

Improving Readability with Arrow Functions



PRATEEK PAREKH

SENIOR STAFF ENGINEER

@ prparekh83



Introduction



Arrow Functions - What and Why

Defining Arrow Functions

Behavior of *this* Keyword

Demo



Introducing Arrow Functions



Arrow Functions

Introduced in ES6

Simpler way to write a function expression

Why to use them?

- shorter syntax
- *this* derives its value from enclosing lexical scope

Side effects

- behavior of this keyword
- No arguments object



Regular Function 1

```
let greetings = function() {  
    return 'Hello World!';  
}  
  
let message = greetings();  
console.log(message); // Hello World!
```



Arrow Function 1

```
let greetings = () => {  
    return 'Hello World!';  
};  
  
let message = greetings();  
console.log(message); // Hello World!
```



Arrow Function 1

```
let greetings = () => 'Hello World!';  
let message = greetings();  
console.log(message); // Hello World!
```



Regular Function 2

```
let greet = function greetings(name) {  
    return 'Hello ' + name;  
}  
  
let message = greet('John');  
console.log(message); // Hello John
```



Arrow Function 2

```
let greetings = name => 'Hello ' + name;
```

```
let message = greetings('John')
```

```
console.log(message); // Hello John
```



Demo



Arrow Function to add 2 numbers



Behavior of *this* Keyword



this

refers to the owner of the function we are executing



Unlike regular functions,
arrow functions do not have
their own *this* value



Demo



this behavior in Arrow Functions



Summary



Arrow Functions

Defining Arrow Functions

Behavior of *this* Keyword

