



# Immuabilité

# Immuabilité - Introduction



- Lors de la modification d'un objet, le changement peut-être muable en modifiant l'objet d'origine ou immuable en créant un nouvel objet
- Les algorithmes de détections de changements préféreront les changements immuables, ayant ainsi juste à comparer les références plutôt que l'ensemble du contenu de l'objet
- Exemple, en JS les tableaux sont muables, les chaines de caractères immuables

```
const firstName = 'Romain';  
firstName.concat('Edouard');  
console.log(firstName); // Romain  
  
const firstNames = ['Romain'];  
firstNames.push('Edouard');  
console.log(firstNames.join(', ')); // Romain, Edouard
```



- Ajouter à la fin

```
const firstNames = ['Romain', 'Edouard'];

function append(array, value) {
  return [...array, value];
}

const newfirstNames = append(firstNames, 'Jean');
console.log(newfirstNames.join(', ')); // Romain, Edouard, Jean
console.log(firstNames === newfirstNames); // false
```

- Ajouter au début

```
const firstNames = ['Romain', 'Edouard'];

function prepend(array, value) {
  return [value, ...array];
}

const newfirstNames = prepend(firstNames, 'Jean');
console.log(newfirstNames.join(', ')); // Jean, Romain, Edouard
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Tableaux



- Ajouter à un indice donné

```
const firstNames = ['Romain', 'Edouard'];

function insertAt(array, value, i) {
  return [
    ...array.slice(0, i),
    value,
    ...array.slice(i),
  ];
}

const newfirstNames = insertAt(firstNames, 'Jean', 1);
console.log(newfirstNames.join(', ')); // Romain, Jean, Edouard
console.log(firstNames === newfirstNames); // false
```



- Modifier un élément

```
const firstNames = ['Romain', 'Edouard'];

function modify(array, value, i) {
  return [
    ...array.slice(0, i),
    value,
    ...array.slice(i + 1),
  ];
}

const newfirstNames = modify(firstNames, 'Jean', 1);
console.log(newfirstNames.join(', ')); // Romain, Jean
console.log(firstNames === newfirstNames); // false
```



- Supprimer un élément

```
const firstNames = ['Romain', 'Edouard'];

function remove(array, i) {
  return [
    ...array.slice(0, i),
    ...array.slice(i + 1),
  ];
}

const newfirstNames = remove(firstNames, 1);
console.log(newfirstNames.join(', ')); // Romain
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Objet



- Ajouter un élément

```
const contact = {
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
};

function add(object, key, value) {
  return {
    ...object,
    [key]: value,
  };
}

const newContact = add(contact, 'city', 'Paris');
console.log(JSON.stringify(newContact));
// {"firstName":"Romain","lastName":"Bohdanowicz","city":"Paris"}
console.log(contact === newContact); // false
```



- Modifier un élément

```
const contact = {
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
};

function modify(object, key, value) {
  return {
    ...object,
    [key]: value,
  };
}

const newContact = modify(contact, 'firstName', 'Thomas');
console.log(JSON.stringify(newContact));
// {"firstName":"Thomas","lastName":"Bohdanowicz"}
console.log(contact === newContact); // false
```





- Supprimer un élément

```
const contact = {
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
};

function remove(object, key) {
  const { [key]: val, ...rest } = object;
  return rest;
}

const newContact = remove(contact, 'lastName');
console.log(JSON.stringify(newContact));
// {"firstName":"Romain"}
console.log(contact === newContact); // false
```

# Immuabilité - Immutable.js



- Pour simplifier la manipulation d'objets ou de tableaux immuables, Facebook a créé Immutable.js
- Installation  
`npm install immutable`

# Immuabilité - Immutable.js List



- Ajouter à la fin

```
const immutable = require('immutable');  
  
const firstNames = immutable.List(['Romain', 'Edouard']);  
  
const newfirstNames = firstNames.push('Jean');  
console.log(newfirstNames.join(', ')); // Romain, Edouard, Jean  
console.log(firstNames === newfirstNames); // false
```

- Ajouter au début

```
const immutable = require('immutable');  
  
const firstNames = immutable.List(['Romain', 'Edouard']);  
  
const newfirstNames = firstNames.unshift('Jean');  
console.log(newfirstNames.join(', ')); // Jean, Romain, Edouard  
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Immutable.js List



- Ajouter à un indice donné

```
const immutable = require('immutable');  
  
const firstNames = immutable.List(['Romain', 'Edouard']);  
  
const newfirstNames = firstNames.insert(1, 'Jean');  
console.log(newfirstNames.join(', ')); // Romain, Jean, Edouard  
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Immutable.js List



- Modifier un élément

```
const immutable = require('immutable');  
  
const firstNames = immutable.List(['Romain', 'Edouard']);  
  
const newfirstNames = firstNames.set(1, 'Jean');  
console.log(newfirstNames.join(', ')); // Romain, Jean  
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Immutable.js List



- Supprimer un élément

```
const immutable = require('immutable');  
  
const firstNames = immutable.List(['Romain', 'Edouard']);  
  
const newfirstNames = firstNames.delete(1);  
console.log(newfirstNames.join(', ')); // Romain  
console.log(firstNames === newfirstNames); // false
```

# Immuabilité - Immutable.js Map



- Ajouter un élément

```
const immutable = require('immutable');

const contact = immutable.Map({
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
});

const newContact = contact.set('city', 'Paris');
console.log(JSON.stringify(newContact));
// {"firstName":"Romain","lastName":"Bohdanowicz","city":"Paris"}
console.log(contact === newContact); // false
```

# Immuabilité - Immutable.js Map



- Modifier un élément

```
const immutable = require('immutable');

const contact = immutable.Map({
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
});

const newContact = contact.set('firstName', 'Thomas');
console.log(JSON.stringify(newContact));
// {"firstName":"Thomas","lastName":"Bohdanowicz"}
console.log(contact === newContact); // false
```



# Immuabilité - Immutable.js Map



- Supprimer un élément

```
const immutable = require('immutable');

const contact = immutable.Map({
  firstName: 'Romain',
  lastName: 'Bohdanowicz',
});

const newContact = contact.remove('lastName');
console.log(JSON.stringify(newContact));
// {"firstName":"Romain"}
console.log(contact === newContact); // false
```