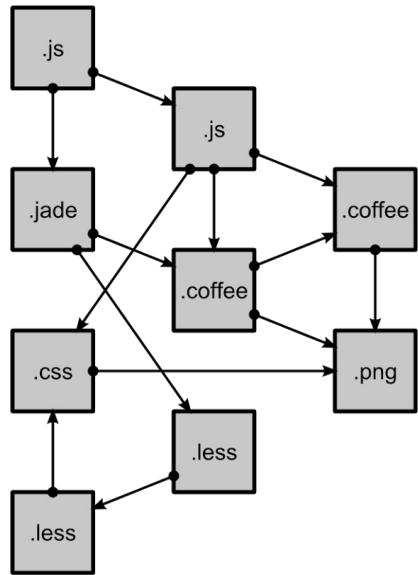


# PACKAGERS D'APPLICATIONS

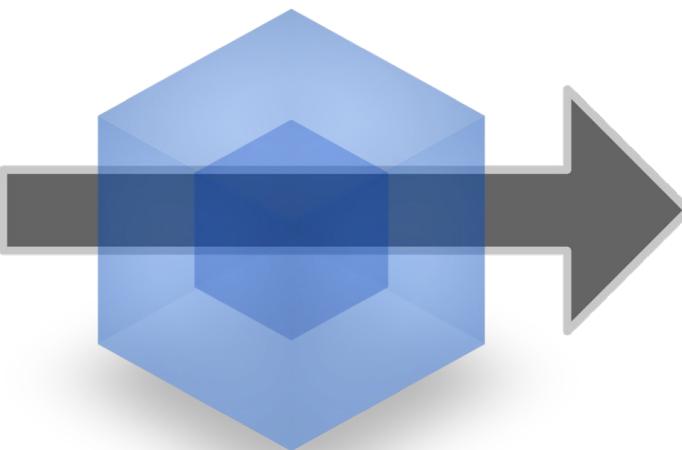
webpack



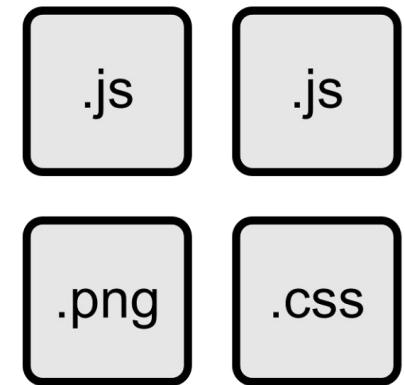
# PACKAGERS D'APPLICATIONS



modules  
with dependencies



**webpack**  
MODULE BUNDLER



static  
assets

# GESTIONNAIRE DE DÉPENDANCES



BOWER



# GESTIONNAIRE DE DÉPENDANCES

bower



BOWER

# GESTIONNAIRE DE DÉPENDANCES

npm



# GESTIONNAIRE DE DÉPENDANCES

yarn



# GESTIONNAIRE DE DÉPENDANCES



BOWER



# GESTIONNAIRE DE DÉPENDANCES



A screenshot of the Bower homepage. The page has a yellow header with navigation links: Docs (blue), Search packages (green), Blog (blue), and Stats (blue). Below the header is a large red bird logo on the left. To the right of the logo, the word "Bower" is written in large red letters, followed by the subtitle "A package manager for the web" in black. At the bottom of the page, there is a blue footer bar with the text "...psst! While Bower is maintained, we recommend [yarn](#) and [webpack](#) for new front-end projects!"  
A red arrow points from the word "SOURCE" to the URL "HTTP://BOWER.IO".

SOURCE  
[HTTP://BOWER.IO](https://bower.io)

# GESTIONNAIRE DE DÉPENDANCES



yarn

# ANALYSE

## N'oubliez pas les devtools !



- Console & Debugger
- DOM & Style
- Performance réseau
- Audits
- Performance du rendu
- Profileur CPU & Mémoire
- Inspecteur de ressources
- Outils responsive & offline

# ANALYSE

“Page speed insight” de Google



EXEMPLE SUR

[HTTP://DEVFEST.GDGNANTES.COM](http://devfest.gdgnantes.com)

# Approche orientée composants

# COMPOSANTS



**Webcomponents**

# COMPOSANTS



## Webcomponents

Custom Elements

# COMPOSANTS



## Webcomponents

Custom Elements

Shadow DOM

# COMPOSANTS



## Webcomponents

Custom Elements

Shadow DOM

HTML imports

# COMPOSANTS



## Webcomponents

Custom Elements

Shadow DOM

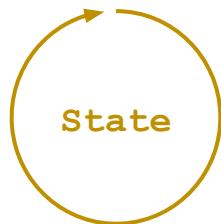
HTML imports

HTML Template

# COMPOSANTS

## Exemple de composant

properties → ElementHTML → View



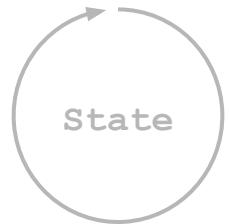
# COMPOSANTS

## Exemple de composant

```
<Like />
```



properties → ElementHTML → View

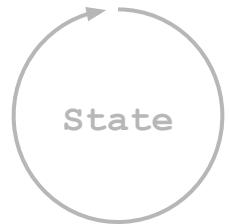


# COMPOSANTS

## Exemple de composant

```
<Like nbLikes="8" />
```

properties → ElementHTML → View



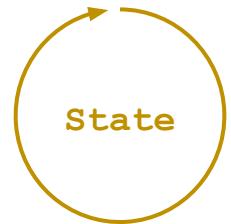
# COMPOSANTS

## Exemple de composant

```
<Like nbLikes="8" />
```

```
//isLiked=true
```

properties → ElementHTML → View



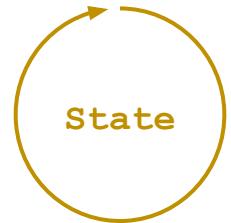
# COMPOSANTS

## Exemple de composant

```
<Like nbLikes="8" />  
//isLiked=true
```



properties → ElementHTML → View



# COMPOSANTS

## Exemple de composant

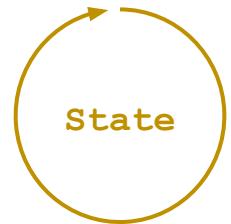
```
<Like nbLikes="8" />  
//isLiked=true
```



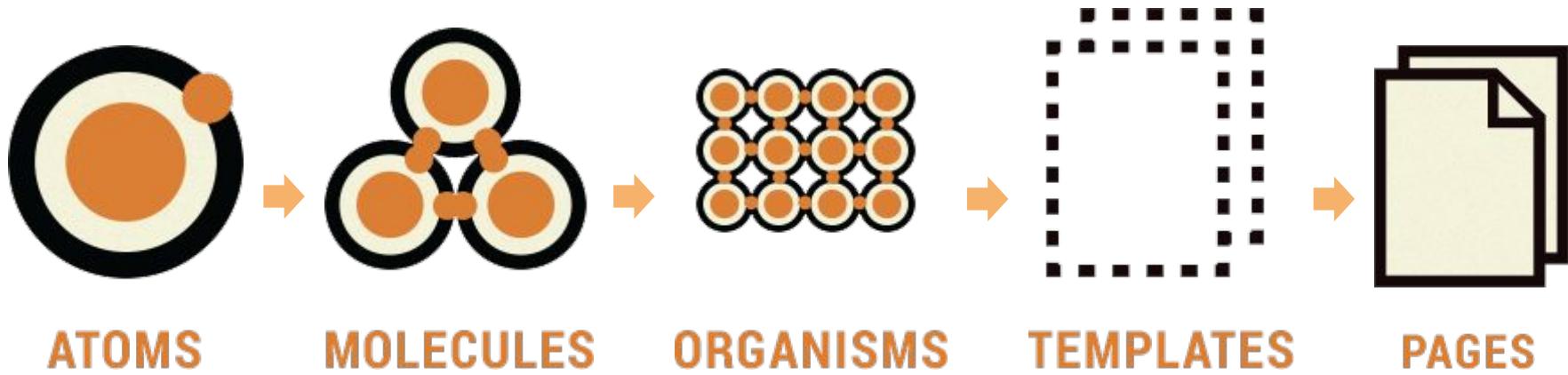
```
<Like nbLikes="278" />  
//isLiked=false
```



properties → ElementHTML → View



# ATOMIC DESIGN



# ATOMIC DESIGN



ATOMS



MOLECULES



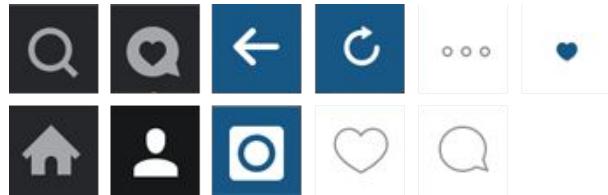
ORGANISMS



TEMPLATES



PAGES



150 x 150

PHOTO

XXXXXX likes

thisistheusersinstagramhandle

2h

“ Lorem ipsum dolor sit amet, consectetur  
adipisicing ”

1080 x 1080

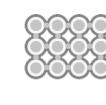
# ATOMIC DESIGN



ATOMS



MOLECULES



ORGANISMS



TEMPLATES



PAGES

The image shows a mobile Instagram interface. At the top, there's a blue header bar with a back arrow, the word "PHOTO" in white, and a refresh/circular arrow icon. Below the header is a user profile card for "thisistheusersinstagramhandle" with a timestamp of "2h". The main area features a large, light-gray placeholder image with the text "1080 x 1080" centered in it. Below the image are standard Instagram interaction icons: a heart for likes, a speech bubble for comments, and three dots for more options. Further down is a row of metrics: "XXXXXX likes" (with a blue heart icon). At the bottom is a caption box containing the text "thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipisicing elit.". At the very bottom is a dark navigation bar with five icons: a house (Home), a magnifying glass (Search), a camera (Post), a speech bubble (Comments), and a person (Profile).

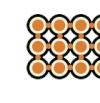
# ATOMIC DESIGN



ATOMS



MOLECULES



ORGANISMS



TEMPLATES



PAGES



# ATOMIC DESIGN



ATOMS



MOLECULES



ORGANISMS



TEMPLATES



PAGES



# ATOMIC DESIGN



ATOMS



MOLECULES



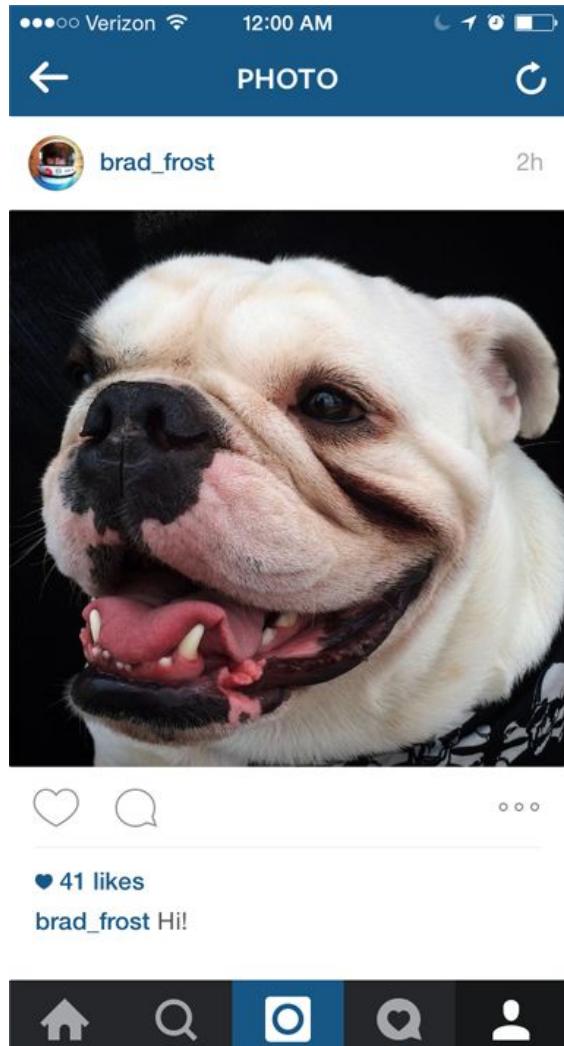
ORGANISMS



TEMPLATES

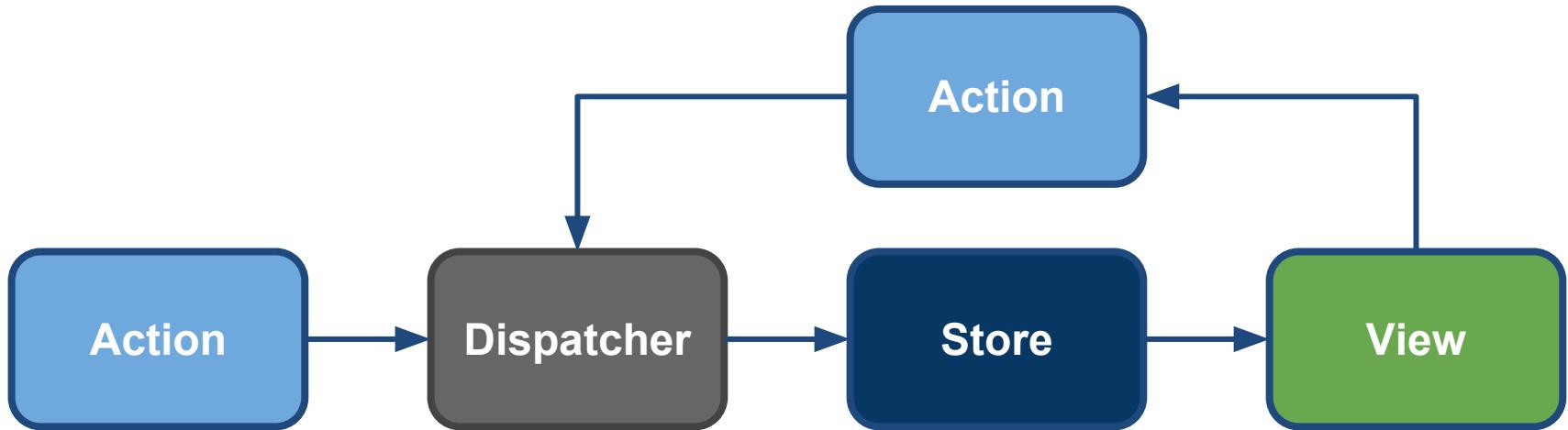


PAGES



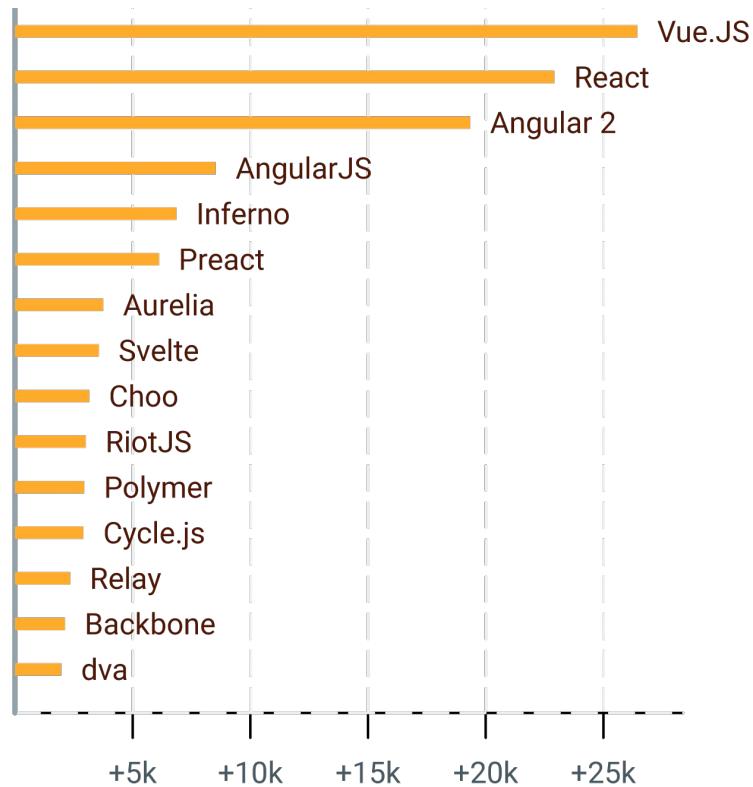
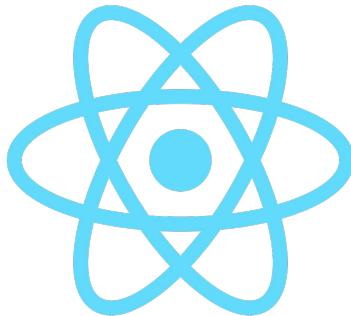
# COMMUNIQUER

Flux



# Pourquoi ?

# Librairies & Frameworks



Source : <https://risingstars2016.js.org/>



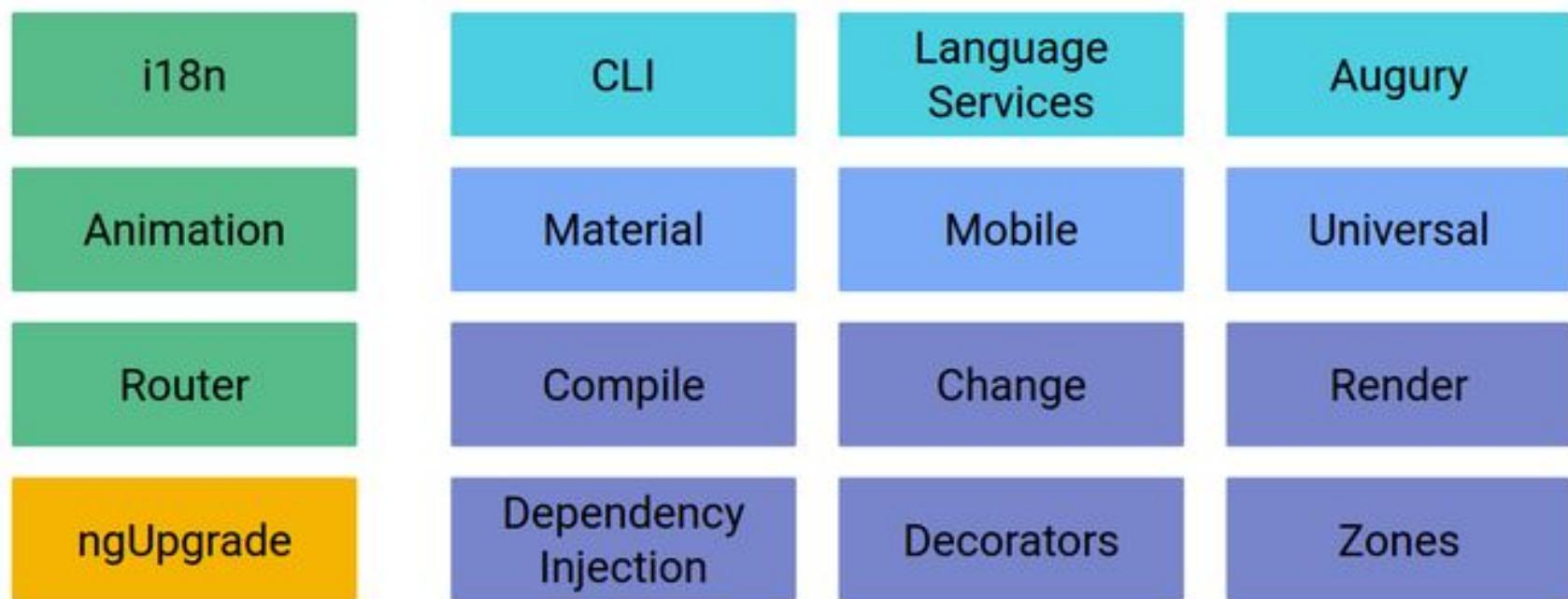
# ANGULAR

Tout court

# DIFFÉRENCES AVEC ANGULAR JS

- Réécriture totale
- API largement simplifiée
- Architecture orientée composant
- TypeScript
- Angular est plus qu'un framework, c'est une plateforme

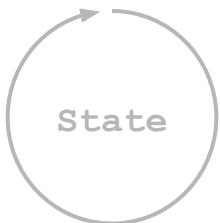
# ANGULAR, UN FRAMEWORK, UNE PLATEFORME



# PREMIER COMPOSANT

```
import { Component } from '@angular/core'  
[...]  
export class LikeComponent {  
    [...]  
}
```

properties → **ElementHTML** → View

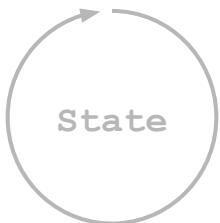


# PREMIER COMPOSANT

```
import { Component } from '@angular/core'

@Component({
  selector: 'like', // <like></like>
  template: `
    <div>
      <i class="icon" />
    </div>
  `,
})
export class LikeComponent {
  [...]
}
```

properties → **ElementHTML** → View

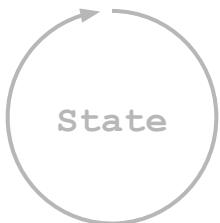


# PREMIER COMPOSANT

```
import { Component } from '@angular/core'

@Component({
  selector: 'like', // <like></like>
  template: `
    <div>
      <i class="icon" />
    </div>
  `,
})
export class LikeComponent {
  [...]
}
```

properties → **ElementHTML** → **View**

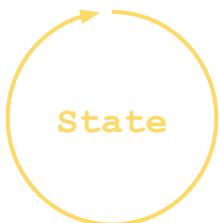


# PREMIER COMPOSANT

```
import { Component } from '@angular/core'

@Component({
  selector: 'like', // <like></like>
  template: `
    <div>
      <i class="icon" [ngClass]="'liked': isLiked" />
    </div>
  `,
})
export class LikeComponent {
  isLiked: boolean = false;
}
```

properties → ElementHTML → View

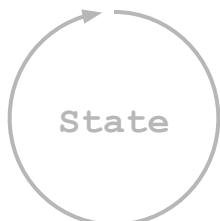


# PROPERTY BINDING = []

```
import { Component, Input } from '@angular/core'

@Component({
  selector: 'like', // <like [nbLike]="3"></like>
  template: `
    <div>
      <p>{{ nbLike }}</p>
      <i class="icon" [ngClass]="'liked': isLiked" />
    </div>
  `,
})
export class LikeComponent {
  isLiked: boolean = false;
  @Input() nbLike: number;
}
```

properties → ElementHTML → View



# EVENT BINDING = ()

```
import { Component, Input } from '@angular/core'

@Component({
  selector: 'like', // <like [nbLike]="3"></like>
  template: `
    <div (click)="clickLike()">
      <p>{{ nbLike }}</p>
      <i class="icon" [ngClass]="'liked': isLiked" />
    </div>
  `,
})
export class LikeComponent {
  isLiked: boolean = false;
  @Input() nbLike: number;

  clickLike() {
    this.isLiked = !this.isLiked;
  }
}
```

properties → ElementHTML → View



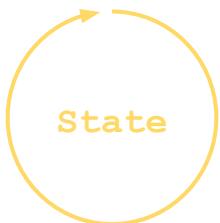
# PREMIER COMPOSANT

```
import { Component, Input } from '@angular/core'

@Component({
  selector: 'like', // <like [nbLike]="3"></like>
  template: `
    <div (click)="clickLike()" >
      <p>{{ nbLike }}</p>
      <i class="icon" [ngClass]="'liked': isLiked" />
    </div>
  `,
})
export class LikeComponent {
  isLiked: boolean = false;
  @Input() nbLike: number;

  clickLike() {
    this.isLiked = !this.isLiked;
  }
}
```

properties → ElementHTML → View



# API

- Types d'objets

- `@NgModule()`
- `@Component()`

```
export class MyComponent { }
```

- `@Injectable()`
- `@Pipe()`

# API

- Types d'objets

- `@NgModule()`
- `@Component()`

```
export class MyComponent { }
```

- `@Injectable()`
- `@Pipe()`

- Paramètres d'un composant

- `@Input()`

# API

- Types d'objets

- `@NgModule()`
- `@Component()`

```
export class MyComponent { }
```

- `@Injectable()`
- `@Pipe()`

- Paramètres d'un composant

- `@Input()`

- État d'un composant

- `private myState: boolean;`

# API

- Types d'objets

- @NgModule()
- @Component()

```
export class MyComponent { }
```

- @Injectable()
- @Pipe()

- Paramètres d'un composant

- @Input()

- État d'un composant

- private myState: boolean;

- Templating

- Directives: \*ngIf, \*ngFor, [ngClass]

# API

- Types d'objets
  - @NgModule()
  - @Component()

```
export class MyComponent {}
```
- Paramètres d'un composant
  - @Input()
- État d'un composant
  - private myState: boolean;
- Templating
  - Directives: \*ngIf, \*ngFor, [ngClass]
- Cycle de vie d'un composant
  - ngOnChanges
  - ngOnInit
  - ngOnDestroy
  - etc...



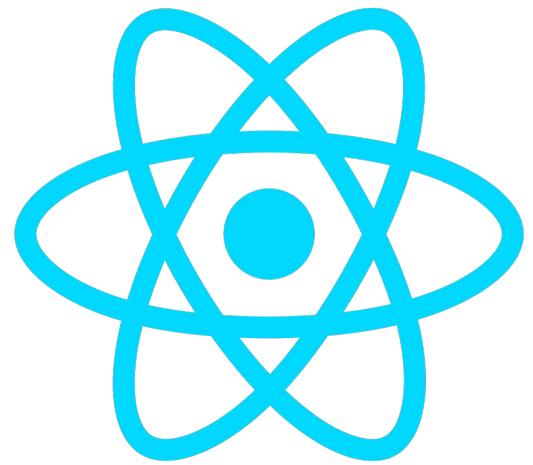
# GÉNÉRATEUR



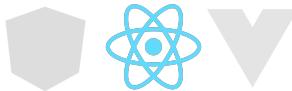
# ANGULAR CLI



- ✓ Performant
- ✓ Plateforme complète
- ✓ Porté par Google
- ✓ Projets robustes et maintenables (TypeScript)
  
- ✗ Prise en main plus longue (TypeScript)
- ✗ Plateforme en cours d'implémentation



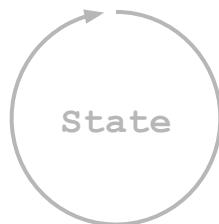
React

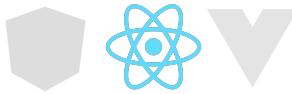


# PREMIER COMPOSANT

```
class Like extends React.Component {  
  [...]  
}  
  
<Like />
```

properties → **ElementHTML** → View

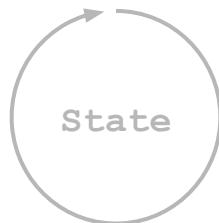


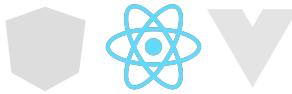


# PREMIER COMPOSANT

```
class Like extends React.Component {  
  render() {  
    return (  
      <div>  
        <p> 3 </p>  
        <i className="icon" />  
      </div>  
    )  
  }  
}  
  
<Like />
```

properties → ElementHTML → **View**

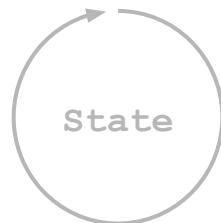


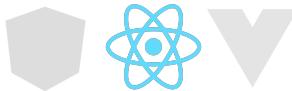


# PREMIER COMPOSANT

```
class Like extends React.Component {  
    render() {  
        return (  
            <div>  
                <p> {this.props.nbLikes} </p>  
                <i className="icon" />  
            </div>  
        )  
    }  
}  
  
<Like nbLikes="3" />
```

properties → ElementHTML → View

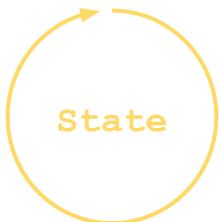


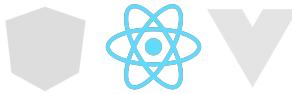


# PREMIER COMPOSANT

```
class Like extends React.Component {  
  constructor() {  
    this.state = {  
      isLiked: true  
    }  
  }  
  render() {  
    iconClasses = this.state.isLiked ? "icon liked" : "icon"  
    return ...  
  }  
}  
  
<Like nbLikes="3"/>
```

properties → ElementHTML → View



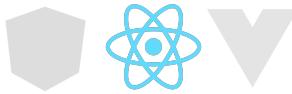


# PREMIER COMPOSANT

```
class Like extends React.Component {  
  constructor() {  
    this.state = { isLiked: true }  
  }  
  toggleLike = () => {  
    this.setState({ isLiked: !this.state.isLiked })  
  }  
  render() {  
    iconClasses = this.state.isLiked ? "icon liked" : "icon"  
    return (  
      <div onClick={this.toggleLike}>  
        <p> {this.props.nbLikes} </p>  
        <i className={iconClasses} />  
      </div>  
    )  
  }  
}  
  
<Like nbLikes="3"/>
```

properties → ElementHTML → View



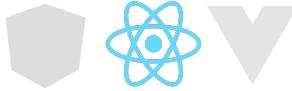


# PREMIER COMPOSANT

```
class Like extends React.Component {  
  constructor() {  
    this.state = { isLiked: true }  
  }  
  toggleLike = () => {  
    this.setState({ isLiked: !this.state.isLiked })  
  }  
  render() {  
    iconClasses = this.state.isLiked ? "icon liked" : "icon"  
    return (  
      <div onClick={this.toggleLike}>  
        <p> {this.props.nbLikes} </p>  
        <i className={iconClasses} />  
      </div>  
    )  
  }  
}  
  
<Like nbLikes="3" />
```

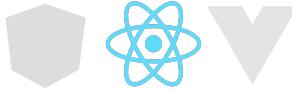
properties → ElementHTML → View





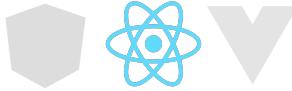
# API

- Créer un composant
  - `createClass`
  - `extends Component`
  - `pure function`



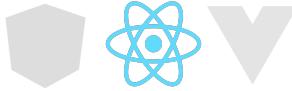
# API

- Créer un composant
  - `createClass`
  - `extends Component`
  - `pure function`
- Paramètres d'un composant
  - `this.props`



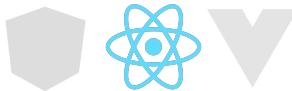
# API

- Créer un composant
  - `createClass`
  - `extends Component`
  - `pure function`
- Paramètres d'un composant
  - `this.props`
- État d'un composant
  - `this.state`



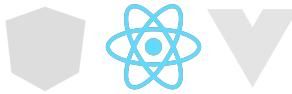
# API

- Crée un composant
  - `createClass`
  - `extends Component`
  - `pure function`
- Paramètres d'un composant
  - `this.props`
- État d'un composant
  - `this.state`
- Templating
  - `JSX`

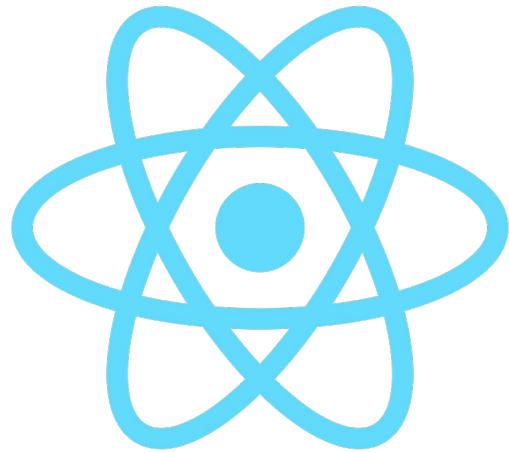


# API

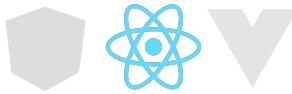
- Crée un composant
  - `createClass`
  - `extends Component`
  - `pure function`
- Paramètres d'un composant
  - `this.props`
- État d'un composant
  - `this.state`
- Templating
  - `JSX`
- Cycle de vie d'un composant
  - `componentWillMount`
  - `componentDidMount`
  - etc...



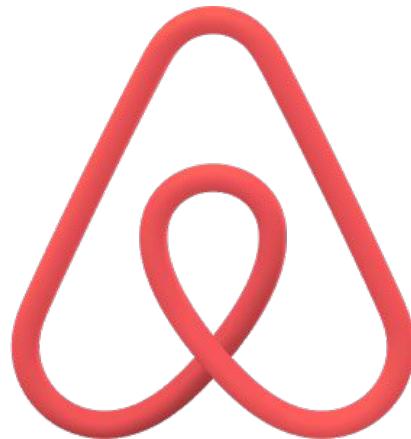
# GÉNÉRATEUR

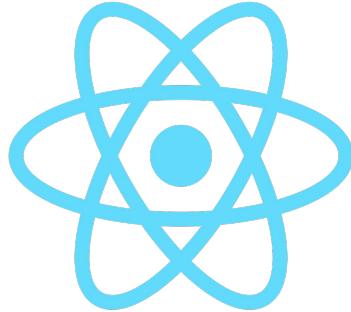


Create React App



# ADOPTION





- ✓ Performant
- ✓ Prise en main simple & rapide
- ✓ Communauté très active (Porté par Facebook)
- ✓ JS Centrique (JSX etc...)
  
- ± Librairie et non framework
  
- ✗ Non respect des standards du web

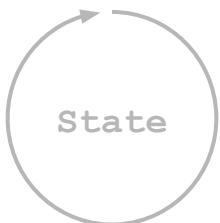


Vue.js

# PREMIER COMPOSANT

```
Vue.component('like', {  
  [ ... ]  
})  
  
<like></like>
```

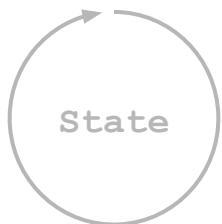
properties → **ElementHTML** → View



# PREMIER COMPOSANT

```
Vue.component('like', {  
  template: `  
    <div>  
      <p> 3 </p>  
      <i class="icon" />  
    </div>  
  `,  
})  
  
<like></like>
```

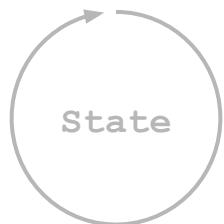
properties → ElementHTML → **View**



# PREMIER COMPOSANT

```
Vue.component('like', {  
  props: ['nbLikes'],  
  template:  
    <div>  
      <p>{{ nbLikes }}</p>  
      <i class="icon" />  
    </div>  
  } )  
  
<like :nb-likes="3"></like>
```

**properties** → ElementHTML → View



# PREMIER COMPOSANT

```
Vue.component('like', {  
  props: ['nbLikes'],  
  data: function() {  
    return {  
      isLiked: false  
    }  
  },  
  template: `  
    <div>  
      <p>{{ nbLikes }}</p>  
      <i :class="{ icon: true, like: isLiked }" />  
    </div>  
  `,  
})  
<like :nb-likes="3"></like>
```

properties → ElementHTML → View



# PREMIER COMPOSANT

```
Vue.component('like', {
  props: ['nbLikes'],
  data: function() {
    return {
      isLiked: false
    }
  },
  toggleLike: () => this.isLiked = !this.isLiked,
  template: `
    <div @click="toggleLike()">
      <p>{{ nbLikes }}</p>
      <i :class="{ icon: true, like: isLiked }" />
    </div>
  `
})
<like :nb-likes="3"></like>
```

properties → ElementHTML → View



# PREMIER COMPOSANT

```
Vue.component('like', {
  props: ['nbLikes'],
  data: function() {
    return {
      isLiked: false
    }
  },
  toggleLike: () => this.isLiked = !this.isLiked,
  template: `
    <div @click="toggleLike()" >
      <p>{{ nbLikes }}</p>
      <i :class="{ icon: true, like: isLiked }" />
    </div>
  `
})
<like :nb-likes="3"></like>
```

properties → ElementHTML → View



# API

- Créer un composant
  - `Vue.component()`

# API

- Créer un composant
  - `Vue.component()`
- Paramètres d'un composant
  - `props: []`

# API

- Crée un composant
  - `Vue.component()`
- Paramètres d'un composant
  - `props: []`
- État d'un composant
  - `data: () => {}`

# API

- Crée un composant
  - `Vue.component()`
- Paramètres d'un composant
  - `props: []`
- État d'un composant
  - `data: () => {}`
- Templating
  - Directives: `v-bind`, `v-on...`

# API

- **Créer un composant**
  - `Vue.component()`
- **Paramètres d'un composant**
  - `props: []`
- **État d'un composant**
  - `data: () => {}`
- **Templating**
  - Directives: `v-bind`, `v-on...`
- **Cycle de vie d'un composant**
  - `beforeCreate`
  - `mounted`
  - `beforeDestroy`
  - `etc...`



# GÉNÉRATEUR

CLI





~~ADOPTION~~ FORT INTÉRÊT





- ✓ Performant
- ✓ Prise en main simple & rapide
- ✓ Templating proche d'AngularJS
  
- ± Librairie et non framework
  
- ✗ Communauté plus jeune (mais en expansion)



zenika

MAINTENANT ON CODE !!!

# Testing

# LINTING

## ESLint ...



# LINTING

## ESLint ...



 **linter-eslint**

Lint JavaScript on the fly, using ESLint

 AtomLinter

758,014 ⚡ 1195 ⭐

**ESLint**  
Dirk Baeumer | 927,951 installs | ★★★★★ (65)

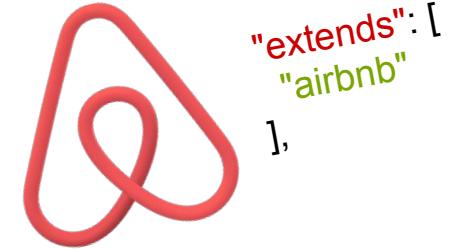
Integrates ESLint into VS Code.

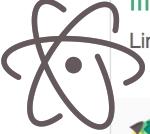
**Installation**  
Launch VS Code Quick Open (⌘+P), paste the following command, and press enter.

```
ext install vscode-eslint | Copy | More Info
```

# LINTING

## ESLint ...



 **linter-eslint**

Lint JavaScript on the fly, using ESLint

 AtomLinter

758,014 ⚡ 1195

  **ESLint**

Dirk Baeumer | 927,951 installs | ★★★★★ (65)

Integrates ESLint into VS Code.

**Installation**

Launch VS Code Quick Open ( $\text{⌘}+\text{P}$ ), paste the following command, and press enter.

`ext install vscode-eslint` [Copy](#) | [More Info](#)

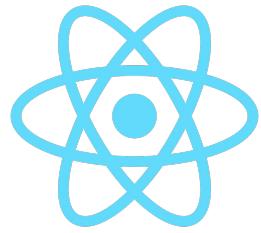
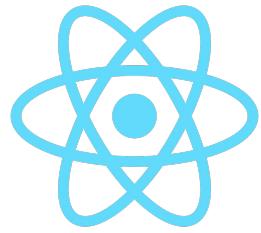
# TESTING

Jest ...



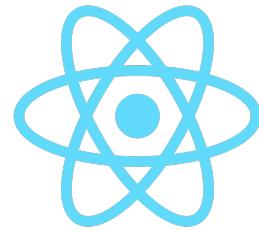
# TESTING

Jest ...



# TESTING

Jest ...



# TESTING

## Jest ...



```
const snap = (props) => {
  const tree = testedComponent.toJSON()
  expect(tree).toMatchSnapshot()
}

describe('common/Number', () => {
  it('should add children', () => snap({ children: 'a child' }))
})
```

# TESTING

## Jest ...



```
PASS  src/number/number.spec.js
      common/Number
          graphical (JSX)
            ✓ should add children (11ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   1 passed, 1 total
Time:        0.655s, estimated 1s
Ran all test suites related to changed files.

Watch Usage
> Press a to run all tests.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press q to quit watch mode.
> Press Enter to trigger a test run.
```

# TESTING

## Jest ...



```
FAIL  src/number/number.spec.js
  ● common/Number > graphical (JSX) > should add children

    expect(value).toMatchSnapshot()

    Received value does not match stored snapshot 1.

      - Snapshot
      + Received

      <div
        className="number primary"
        style={Object {}}
      >
      +  <div>
      +    Jest snapshot
      +  </div>
      a child
    </div>
```

# Mobile is today !

# JS MOBILE

## Usage Internet par plateforme

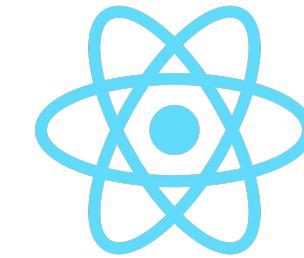
Octobre 2009 - Octobre 2016



Source: <http://gs.statcounter.com/>



# JS MOBILE



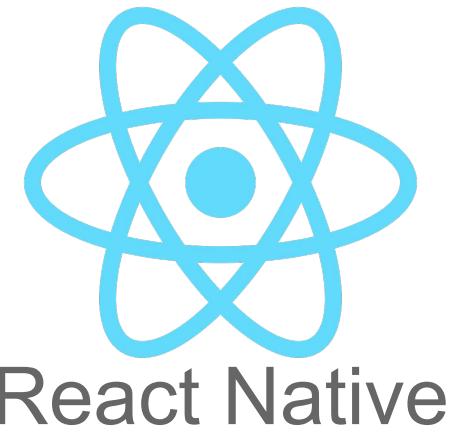
React Native



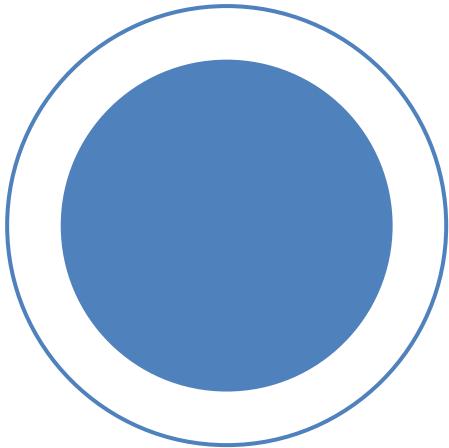
# HYBRIDE



# LE “NATIF”

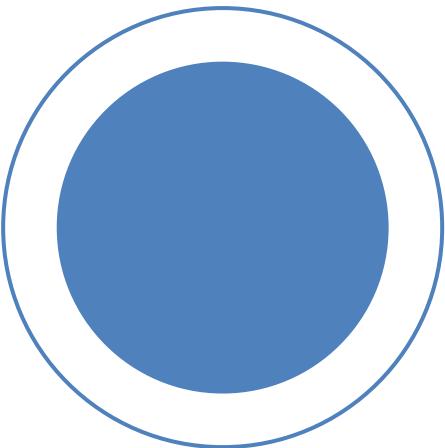


# PROGRESSIVE WEB APP

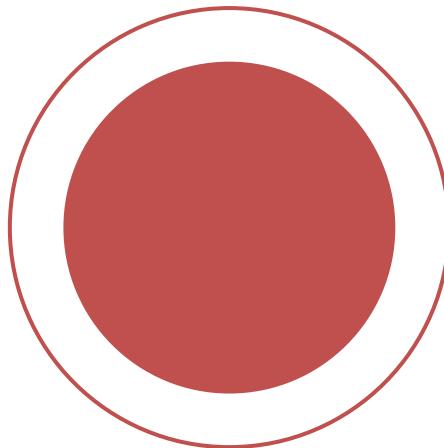


Immersive

# PROGRESSIVE WEB APP

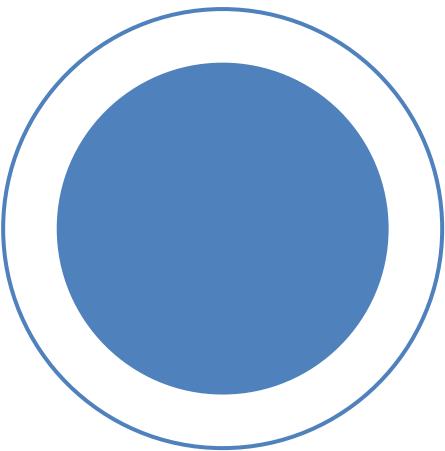


Immersive

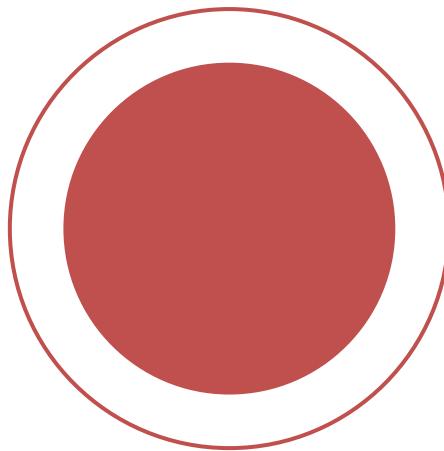


Performante

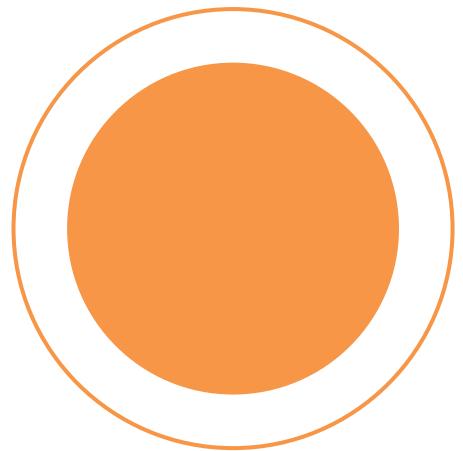
# PROGRESSIVE WEB APP



Immersive



Performante



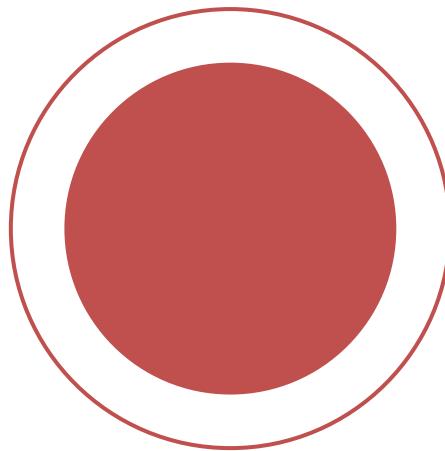
Engageante

# PROGRESSIVE WEB APP

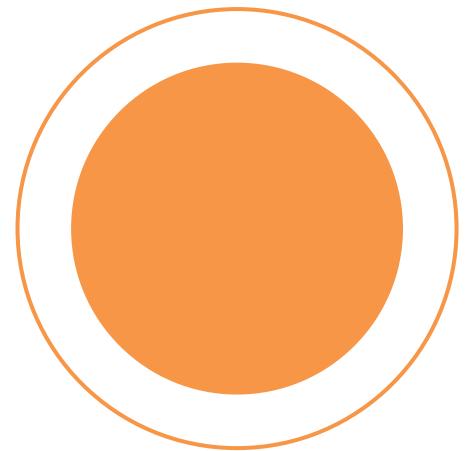
Immersive

---

Full screen,  
theming,  
orientation,  
etc



Performante



Engageante

# PROGRESSIVE WEB APP

Immersive

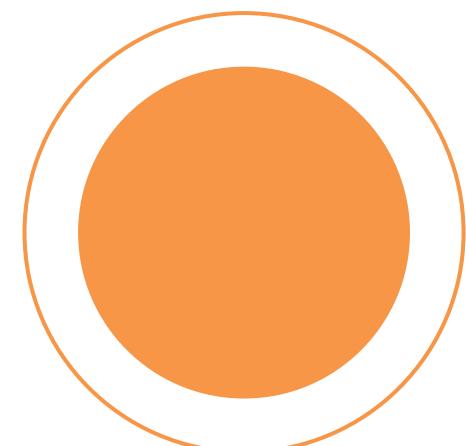
---

Full screen,  
theming,  
orientation,  
etc

Performante

---

Offline,  
60 FPS,  
Cache,  
etc



Engageante

# PROGRESSIVE WEB APP

## Immersive

---

Full screen,  
theming,  
orientation,  
etc

## Performante

---

Offline,  
60 FPS,  
Cache,  
etc

## Engageante

---

Push  
notifications



# PROGRESSIVE WEB APP

Since launching their PWA, **Lancôme**'s mobile sales have increased **16%** year over year with overall speed increases of **50%**.



# PROGRESSIVE WEB APP

Since launching their PWA, **Lancôme**'s mobile sales have increased **16%** year over year with overall speed increases of **50%**.



**Twitter Lite** is interactive in less than **5** seconds over 3G on most devices, with average load times reduced by over **30%**.



# MOBILE TODAY !

App complexe / Performance >

**Responsive**

Immersion utilisateur >

# MOBILE TODAY !

App complexe / Performance >

PWA

Responsive

Immersion utilisateur >

# MOBILE TODAY !

App complexe / Performance >

**Hybride**

**PWA**

**Responsive**

Immersion utilisateur >

# MOBILE TODAY !

App complexe / Performance >

**Natif multi-plateforme**

**Hybride**

**PWA**

**Responsive**

Immersion utilisateur >

# MOBILE TODAY !

App complexe / Performance >

Natif par plateforme

Natif multi-plateforme

Hybride

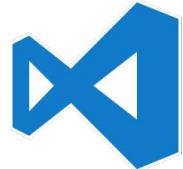
PWA

Responsive

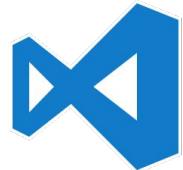
Immersion utilisateur >

Ma stack !  
2017

### EDITEURS



EDITEURS



DÉPENDANCES



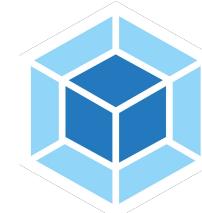
EDITEURS



DÉPENDANCES



PACKAGERS



### ÉDITEURS

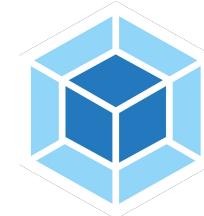


### DÉPENDANCES

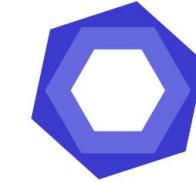


yarn

### PACKAGERS



### QUALITÉ



### ÉDITEURS

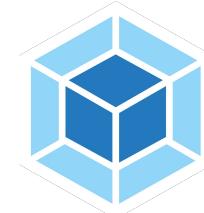


### DÉPENDANCES



yarn

### PACKAGERS



### QUALITÉ



### LANGAGES

JS

### ÉDITEURS



### DÉPENDANCES



### PACKAGERS



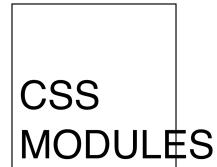
### QUALITÉ



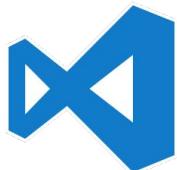
### LANGAGES



### STYLING



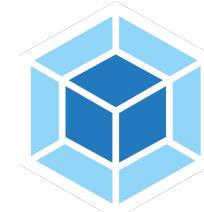
## EDITEURS



## DÉPENDANCES



## PACKAGERS



## QUALITÉ



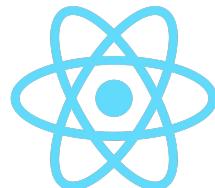
## LANGAGES



## STYLING



## FRAMEWORKS



## EDITEURS

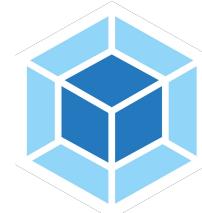


## DÉPENDANCES



yarn

## PACKAGERS



## QUALITÉ



## LANGAGES

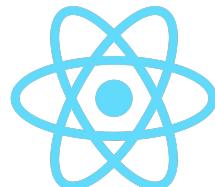


## STYLING

CSS  
MODULES



## FRAMEWORKS



## LIBRAIRIES

Lo (v)



zenika

QUESTIONS ? :D