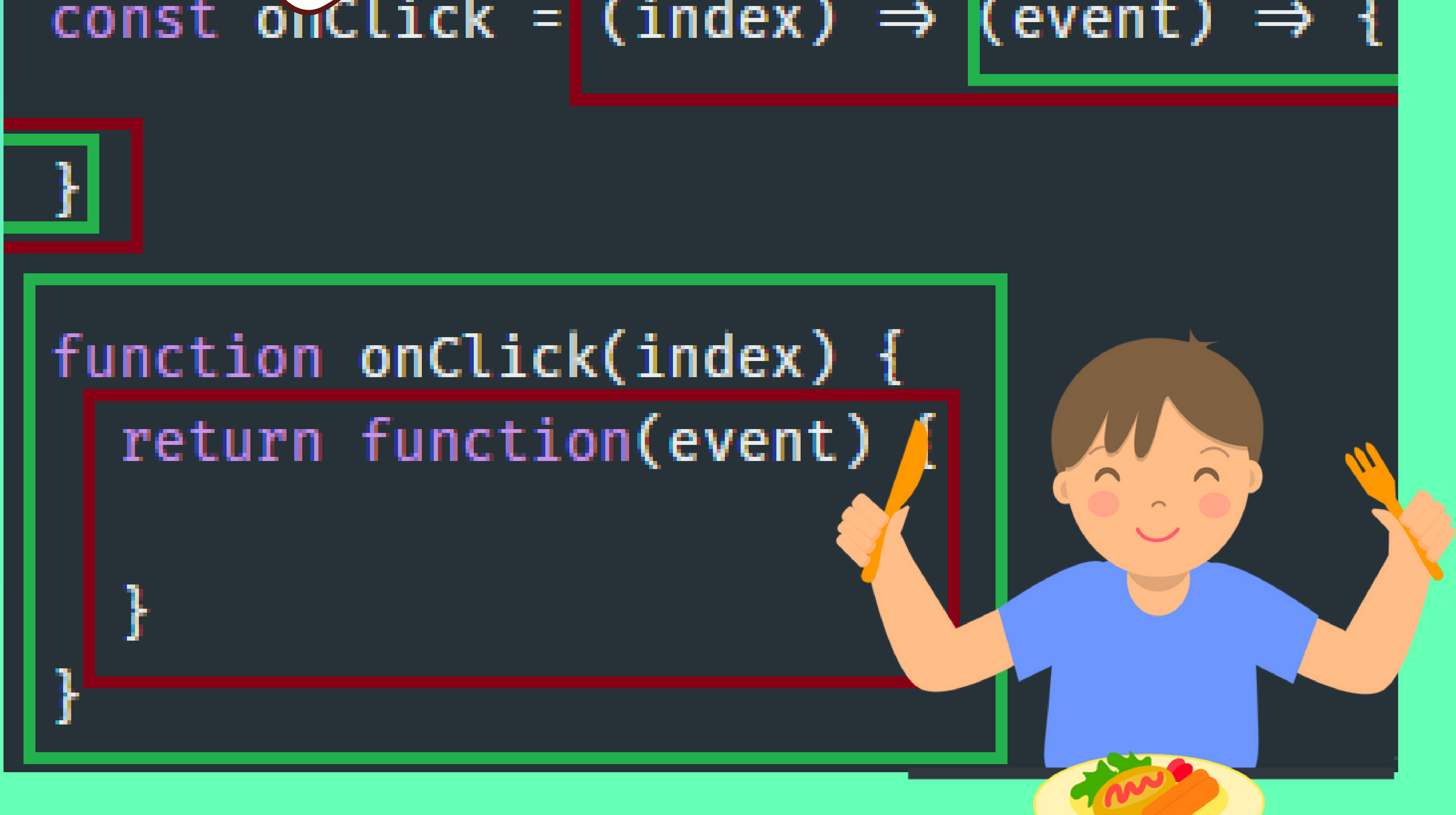
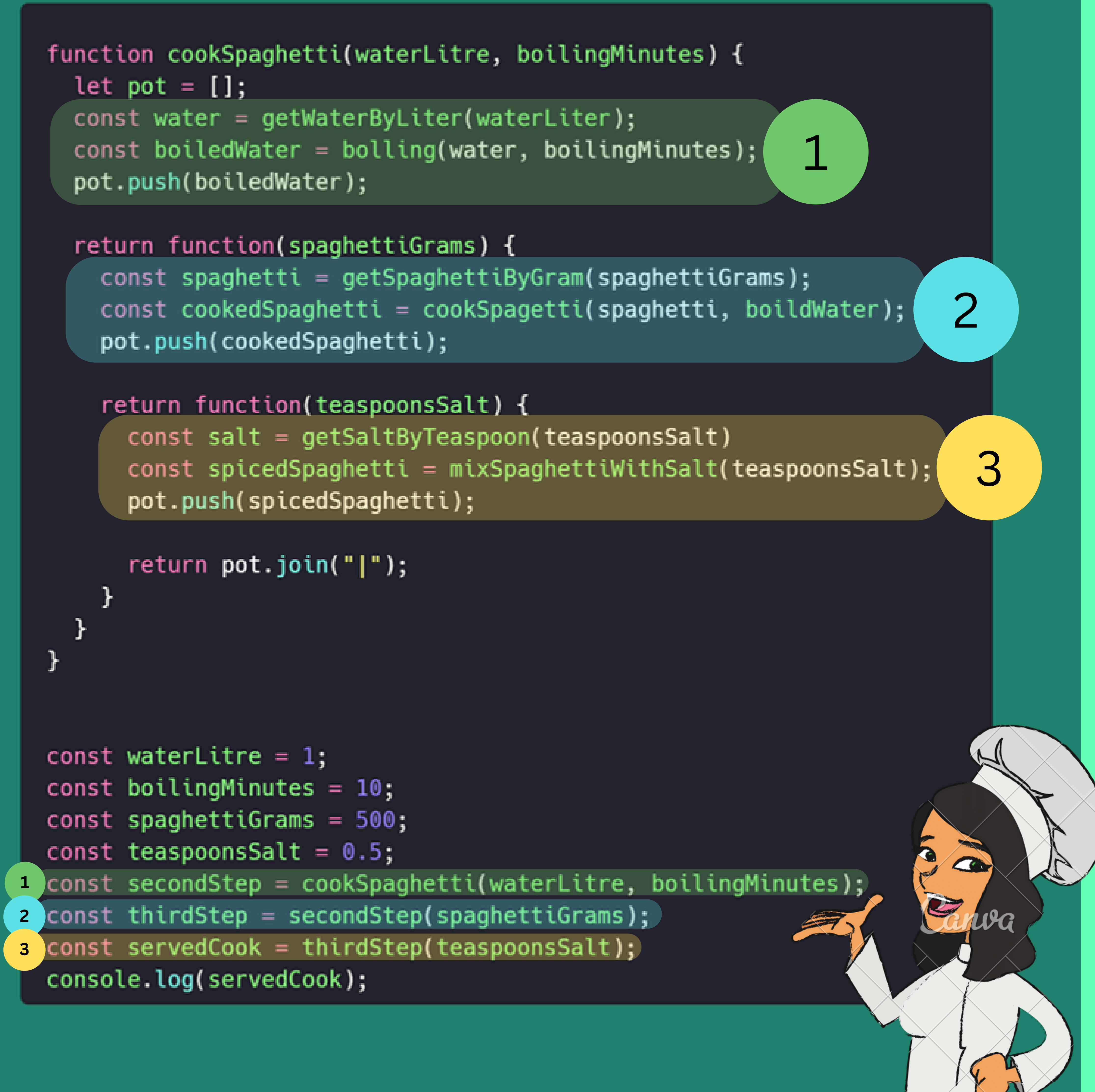
Currying in JavaScript



```
const onClick = (index) => (event) => {

}

function onClick(index) {
  return function(event) {

  }
}
```

“Currying” is a way to mix something to get a result. Imagine you have to cook food inside a pot. Some ingredient is added to the pot step by step, and as a result, we have food which is cooked by adding ingredient.



```
function cookSpaghetti(waterLitre, boilingMinutes) {
  let pot = [];
  const water = getWaterByLiter(waterLiter);
  const boiledWater = bolling(water, boilingMinutes);
  pot.push(boiledWater);

  return function(spaghettiGrams) {
    const spaghetti = getSpaghettiByGram(spaghettiGrams);
    const cookedSpaghetti = cookSpagetti(spaghetti, boildWater);
    pot.push(cookedSpaghetti);

    return function(teaspoonsSalt) {
      const salt = getSaltByTeaspoon(teaspoonsSalt)
      const spicedSpaghetti = mixSpaghettiWithSalt(teaspoonsSalt);
      pot.push(spicedSpaghetti);

      return pot.join("|");
    }
  }
}


const waterLitre = 1;
const boilingMinutes = 10;
const spaghettiGrams = 500;
const teaspoonsSalt = 0.5;
const secondStep = cookSpaghetti(waterLitre, boilingMinutes);
const thirdStep = secondStep(spaghettiGrams);
const servedCook = thirdStep(teaspoonsSalt);
console.log(servedCook);
```

Canva

Three key points

1 Encapsulation

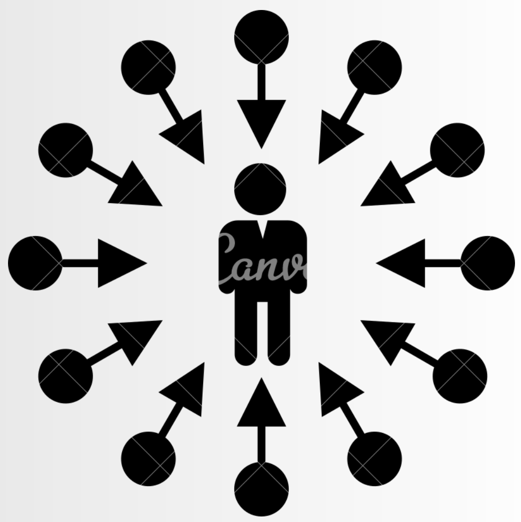
Data not be visible out of function only returned object is shown

2 Sequenceable

Every function returns another function, until the last returned object

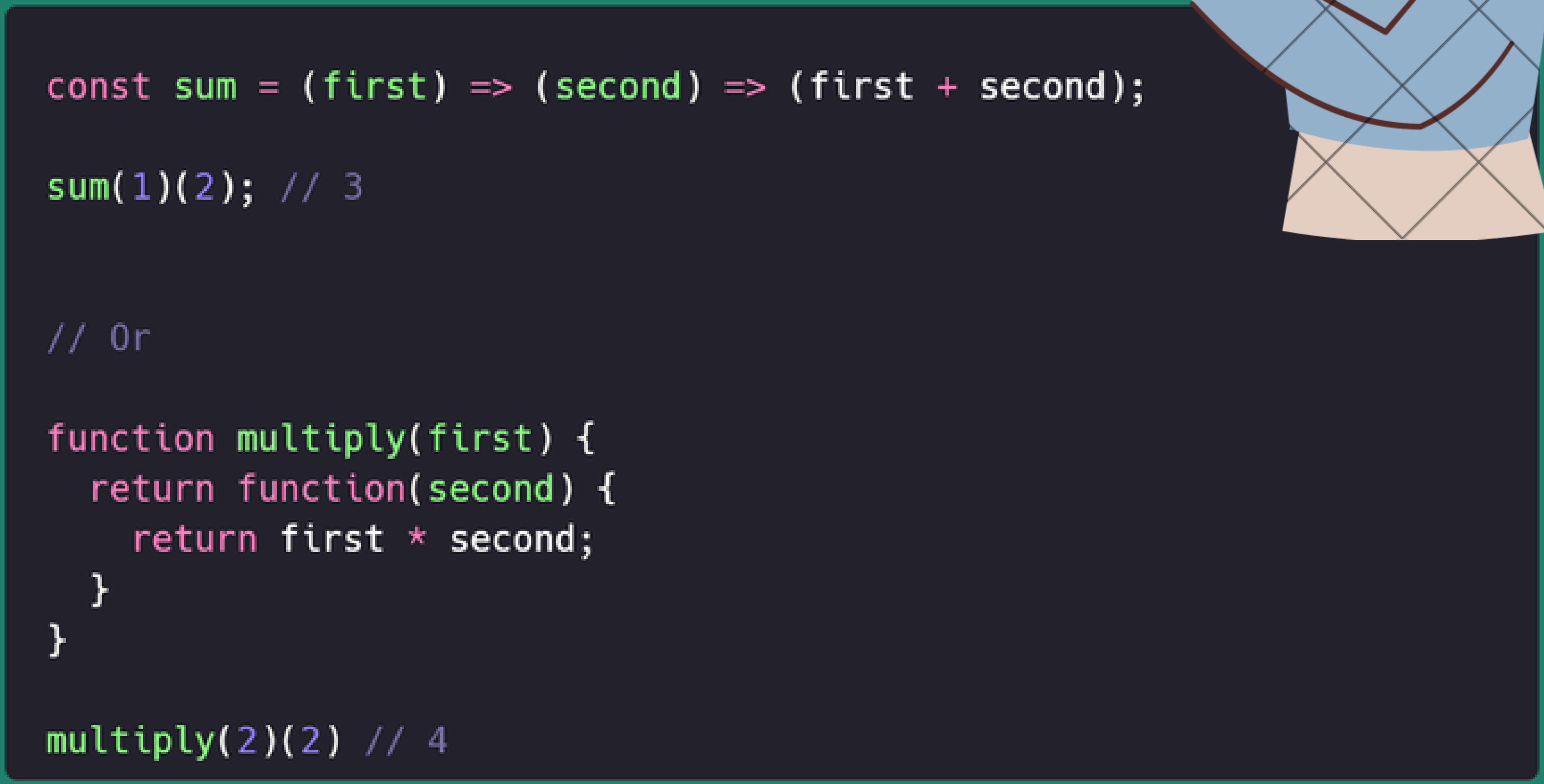
3 centralisation

The main goal is to have a main result and computing be effected on the next function



A real example!

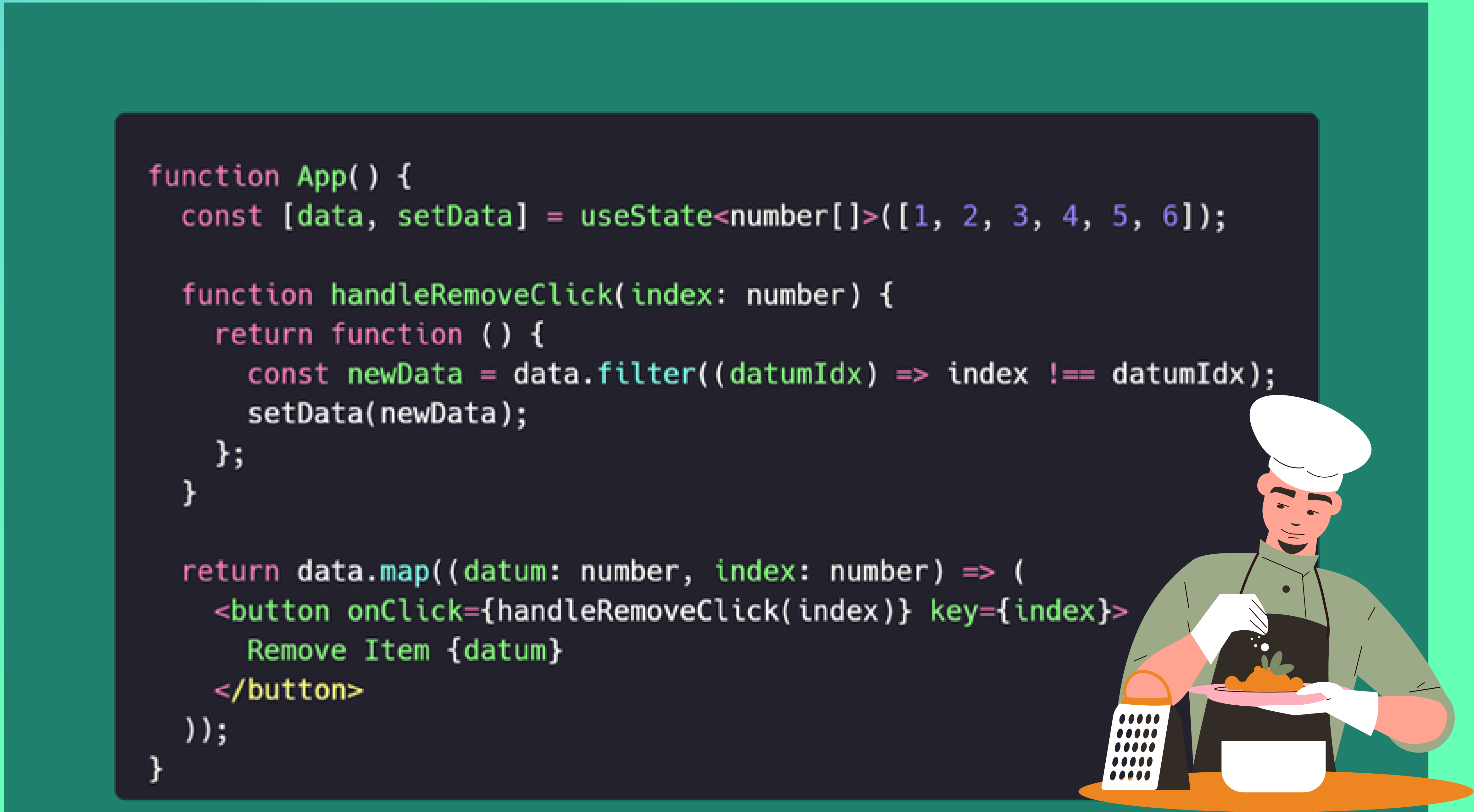
Also we can use **arrow function!**



```
const sum = (first) => (second) => (first + second);

sum(1)(2); // 3


// Or

function multiply(first) {
  return function(second) {
    return first * second;
  }
}

multiply(2)(2) // 4
```

A React Example

To remove an item from data, the index immediately passed and stored in function after button is clicked, another method is called.



```
function App() {
  const [data, setData] = useState<number[]>([1, 2, 3, 4, 5, 6]);

  function handleRemoveClick(index: number) {
    return function () {
      const newData = data.filter((datumIdx) => index !== datumIdx);
      setData(newData);
    };
  }

  return data.map((datum: number, index: number) => (
    <button onClick={handleRemoveClick(index)} key={index}>
      Remove Item {datum}
    </button>
  ));
}
```

Binding Vs Currying

Currying simplifies a function by breaking it down into multiple functions. bind() is a built-in JavaScript function that helps us to pass instance and currying together.