Class in JavaScript (ES6)

When we define a function inside a class then we don't need the function key word.

Function defined within the class is called a 'METHOD' in that class.

class Logger {

logMessage(message) {

console.log(message);

}

};

keyword ‘extends’ : comes in ES6

**Script.js**

class Shapes {

logShape(){

console.log(this); //this refers to class

}

static staticLogFunc(){

console.log("This funciton used static keyword");

}

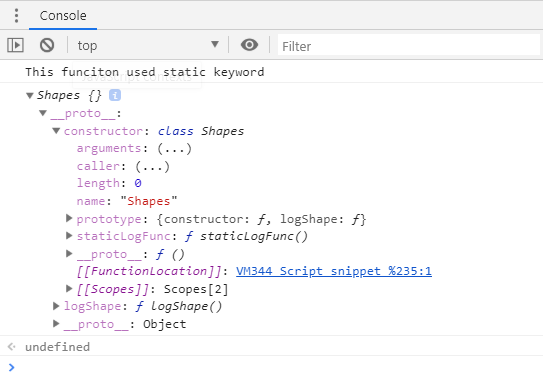
}

Shapes.staticLogFunc();

const myShape = new Shapes();

myShape.logShape();

output:



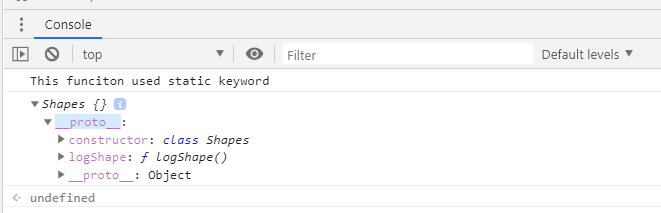
In above console output:

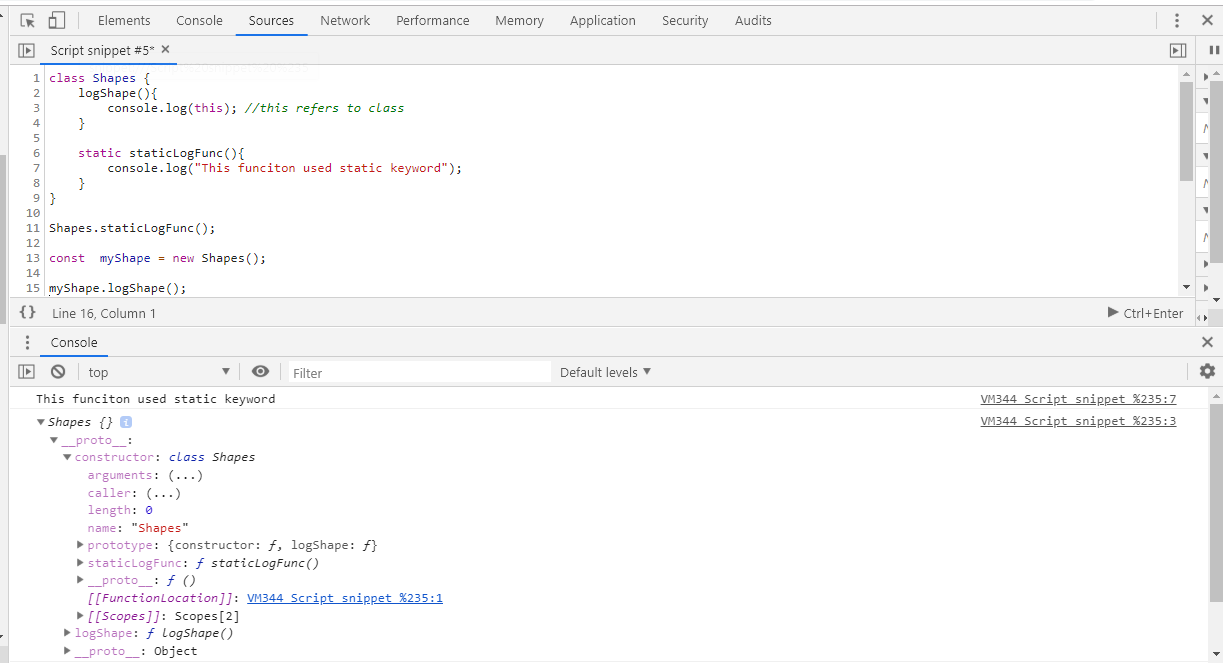
The **static** function staticLogFunc() is shown a part of constructor property of Shapes class object.

While regular function logShape() is showing as property of Shapes class object.

## Static Method:

The static keyword defines a static method for a class. Static methods are called without instantiating their class and cannot be called through a class instance. Static methods are often used to create utility functions for an application.





## Constructor

The constructor method is a special method for creating and initializing an object created with a class. There can only be one special method with the name "constructor" in a class. A SyntaxError will be thrown if the class contains more than one occurrence of a constructor method.

A constructor can use the super keyword to call the constructor of the super class.

class Polygon {

constructor() {

this.name = "Polygon";

}

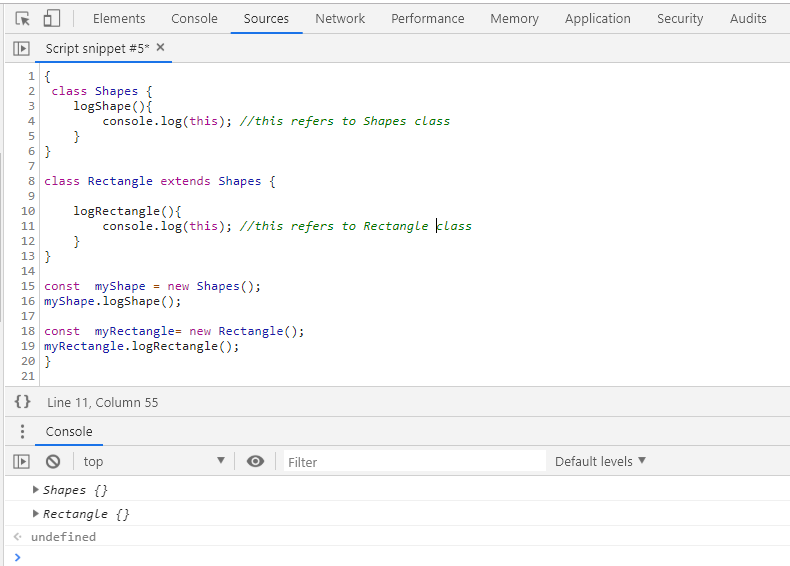
}

var poly1 = new Polygon();

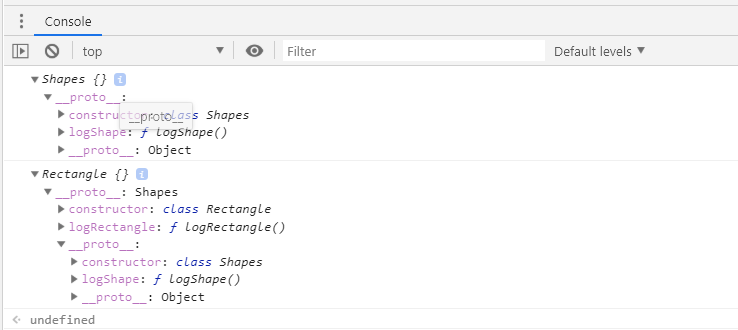
console.log(poly1.name);

\*\*\* If there is a constructor present in the subclass, it needs to first call super() before using "this".\*\*\*

“this” keyword in parent and child class refers to respective class objects



On extending console output:



Rectangle class object contains Shape class object along with function logShapes().



