CLASSES

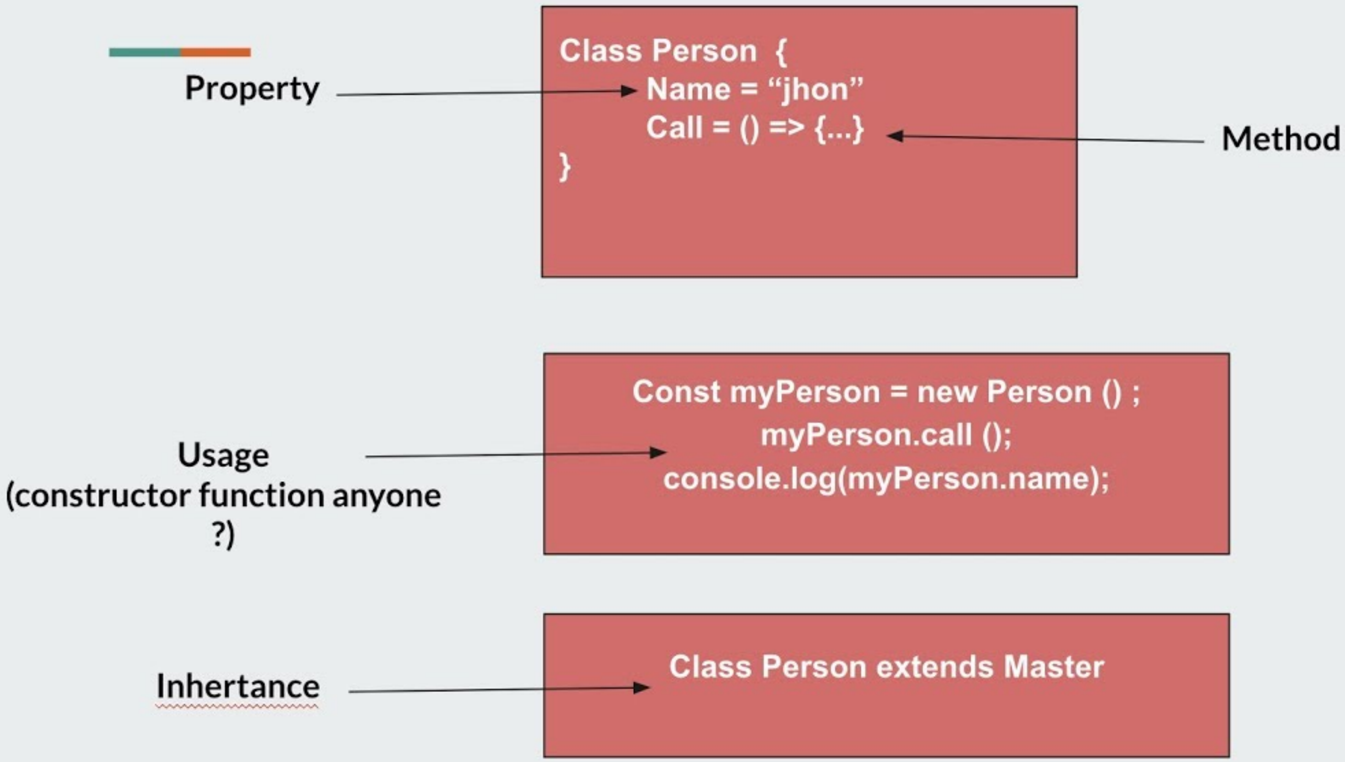
CLASSES ARE...

- a concept in OOP (object oriented programming)
- a template or blueprint for an object
- quite hard to get your head around the first time!

JUST ONE MORE THING...

Classes (in JavaScript) were added in ES6 Before we used prototypes (not covered in this course)

Classes



Property

```
Class Person  {
        Name = “jhon”
        Call = () => {...}
}
```

Method

Usage
(constructor function anyone
?)

```
Const myPerson = new Person () ;
myPerson.call ();
console.log(myPerson.name);
```

Inhertance

```
Class Person extends Master
```

NEW WORDS WE MUST LEARN

- instantiate (verb) - to make a copy of something
- instance (noun) - refers to the copy
- method (noun) - a special function which is attached to an object. Describes some behaviour on that object.

RESTRICTED
AREA
NO UNAUTHORIZED
PERSONNEL
NO
TRESPASSIN

WHAT DOES A CLASS LOOK LIKE IN JAVASCRIPT?

```
class Person {

 }
```

Classes are just like objects - we can store methods and properties on them

In fact, we could represent our classes as objects.

However classes give us a few extra advantages (more on this later)

We don't use classes directly, we must instantiate (copy) them

We copy using the new keyword

```
class Animal {
}

const dog = new Animal();
```

```
const cat = new Animal();
```

```
const horse = new Animal();
```

Naming conventions when writing a class

Class names should be capitalised!

Animal {} ✔

not

animal {} ❌

Instances should NOT be capitalised!

```
const dog = new Animal() ✔
```

not

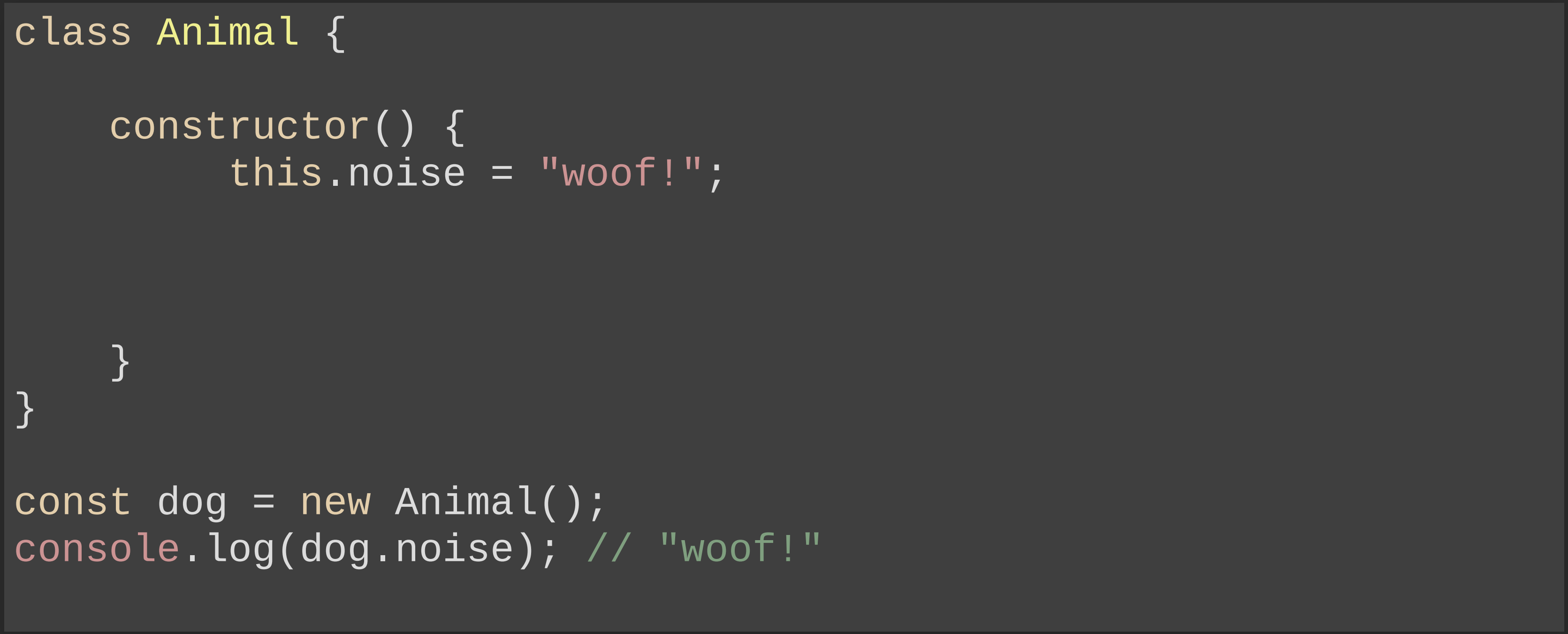
```
const Dog = new Animal() ❌
```

Classes can include a special method called the constructor, which is called when the class is instantiated

```
class Animal {

    constructor() {


    }
}

const dog = new Animal();
```

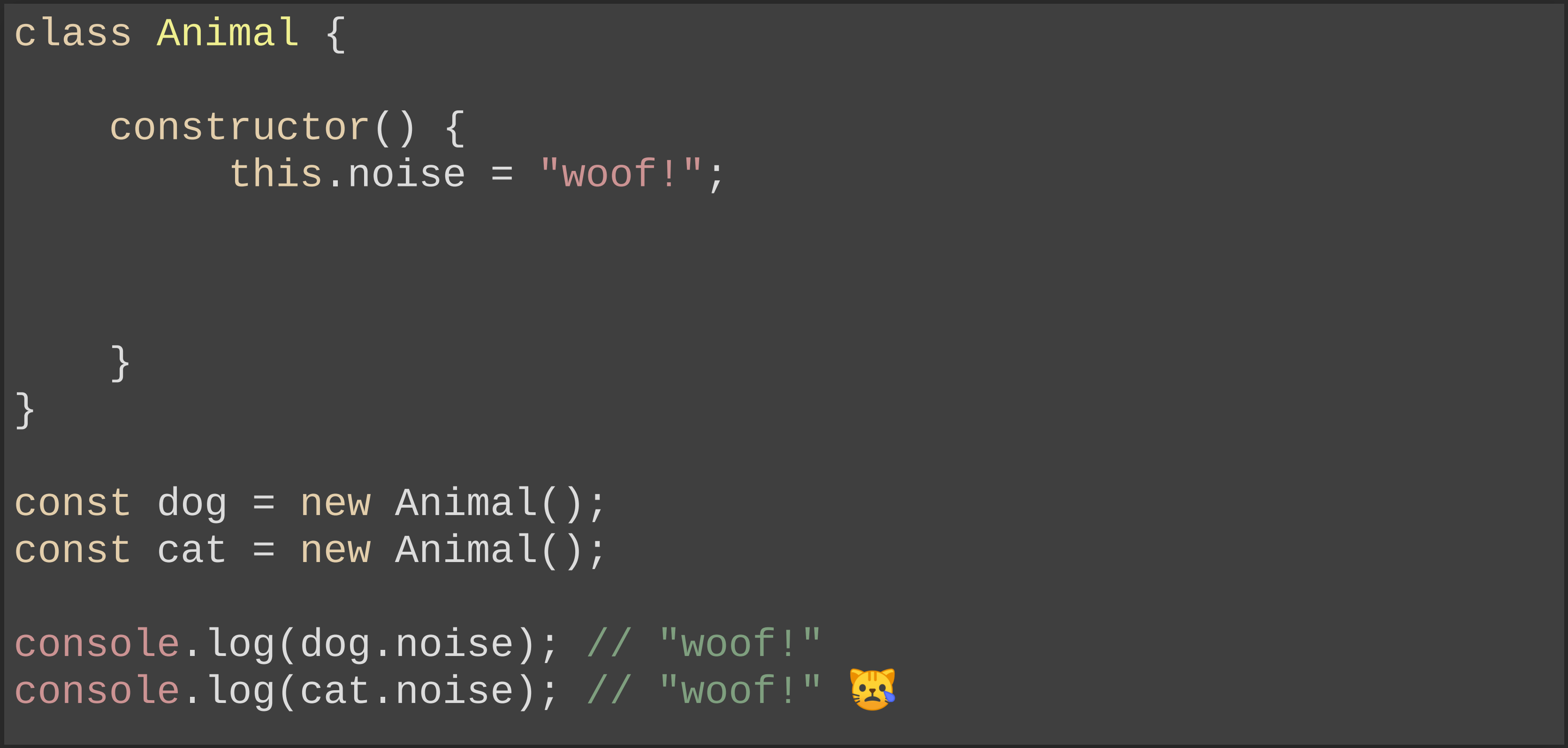
We can also use the constructor to set properties

We must use this to refer to the itself



```
class Animal {

    constructor() {
         this.noise = "woof!";



    }
}

const dog = new Animal();
console.log(dog.noise); // "woof!"
```

constructor is optional!

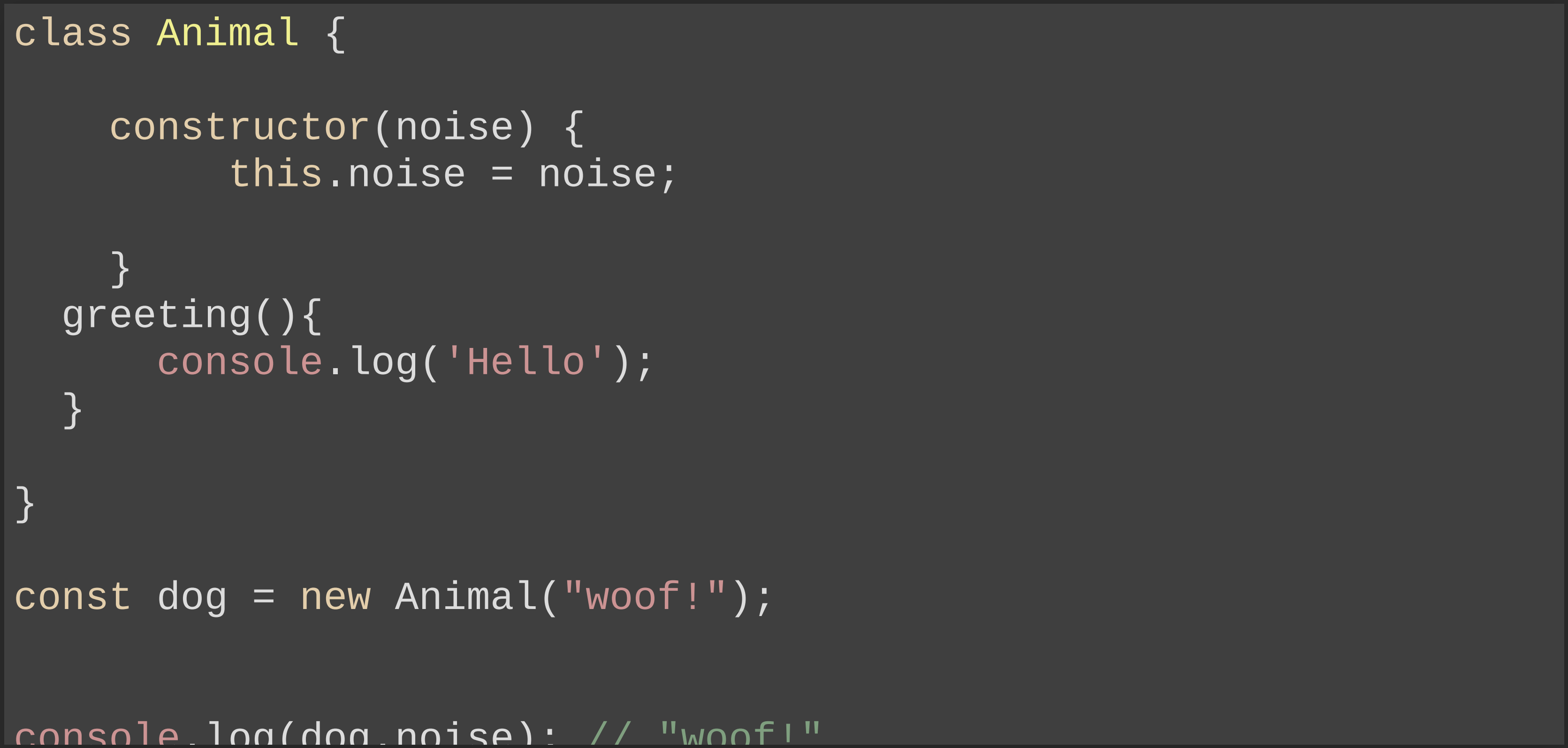


```
class Animal {

    constructor() {
         this.noise = "woof!";



    }
}

const dog = new Animal();
const cat = new Animal();

console.log(dog.noise); // "woof!"
console.log(cat.noise); // "woof!" 😿
```

We can also pass in properties via the constructor
Remember the constructor is basically just a function

```
class Animal {

constructor(noise) {

 this.noise = noise;

}
}
```

LET'S ADD A METHOD NOW



```
class Animal {

    constructor(noise) {
         this.noise = noise;

    }
  greeting(){
      console.log('Hello');
  }

}

const dog = new Animal("woof!");


console.log(dog.noise); // "woof!"
```