

PROFOUND SURREAL

C'EST PAS SIMPLEMENT DÉBUTER REACT EN 2021. SURTOUT REACT C'EST DATA

MAGNET 2-DUALS - THOUGHTS



2014

2016

2019

2017

+ HOC

+ REDUX

+ Render props

+ Context

+ Mixins

+ Flux

+ Hook

+ Context

```
var Component2014 = React.createClass({  
  render(props){  
    ...  
  }  
  ...  
});
```

```
const StatelessComponent2016 = (props) => <div>...</div>;
class StatefulComponent extends React.Component {
  constructor(props) {
    super(props);
    ...
  }
  render(props) {
    ...
  }
}
```



```
class StatelessComponent2017 extends React.PureComponent {  
  render(props) {  
    ...  
  }  
}  
  
class StatefullComponent extends React.Component {  
  constructor(props) {  
    super(props);  
    ...  
  }  
  render(props) {  
    ...  
  }  
}
```

+ API changes

+ API changes

```
const futureComponent = (props) => {  
  ...  
  return <div> ... </div>  
}
```

Les approches >2016 fonctionnent encore

Autant de « visions » que de projets ... c'est normal react est une lib qui laisse beaucoup de liberté dans l'architecture du projet

DANS LE CADRE DU COURS JE VOUS IMPOSE UNE « VISION » à respecter

... même si vous utilisez déjà react autrement

POINT D'ATTENTION SUR REACT

C'EST PAS SIMPLE DE DÉBUTER REACT EN 2021 ... SURTOUT QUE REACTJS.ORG EST DATÉ

2014

2016



2017

2019

```
var Component2014 = React.createClass({
  render(props){
    ...
  }
});
```

```
const StatelessComponent2016 = (props) => <div>...</div>;
class StatefullComponent extends React.Component {
  constructor(props) {
    super(props);
    ...
  }
  render(props) {
    ...
  }
}
```

```
class StatelessComponent2017 extends React.PureComponent {
  render(props) {
    ...
  }
}
class StatefullComponent extends React.Component {
  constructor(props) {
    super(props);
    ...
  }
  render(props) {
    ...
  }
}
```

```
const futureComponent = (props) => {
  ...
  return <div>...</div>
}
```

+ Mixins
+ Flux

+ HOC
+ REDUX

+ API changes

+ Render props
+ Context

+ API changes

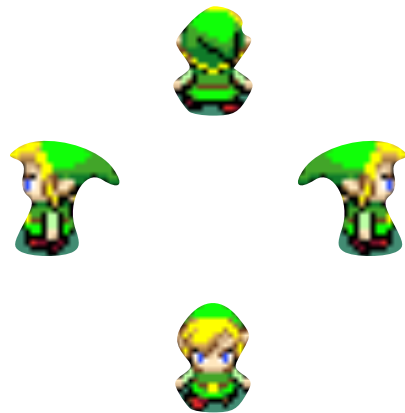
+ Hook
+ Context

DANS LE CADRE DU COURS JE VOUS IMPOSE UNE « VISION » à respecter

... même si vous utilisez déjà react autrement

STATE MACHINE STRIKE BACK

SI ON ADAPTAIT NOTRE MACHINE EN TYPESCRIPT



```
interface North {  
  type: « north »;  
}  
interface East {  
  type: "east";  
}  
interface South {  
  type: "south";  
}  
interface West {  
  type: "west";  
}  
  
type Direction = North | East | South | West  
  
const label = (d: Direction) => {  
  switch (d.type) {  
    case "north": return "North"  
    case "east": return "East"  
    case "south": return "South"  
    case "west": return "West"  
  }  
}
```

