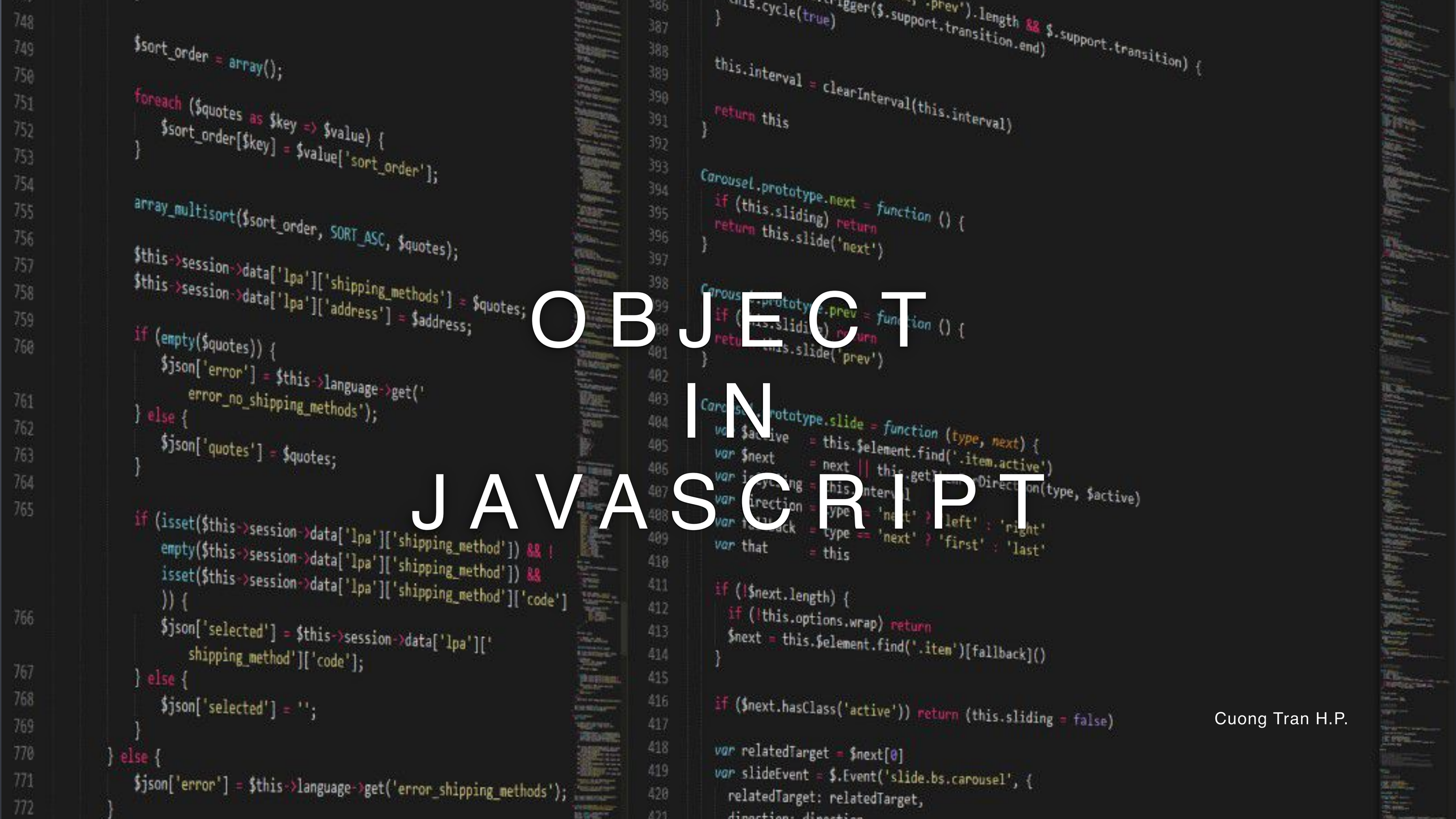


Cuong Tran H.P.

TYPES

TYPES

PRIMITIVE VALUE

Undefined, Null, Boolean,
Number, String and Symbol

REFERENCE VALUE

object, function, array

TYPES

PRIMITIVE VALUE

```
const a = 2;
const b = a;
```

=

```
const a = 2;
const b = 2;
```

REFERENCE VALUE

```
const a = { };
const b = a;
```

≠

```
const a = { };
const b = { };
```

`this` KEYWORD

Most of the time, `this` refers to the object that executes the code

`this` KEYWORD

Global

```
this;
```

The global object

Function call (ES5)

```
myFunc();
```

The global object or
undefined in strict mode

Method call

```
obj.myFunc();
```

The object that executes
the function

Constructor

```
new Obj();
```

The newly created object

Explicitly set (ES5)

```
myFunc.bind(obj);
```

explicitly set to an object,
Can not set for arrow function

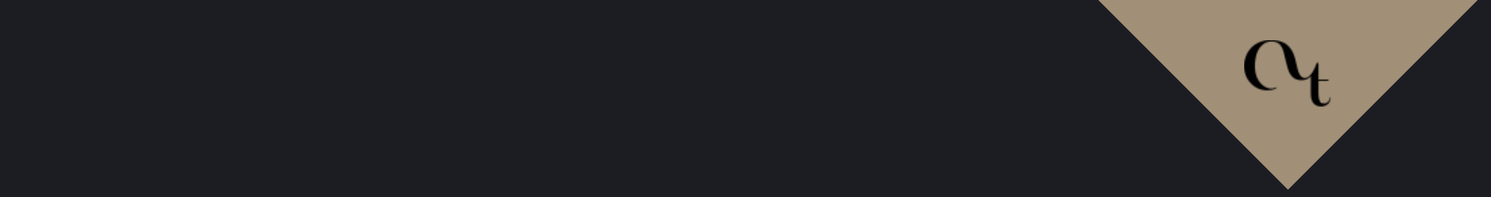
Arrow function

```
() => {}
```

The object outside the
function
Note: should not use as
method

Prototype

Prototype is how inheritance
in javascript works



`__proto__` property

__proto__ property

An object doesn’t have prototype property but it has the __proto__ property. The inheritance object of an object is the __proto__ property.

prototype property

```
Box.prototype.show = function () {
  // ...
}
```

prototype property

```
Box.prototype.show = function () {
  // ...
}
```

The prototype property is created when function is created and only exists in functions. It’s only used for the new keyword

new keyword

```
const data = new Employee();
```

new keyword

```
const data = new Employee();
```

It does 4 things

new keyword

```
const data = new Employee();
```

1 Create a new plain object

It does 4 things

new keyword

```
const data = new Employee();
```

It does 4 things

1. Create a new plain object
2. Assign that object's __proto__ property to the prototype property of the function

`new` keyword

```
const data = new Employee();
```

It does 4 things

1. Create a new plain object
2. Assign that object's `__proto__` property to the `prototype` property of the function
3. Executes the function with `this` binds to the newly created object

`new` keyword

```
const data = new Employee();
```

It does 4 things

1. Create a new plain object
2. Assign that object's `__proto__` property to the `prototype` property of the function
3. Executes the function with `this` binds to the newly created object
4. Return the object

`Class`

`Class` is just an implementation that
builds on top of the prototype

OVERVIEW

- Javascript is prototype-based
- Understand different of primitive value and reference value
- Understand 6 different scope of `this` keyword
- Understand `prototype, __proto__`
- Understand `class` keyword


THANK YOU
FOR LISTENING