

Pure function & Hooks

Journey from "Pure function" to
"Hooks"





Jong

- Jongeun Lee
- Organizer of JavaScript Developer Forum Korea (jsdev.kr)
- Fullstack Developer
- JavaScript ❤️
- react-native, redux, parse-server, terraform, AWS... 🔍

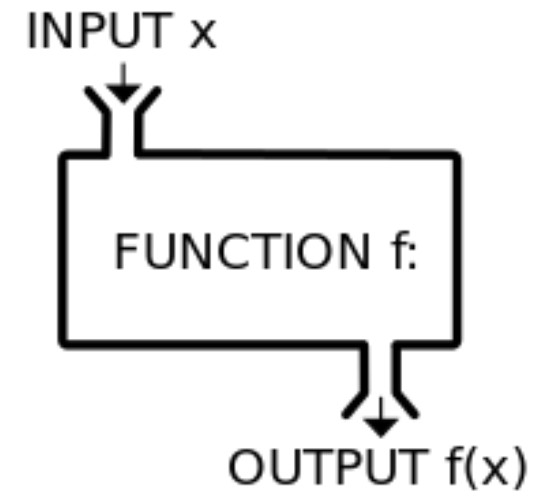
From

to



What is

🤔 **Function?**



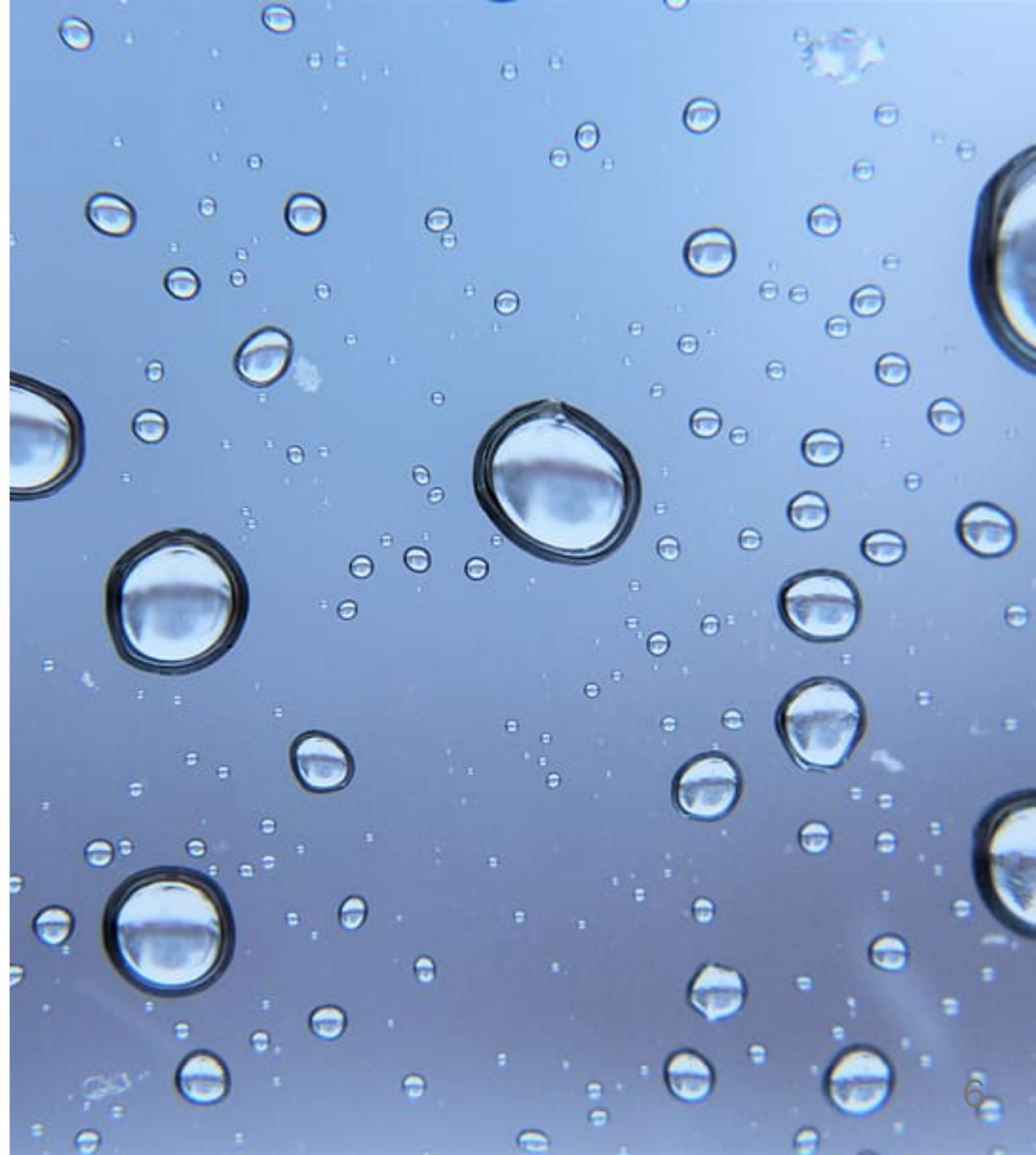
Function

```
function add(a, b) {  
  return a + b;  
}  
add(1, 2);
```

- Input (arguments)
- Output (return value)

What is

Pure Function?



Pure Function

👉 Same input, same output.

👉 NO side effect

Why Pure Function?

 predictable

If

👉 Same input, same output.

✌️ NO side effect

it is

👍 predictable

1. Pure?

```
let c = 0;  
function add(a, b) {  
  return a + b + c;  
}
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

1. Pure?

```
let c = 0;  
function add(a, b) {  
  return a + b + c;  
}  
add(1, 2);  
c = 3;  
add(1, 2);
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

1-1. Pure!

```
function add(a, b, c) {  
  return a + b + c;  
}
```

```
//-----
```

```
add(1, 2, 3);  
add(1, 2, 4);  
add(1, 2, 3);
```

👉 Same input, same output.

✌ NO side effect

👍 predictable

2. Pure?

```
let globals = {  
  count: 0  
};  
function add(a, b) {  
  globals.count++;  
  return a + b;  
}
```

```
//-----
```

```
add(1, 2); // 3  
add(1, 2); // 3
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

2. Pure?

```
let globals = {  
  count: 0  
};  
function add(a, b) {  
  globals.count++;  
  return a + b;  
}
```

//-----

```
add(1, 2); // 3  
globals = undefined;  
add(1, 2); // ??
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

2-1. Pure!

```
function add(a, b, count) {  
  return {  
    value: a + b,  
    count: count + 1  
  };  
}  
  
//-----  
  
let myCount = 0;  
const result = add(1, 2, myCount);  
myCount = result.count;
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

2-2. Pure!

```
function add(a, b, callback) {  
  callback();  
  return a + b;  
}
```

```
//-----
```

```
let count = 0;  
add(1, 2, () => {  
  count++;  
}); // 3
```

👉 Same input, same output.

✌ NO side effect

👍 predictable

3. Pure?

```
function add(args) {  
  const result = args.left + args.right;  
  args.left = 0;  
  return result;  
}  
  
add({  
  left: 1,  
  right: 2  
});
```

👉 Same input, same output.


✌️ NO side effect

👍 predictable

If you want to make PURE function

- 👉 Make an output value using input value.
- ✌️ Do NOT change anything existed in outside of function.

| For React Developer

 **Why understanding pure function is important?**

React Functional programming

- Pure function
- Declarative

Evidence 1.

**Conceptually,
components are like
JavaScript functions.**

reactjs.org/docs/components-and-props.html

Conceptually,

Is this **Button** a function?

```
<Button title="Press Me" color="green" onPress={this.onPress} />
```

Function is "using _____, make _____"


```
<Button title="Press Me" color="green" onPress={this.onPress} />
```

input	output
title, color, onPress	<div>Press Me</div>

Evidence 2.

**All React components
must act like *pure functions*
with respect to their props.**

reactjs.org/docs/components-and-props.html

Pure Function?

```
const MyH1 = ({ title }) => {  
  return <h1 style={{ color: 'red' }}>{title}</h1>;  
};
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

PURE!

same **props** makes same UI

```
<MyH1 title="I love Bali"/>  
<MyH1 title="I love Bali"/>
```

I love Bali

I love Bali

| pure function

Pros.

1. Test

2. Performance

Because it is predictable.



Output?

```
function specialCrazyCalculator(a, b) {  
    return Math.floor(Math.asin((a + b) / (a + b + 1) / 10) * 123.44566834);  
}  
specialCrazyCalculator(1, 2);
```

Speed quiz

What is return value?

```
specialCrazyCalculator(1, 2);
```


Test code

```
test('specialCrazyCalculator', () => {  
  expect(specialCrazyCalculator(1, 2)).toBe(9);  
});
```

| pure function

Pros.

1. Test

2. Performance

Because it is predictable.



| pure function

Cons.

**Can not remember anything
by itself**

| In other words, It doesn't have
state.



CounterButton Example



Which one has to count?

1. CounterButton
2. Parent component of CounterButton

**Move `count` state
to `CounterButton` component**

Hook



We don't know what is `useState` yet.

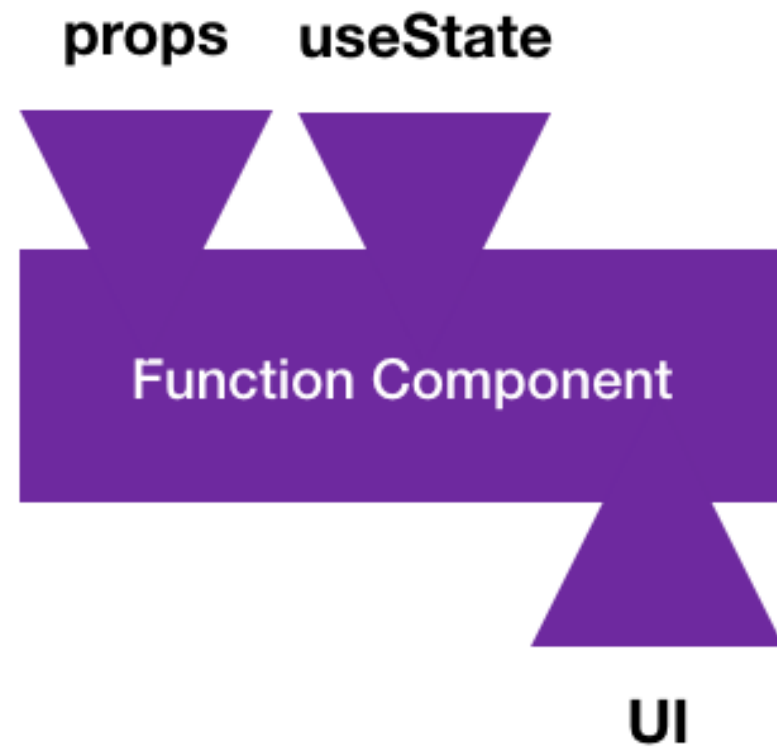
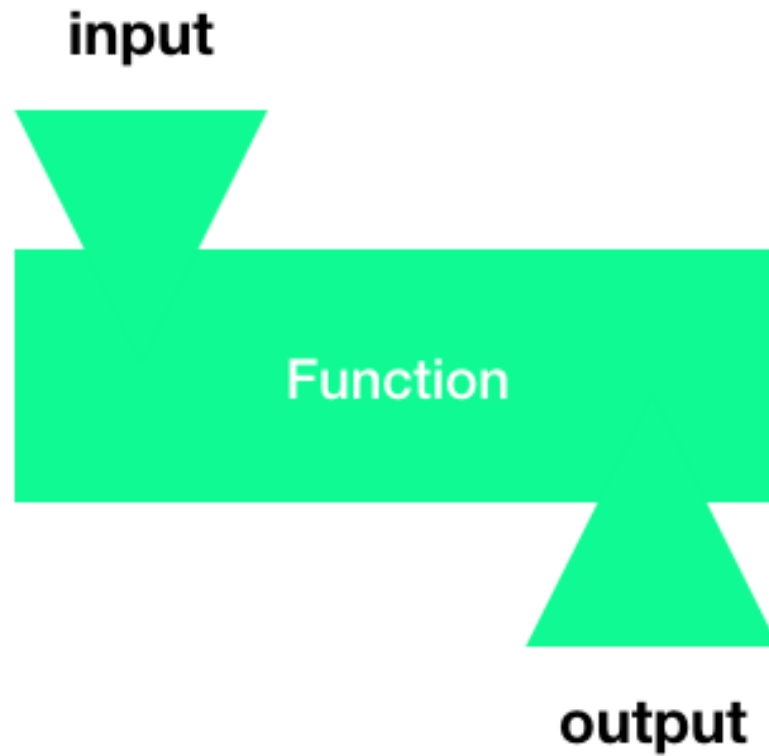
Try `useState`


Pure or Impure

```
const CounterButton = ({ title }) => {  
  const [count, setCount] = useState(0);  
  return (  
    <button  
      onClick={() => {  
        setCount(count + 1);  
      }}  
    >  
      {title} : {count}  
    </button>  
  );  
};
```



But conceptually PURE!



 **useState(hook) is
another input**

3. Pure?

`props`, `useState` are input

```
const CounterButton = ({ title }) => {  
  const [count, setCount] = useState(0);  
  return (  
    <button  
      onClick={() => {  
        setCount(count + 1);  
      }}  
    >  
      {title} : {count}  
    </button>  
  );  
};
```

👉 Same input, same output.

✌️ NO side effect

👍 predictable

Wanna *Remember* something and *Re-render* when it changes?

```
const [count, setCount] = useState(0)
```

- `[0]` : Stored value or initial value. you can NOT change it directly.
- `[1]` : A Function that changes the value and triggers re-rendering



Wanna make some *changes outside of function?*

`useEffect()`

- After rendering, do *something*
- For example
 - `setTimeout`, `setInterval`
 - `subscribe`
 - data fetching
 - imperative API



More built-in hooks

- useContext
- useReducer
- useCallback
- useMemo
- ...

🖱️ LIVE CODING

Custom Hook

| In other words, Custom Input





Summary

Pure Function

 Same input, same output.

 NO side effect

 predictable

Props and
useState are
input of
*Function
Component*



Props : input
from *Parent*
Component

useState : input
from *Hook*





Terima kasih banyak

twitter / GitHub : @yomybaby

Refs

- Images
 - function : [https://en.wikipedia.org/wiki/Function_\(mathematics\)#/media/File:Function_machine2.svg](https://en.wikipedia.org/wiki/Function_(mathematics)#/media/File:Function_machine2.svg)
 - Pure water : <https://www.piqsels.com/en/public-domain-photo-jfecp>
 - Hook Lego : <https://www.amazon.com/LEGO-Disney-Collectible-Minifigure-Captain/dp/B01F1FNLWC>
 - typing gif : <https://media.giphy.com/media/ICbSAbRrFEfkY/giphy.gif>
 - Other images (except emoji) by [Jong Eun Lee](#) under CC0 License