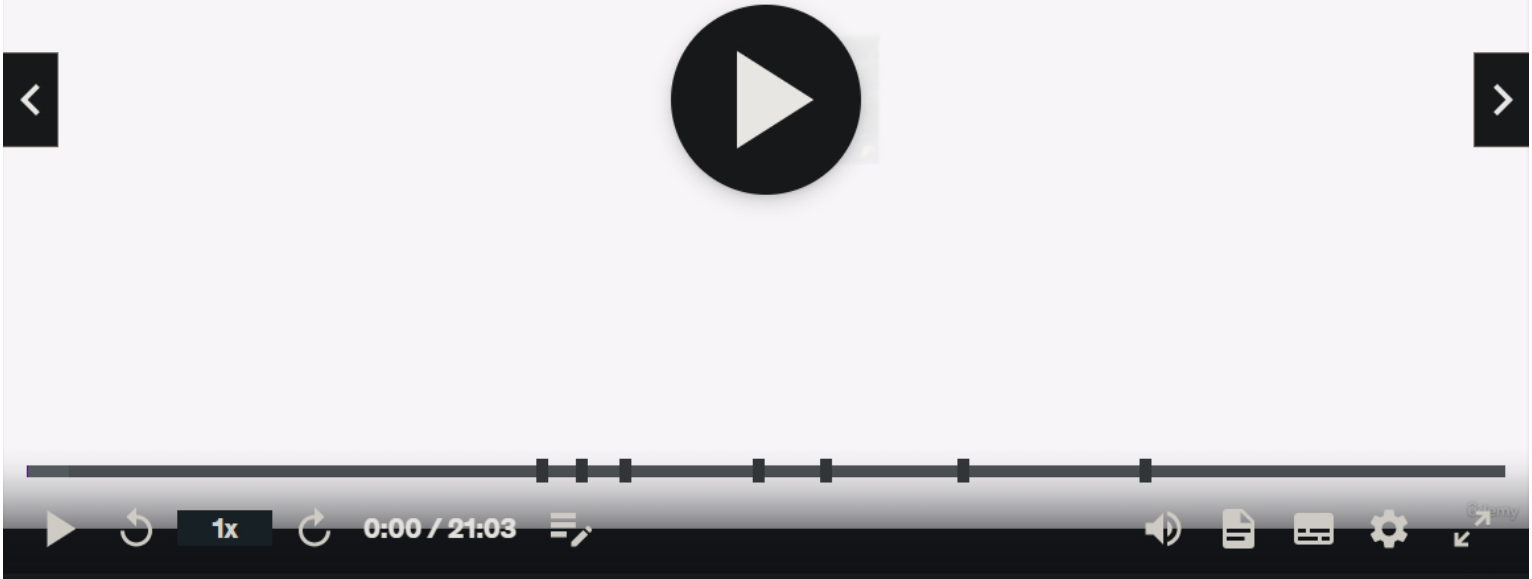
218. Inheritance Between "Classes": Constructor Functions



Back to All Questions

M

"Student.prototype.constructor = Student" doesn't work

44

Mirela · Lecture 218 · 2 years ago

mike.__proto__ is still Person even after Student.prototype.constructor = Student;

The correct __proto__ of mike would be **Student or Person**???

Thank You

Here is the code:

```
const Person = function (firstName, birthYear) {
  this.firstName = firstName;
  this.birthYear = birthYear;
};

Person.prototype.calcAge = function () {
  console.log(2037 - this.birthYear);
```

```
};

const Student = function (firstName, birthYear, course) {
  Person.call(this, firstName, birthYear);
  this.course = course;
};
Student.prototype = Object.create(Person.prototype);
Student.prototype.constructor = Student;

Student.prototype.introduce = function () {
  console.log(`My name is ${this.firstName} and I study ${this.course}`);
};

const mike = new Student('Mike', 2020, 'Science');
mike.calcAge();
mike.introduce();
console.log(mike.__proto__);  // Person
```

12 replies **Follow replies**

Ayo

2 years ago

6

same for me

VV

Vigil

2 years ago

3

Yep! Came here to comment the same.

AS

Arthur

2 years ago

3

Same here!! Someone please clarify???

Aleksander — Teaching Assistant

2 years ago

11

Hello Everyone

The Student function is a constructor for the mike object, and the Student.prototype is its prototype.

The problem is with how Chrome outputs objects to the console. If you look at the output in Firefox, you will get a more specific information

```
17                                                    script.js:7:11
My name is Mike and I study Science                   script.js:19:11
▶ Object { constructor: Student(firstName, birthYear,  script.js:25:9
course) ↱≡, introduce: introduce() ↱≡ }
```

M

Melo

2 years ago

0

Aleksander is right. You can change to firefox to see it. But the question is why google chrome shows that?

Aleksander — Teaching Assistant

2 years ago

1

That's a good question, Melo. I don't know the reason, but it may change in the future. If I remember correctly, the output was different in older versions of Chrome

JH

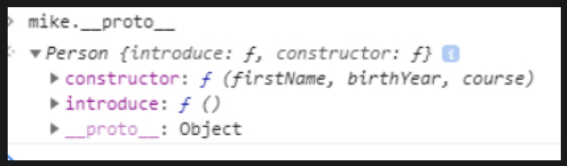
Jay

2 years ago

2

I'm on the same boat and just wanted to confirm what you were saying Aleksander.

This is an issue specific to Chrome where the name of the prototype is incorrectly displayed, but in the background the prototype of 'mike' is Student.prototype?



In the above picture, mike.__proto__ shows "Person" as the prototype name, but everything else in the prototype is actually from Student.prototype. So it is just incorrectly displaying "Person" instead of "Student"?

MR

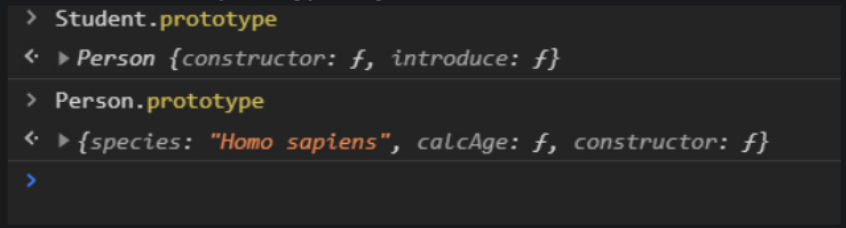
Mushfiqur

2 years ago

1

Hi,

I don't know if I'm right, but I think Chrome shows the name of the prototype that is one level above the current prototype in the prototype chain. In this case, Mike's prototype inherits from Person prototype, so maybe that's why Person is being shown before the prototype object. I've also noticed that Chrome doesn't show any prototype name when the preceding prototype is the Object prototype, i.e. if you check Person.prototype in the console, you will see that there is no name before the prototype object.



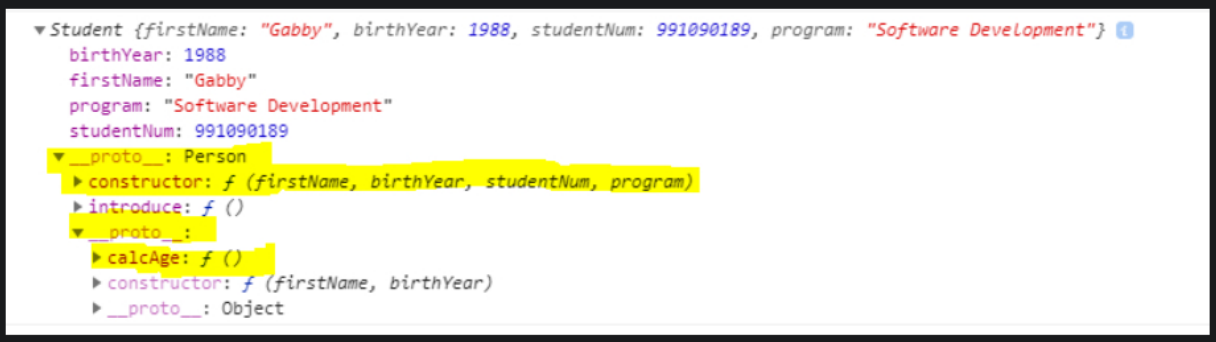
Maybe this is Chrome's weird way of showing the preceding prototype in the prototype chain.

Gabby

0

GD

2 years ago



It is weird. It states the __proto__ points to the Person class but if you look at the constructor function, its actually the constructor function of the Student class. Looking at the complete prototype chain, you can see its pointing to the correct constructor function. Did we find a bug in chrome or was this on purpose? lol I mean, the way its currently set up is confusing, why show the name that is one level above when the constructor function clearly doesn't belong to that specific class? IDK lol but I'd recommend everyone get Firefox developer edition, it's better than using chrome developer tools.

Aleksander — Teaching Assistant

3

2 years ago

Yes, it's just Chrome that outputs the prototype chain in a confusing way. Don't know if it's a bug. Maybe it's on purpose that I'm missing. Anyway, Firefox shows it in a much cleaner and understandable manner.

You can always check the equality between prototypes, like that

```
1 | Object.getPrototypeOf(mike) === Student.prototype
```

KP

Krishna Hari

1

2 years ago

Thanks to all of you, I got the answer to my same confusion...

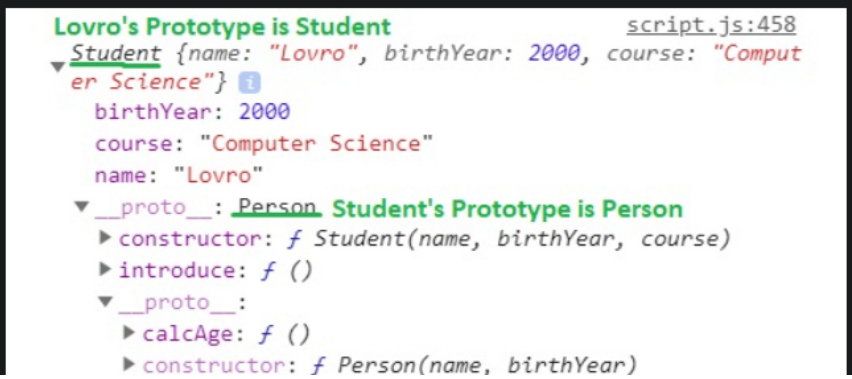
Nejc — Teaching Assistant

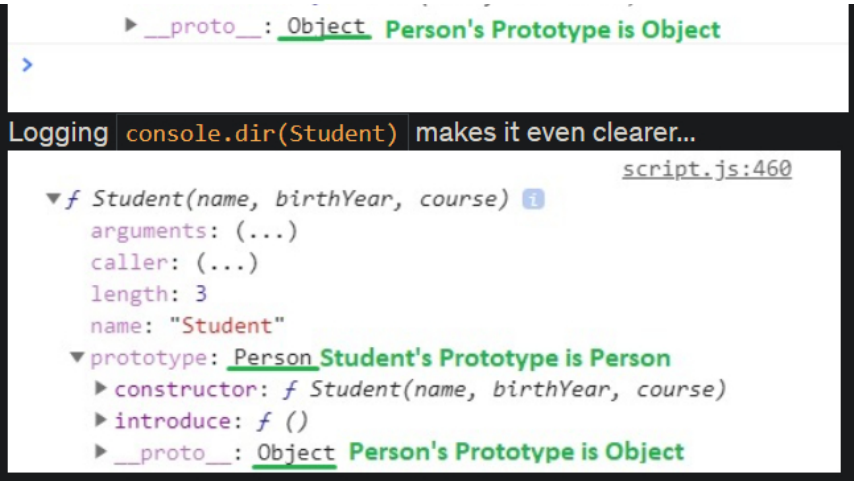
15

2 years ago

I think that it works as intended. Maybe not the best implementation from Google, but what can you do...

```
1 | const lovro = new Student('Lovro', 2000, 'Computer Science');
```





Load more answers

MR

Add reply