call(), apply() bind()

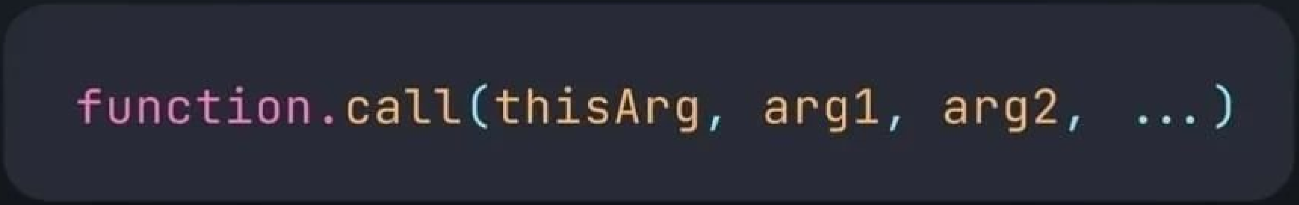
Functions | In Detail

Mallikarjun | @CodeBustler

call() Function

- The **call** method is used to invoke a function with a specified **“this“** "value and a list of arguments (parameters).

- **Syntax**:

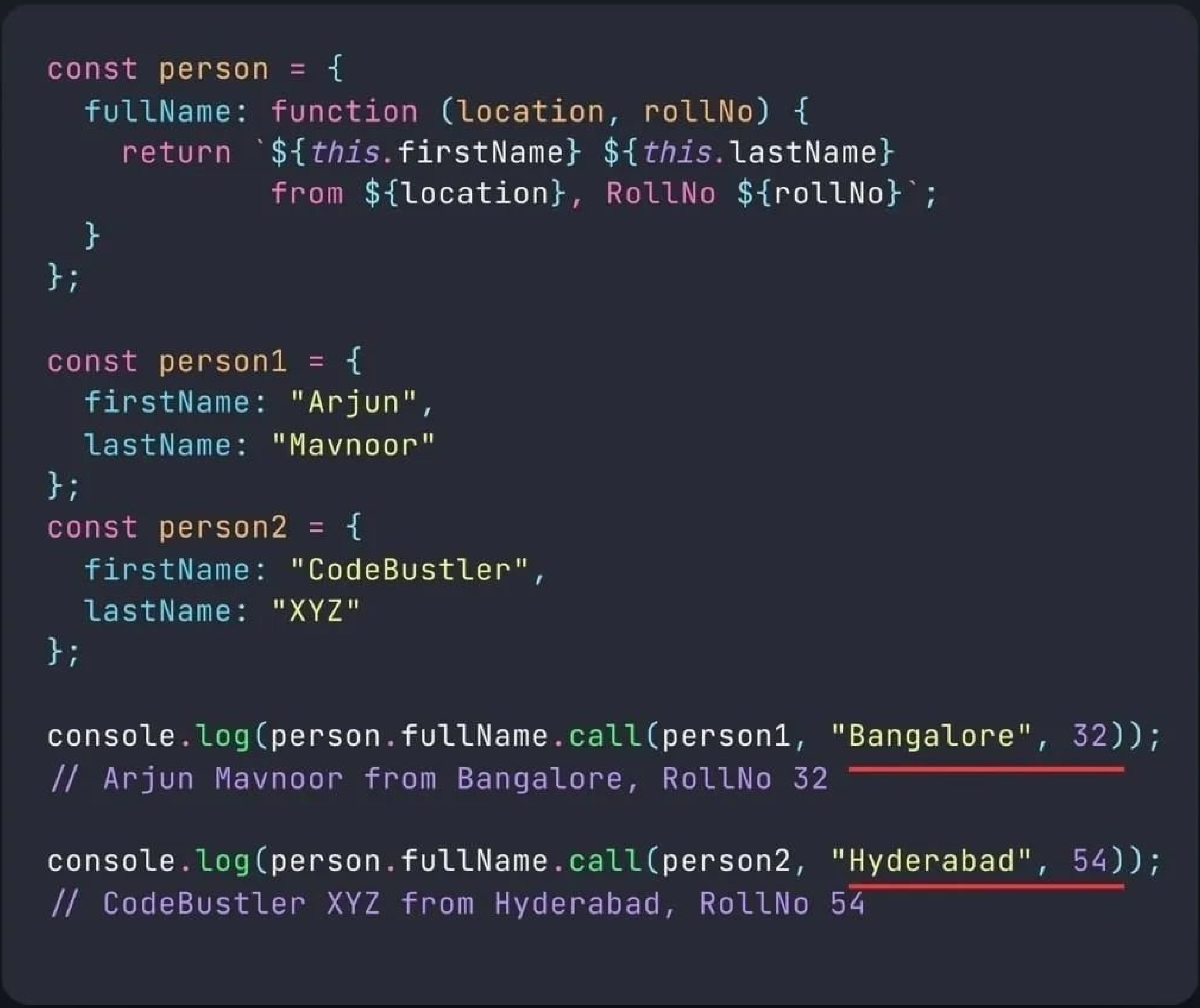


```
function.call(thisArg, arg1, arg2, ...)
```

The first argument (**thisArg**) specifies the value of "**this**" within the function when it is invoked

@CodeBustler

call() Function | Example



```javascript
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
            from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};
const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

console.log(person.fullName.call(person1, "Bangalore", 32));
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(person.fullName.call(person2, "Hyderabad", 54));
// CodeBustler XYZ from Hyderabad, RollNo 54
```

@CodeBustler

apply() Function

- The apply method is similar to call, but it takes an array-like object of arguments instead of individual arguments.

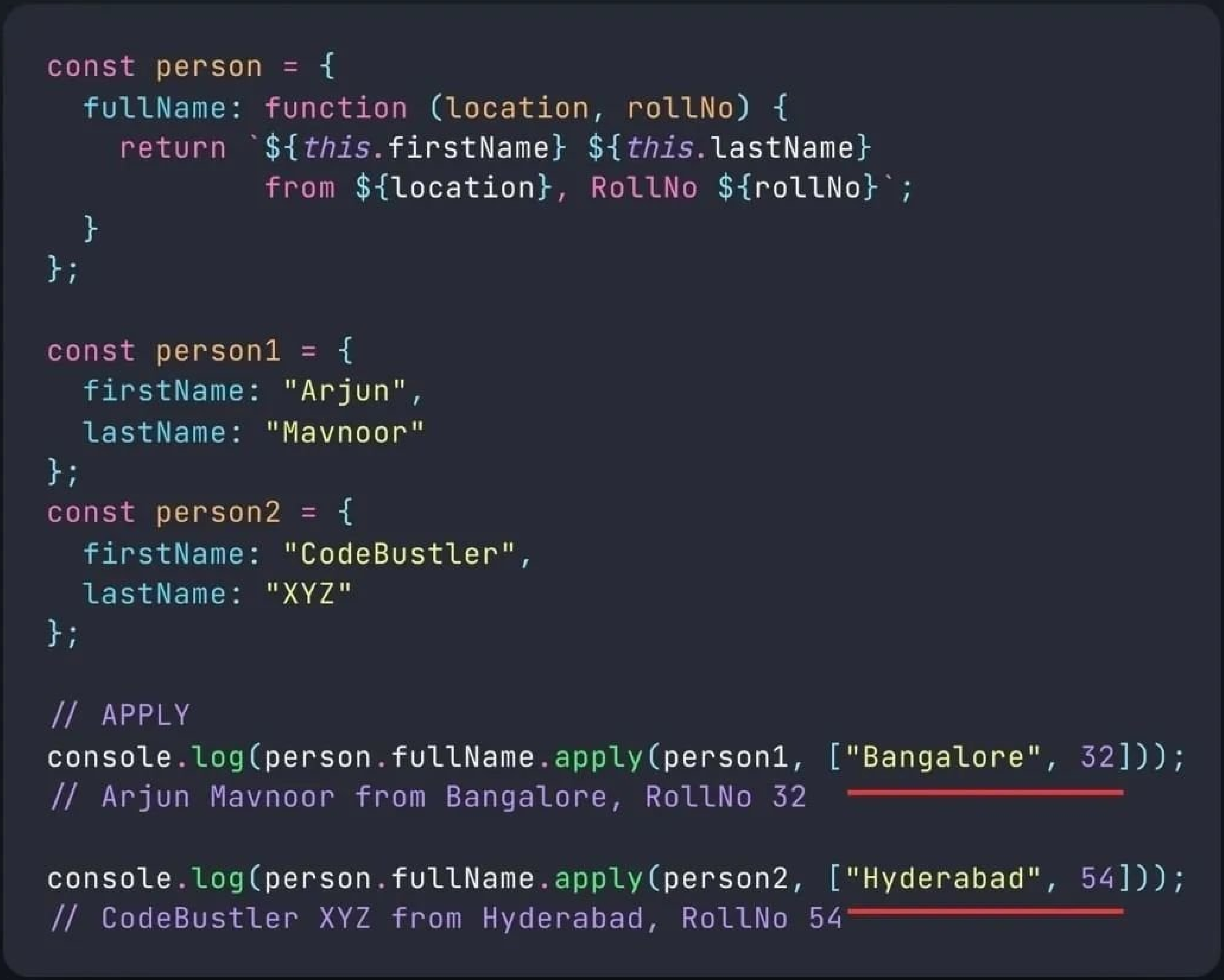
- **Syntax :**

```
function.apply(thisArg, [arg1, arg2, ...])
```

array-like object
of arguments

@CodeBustler

apply() Function | Example



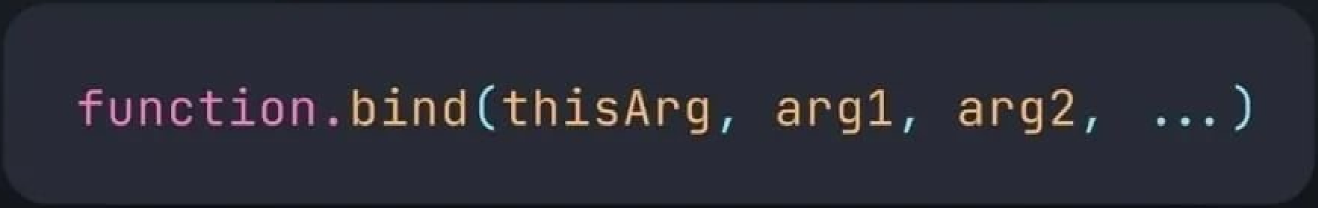
```javascript
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
            from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};
const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

// APPLY
console.log(person.fullName.apply(person1, ["Bangalore", 32]));
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(person.fullName.apply(person2, ["Hyderabad", 54]));
// CodeBustler XYZ from Hyderabad, RollNo 54
```

@CodeBustler

bind() Function

- The bind method is used to create a new function that, when invoked, has a specified this value and optional arguments.

- **Syntax :**

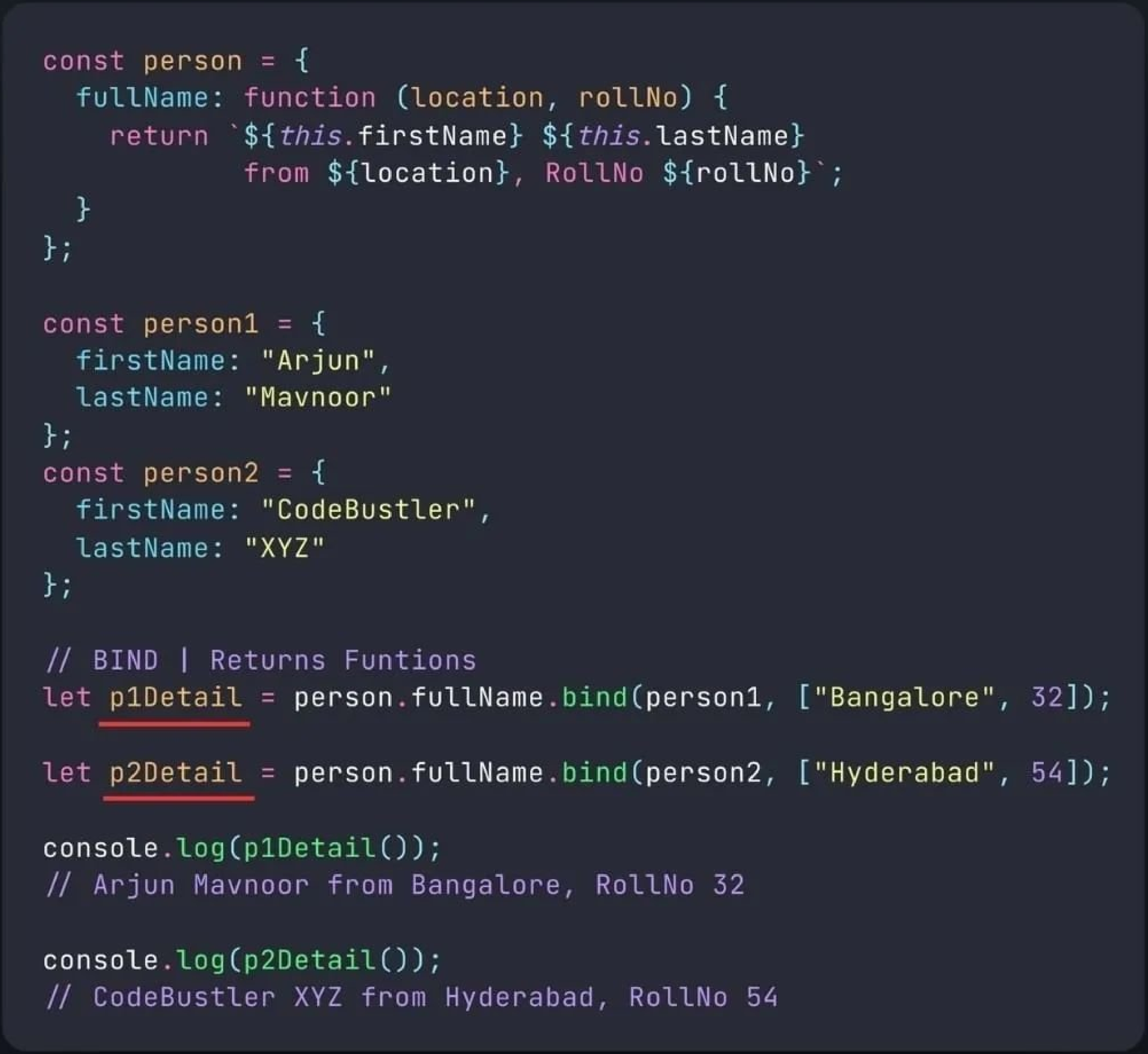


```
function.bind(thisArg, arg1, arg2, ...)
```

create a new
function

@CodeBustler

bind() Function | Example



```
const person = {
  fullName: function (location, rollNo) {
    return `${this.firstName} ${this.lastName}
            from ${location}, RollNo ${rollNo}`;
  }
};

const person1 = {
  firstName: "Arjun",
  lastName: "Mavnoor"
};
const person2 = {
  firstName: "CodeBustler",
  lastName: "XYZ"
};

// BIND | Returns Funtions
let p1Detail = person.fullName.bind(person1, ["Bangalore", 32]);

let p2Detail = person.fullName.bind(person2, ["Hyderabad", 54]);

console.log(p1Detail());
// Arjun Mavnoor from Bangalore, RollNo 32

console.log(p2Detail());
// CodeBustler XYZ from Hyderabad, RollNo 54
```

@CodeBustler

Important bullet points

- **call** and **apply** set **“this”** and pass arguments to a function;
  - **call** uses **individual args**,
  - **apply** uses **an array of args**.

- **bind** creates a **new function** with a fixed this & optional args, without immediate execution.

- **call** and **apply** invoke functions **immediately**;
  bind allows delayed execution.

@CodeBustler