```
tags:
- react
MOC: [[React]]
```

useState

Syntaxe

```
import { useState } from "react";

const [number, setNumber] = useState(0);
// or
const [string, setString] = useState("toto");
// or
const [boolean, setBoolean] = useState(true);
// or
const [array, setArray] = useState([0, 1, 2]);
// or
const [object, setObject] = useState({
        id: 1,
        name: "toto",
        activities: ["painting", "running", "reading"]
});
// or
const [nothing, setNothing] = useState(); // -> undefined
```

Utilisation de setState...

... avec des nombres

```
const [number, setNumber] = useState(0);

setNumber(previousNumber => previousNumber + 1);
// or
setNumber(number + 1);
// or
setNumber(0);
```

... avec des chaînes de caractères

```
const [string, setString] = useState("toto");

setString(previousString => previousString + "a");
// or
setString(string + "b");
```

```
// or
onChange={(e) => setString(e.target.value)};
```

... avec des booléens

```
const [boolean, setBoolean] = useState(true);

setBoolean(false);
// or
setBoolean(previousBoolean => !previousBoolean);
// or
setBoolean(!boolean);
```

... avec des tableaux

```
const [array, setArray] = useState([0, 1, 2]);

setArray(previousArray => [...previousArray, newElement]);
```

...avec des objets

```
const [object, setObject] = useState({
        id: 1,
        name: "toto",
        activities: ["painting", "running", "reading"]
});

setObject(newObject);
// or
setObject(previousObject => { ...previousObject, name: "tata" });
// or
setObject(previousObject => { ...previousObject, isSingle: true });
// or
setObject(previousObject => { ...previousObject,  });
```