

Язык JavaScript (Классы)

JS

Классы

В стандарте ES-2015 появился новый, «более красивый» синтаксис для классов.

Новая конструкция `class` – удобный «синтаксический сахар» для задания конструктора вместе с прототипом.

Синтаксис для классов выглядит так:

```
class Название [extends Родитель] {  
    constructor  
    методы  
}
```

Классы

```
class Human{
    constructor(name){
        this.name = name;
    }
    toString(){
        return "Привет. Меня зовут " + this.name + ".";
    }
}

class Student extends Human{
    constructor(name, hobby){
        super(name);
        this.hobby = hobby;
    }
    toString(){
        return super.toString() + " Хобби: " + this.hobby;
    }
}
```

Классы

Функция `constructor` запускается при создании `new Human`, остальные методы записываются в `Human.prototype`.

При наследовании через `extends` формируется стандартная цепочка прототипов: методы `Human` находятся в `Human.prototype`, методы `Student` – в `Student.prototype`, и они связаны через `__proto__`

Конструктор `constructor` родителя наследуется автоматически. То есть, если в потомке не указан свой `constructor`, то используется родительский.

В классах можно создавать статические методы, геттеры и сеттеры.

Классы

Вызвать конструктор родителя можно только изнутри конструктора потомка используя метод `super`. В конструкторе потомка мы обязаны вызвать `super()` до обращения к `this`. До вызова `super` не существует `this`.

При объявлении через `class` есть и ряд отличий:

- ❖ `Human` нельзя вызывать без `new`, будет ошибка.
- ❖ Объявление класса с точки зрения области видимости ведёт себя как `let`.

Методы, объявленные внутри `class`, также имеют ряд особенностей:

- ❖ Метод `toString` является именно методом, то есть имеет доступ к `super`.
- ❖ Все методы класса работают в строгом режиме `use strict`, даже если он не указан.

Классы

Вызвать конструктор родителя можно только изнутри конструктора потомка используя метод `super`. В конструкторе потомка мы обязаны вызвать `super()` до обращения к `this`. До вызова `super` не существует `this`.

При объявлении через `class` есть и ряд отличий:

- ❖ `Human` нельзя вызывать без `new`, будет ошибка.
- ❖ Объявление класса с точки зрения области видимости ведёт себя как `let`.

Методы, объявленные внутри `class`, также имеют ряд особенностей:

- ❖ Метод `toString` является именно методом, то есть имеет доступ к `super`.
- ❖ Все методы класса работают в строгом режиме `use strict`, даже если он не указан.

ЧТО ПОЧИТАТЬ ДОПОЛНИТЕЛЬНО

<https://learn.javascript.ru/classes> - Раздел онлайн учебника
«Классы»