深入理解类数组以及多种方式实现类数组转真数组

jCodeLife
2020.08.12 11:35:22 字数 1,147 阅读 3

编辑文章

我们经常会听人提到类数组，也经常会遇到类数组转真数组的怎么转的问题？要理解类数组，那得说一下真数组

我们知道，js数组是用一个变量存储多个数据的一种特殊的数据结构，可以通过数组下标获取对应位置的数据，并且js提供了一系列的属性和方法来操作数组。

我理解的类数组其实就是类似数组的对象；本质是长得很像的两个东西 一个对象一个数组，因为它长得像数组，使用起来也挺像数组，所以大家后面习惯把它叫做类数组。

长得相似的行星

类数组有几个必要组成部分：

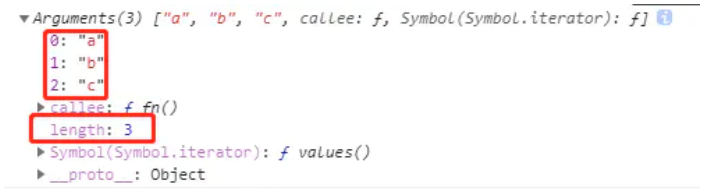
1. 属性要为索引（数字）属性；
2. 必须有length属性
3. 最好加上数组的push和splice方法

类数组并不陌生，函数中的arguments就是属于类数组，在DOM操作中，我们也经常会得到一个类数组（比如document.getElementsByClassName()）

那再来深入看看，到底里面都有什么？
先定义一个普通方法并执行，把打印 arguments看看

```
function fn(){
  console.log(arguments)
}
fn('a','b','c')
```

结果如下：



这验证了我们上面说的两点：1.属性要为索引属性；2.必须有length属性

接下来继续试试，看类数组能不能像数组那样进行一些操作

jCodeLife
总资产4 (约0.41元)

推荐阅读

结果如下

发现我们通过下标的方式修改、查看和添加类数组的数据，但是这种方式不会影响length。仔细一想，这更像对象的基本特性，就是将下标当他属性来存取。只是他看起来像数组一样。

接着看看能不能用数组的push方法

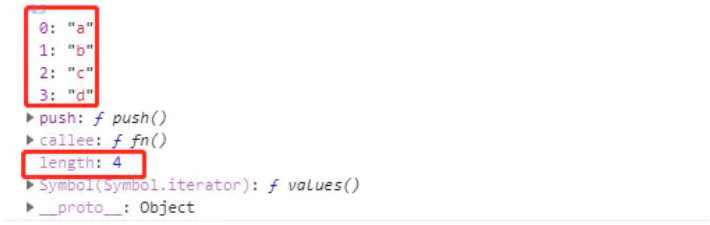
```
arguments.push("push");
```

答案跟你想的一样，是不可以的！
它会一个错误：`Uncaught TypeError: arguments.push is not a function`
其实你只要想清楚一个事就很容易理解了，
我问问大家，我们在数组中使用的一些列方法（如push、pop、shift、splice、join、concat...等等）都是定义在哪里的？
是不是定义在构造函数Array的原型上的，这样所以的实例数组都能使用！而这里arguments即没有添加对应的方法，也没有继承自Array.prototpe，自然没有这写数组方法。

接下来我们来给arguments添加一个push方法

```
function fn() {
  arguments.push = Array.prototype.push;
  arguments.push('d');
  console.log(arguments)
}
fn('a', 'b', 'c')
```

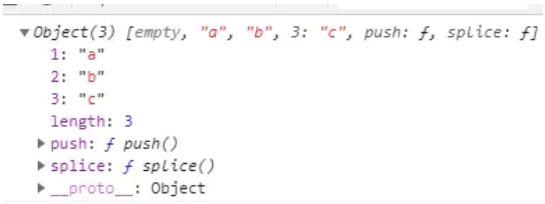
结果：`添加push方法的类数组，push数据的时候会动态的增长length属性。`



除了arguments我们可以自定义一个类数组

```
//自定义数组
var obj = {
  1: "a",
  2: 'b',
  3: 'c',
  length: 3,
  push: Array.prototype.push,
  splice: Array.prototype.splice
}
```

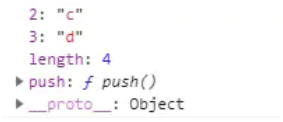
数组打印出来时看上去跟数组一模一样。不过我目前的最新谷歌版本是打印出来还是类数组的样子：



不过这样不要紧，splice加不加都行。
这里插播一个笔试题吧：

```
var obj = {
  2: 'a',
  3: 'b',
  length : 2,
  push: Array.prototype.push,
}
obj.push('c');
obj.push('d');
console.log(obj);
```

答案可能出乎你得预料：



其实的理解push的内部原理，得回到我们之前手动封装push方法的时候

```
Array.prototype._push = function(){
    for(var i=0;i<arguments.length;i++){
        this[this.length] = arguments[i];
        this.length++
    }
    return this.length;
}
```

里头核心操作`this[this.length] = arguments[i]`和`this.length++`，就是跟数组的length位置添加数据，然后让length++。所以有了上面的答案。

最后，如果你希望将数组转成真数组，那继续往下看：

类数组转成真数组的几种方法

`1. 用Array.prototype.slice.call（obj）方法转化为数组`
在讲数组的时候有提到，通过slice方法可以拷贝得到新数组

```
var arr = [1,2,3,4,5];
var newArr = arr.slice()
console.log(newArr);//截取整个数组       [1,2,3,4,5];
```

`2. Array.from()方法`

```
var arr= [...obj];
```

以上是最简单也最常见的3种方式。还有几种利用call、apply、bind和map forEach几个方法配合实现，想了解的可以点击下方第一个链接

参考资料：

类数组对象转换为数组的六种方法—作者：霓裳依旧

数组 类数组—腾讯课堂渡一教育

JS数组及手动封装ES3的数组核心方法—作者：jCodeLife

jCodeLife 书山有路勤为径，学海无涯苦作舟

总资产4 (约0.41元) 共写了9.0W字 获得101个赞 共19个粉丝