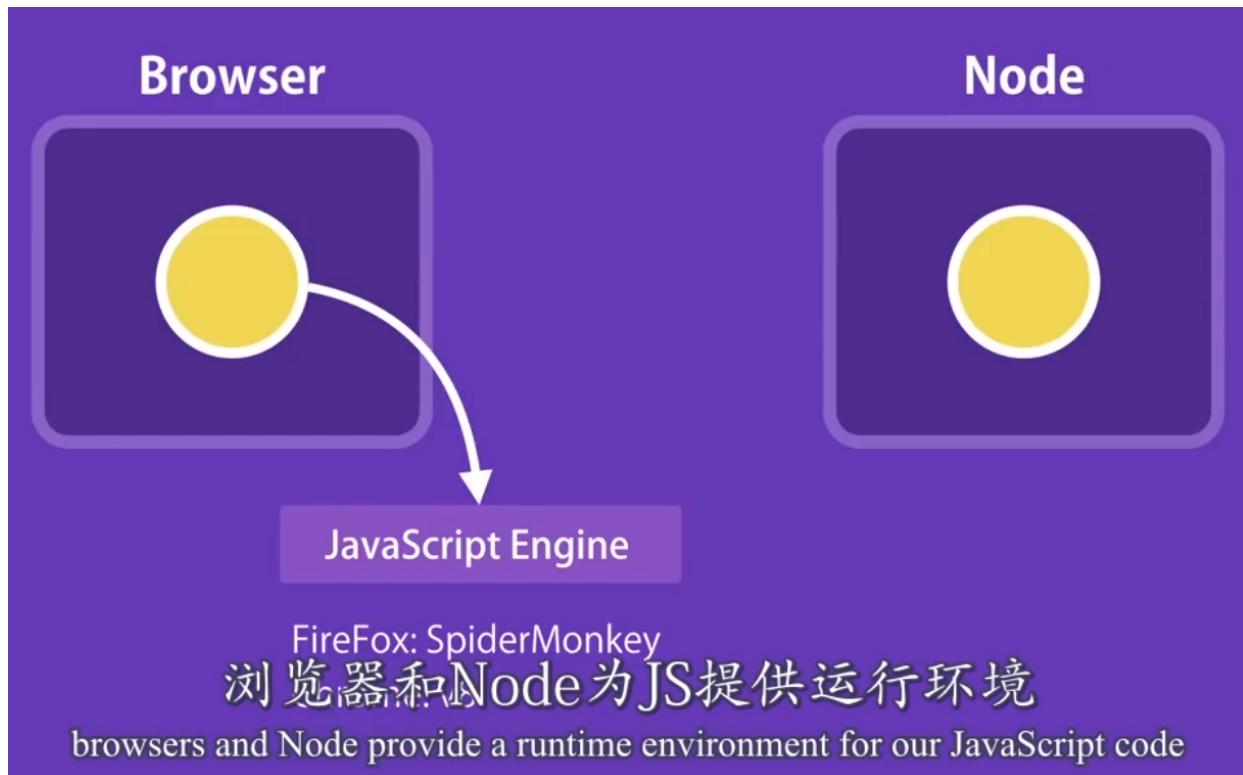


JS Crash Course For beginners



JS变量的属性都没有括号，方法才有括号

字符串模版用反引号

```
const name = 'Frank'
```

```
const age = 22
```

```
console.log(`My name is ${name} and I am ${age}`)
```

js可以把函数当成参数，forEach, map, filter

```
1 const addNums = num1 => num1 + 5;  
2  
3  
4 console.log(addNums(5));  
5  
6 todos.forEach((todo) => console.log(todo));
```

arrow function: functionname(param) => 返回值 / 方法体

```
1 // Constructor function
2 function Person(firstName, lastName, dob) {
3     this.firstName = firstName;
4     this.lastName = lastName;
5     this.dob = new Date(dob);
6 }
7
8 // Instantiate object
9 const person1 = new Person('John', 'Doe', '4-3-1980');
10 const person2 = new Person('Mary', 'Smith', '3-6-1970');
11
12 console.log(person2.dob);
13
```

JS Crash Course

Fri Mar 06 main.js:12
1970 00:00:00 GMT-0500
(Eastern Standard Time)

```
1 // Constructor function
2 function Person(firstName, lastName, dob) {
3     this.firstName = firstName;
4     this.lastName = lastName;
5     this.dob = new Date(dob);
6 }
7
8 Person.prototype.getBirthYear = function() {
9     return this.dob.getFullYear();
10 }
11
12 Person.prototype.getFullName = function() {
13     return `${this.firstName} ${this.lastName}`;
14 }
15
16
17 // Instantiate object
18 const person1 = new Person('John', 'Doe', '4-3-1980');
19 const person2 = new Person('Mary', 'Smith', '3-6-1970');
20
21 console.log(person2.getFullName());
22 console.log(person1);
```

JS Crash Course

Person {firstName: "John", lastName: "Doe", do...
b: Thu Apr 03 1980 00:00:00 GMT-0500 (Eastern Standard Time)} ↴
▶ dob: Thu Apr 03 1980 00:00:00 GMT-0500 (Eastern Standard Time)
▶ firstName: "John"
▶ lastName: "Doe"
▶ __proto__:
▶ getBirthYear: f ()
▶ getFullName: f ()
▶ constructor: f Pers...
▶ __proto__: Object ↴

这就是ES5中的面向对象编程
all right so that is basically object-oriented programming in es5

index.html JS main.js x

```

1 // Class
2 class Person {
3   constructor(firstName, lastName, dob) {
4     this.firstName = firstName;
5     this.lastName = lastName;
6     this.dob = new Date(dob);
7   }
8
9   getBirthYear() {
10    return this.dob.getFullYear();
11  }
12
13  getFullName() {
14    return `${this.firstName} ${this.lastName}`;
15  }
16 }
17
18 // Instantiate object
19 const person1 = new Person('John', 'Doe', '4-3-1980');
20 const person2 = new Person('Mary', 'Smith', '3-6-1970');
21
22 console.log(person2.getFullName());
23 console.log(person1.getBirthYear());
24
25 
```

Mary Smith main.js:22
main.js:23
Person {firstName: "John", lastName: "Doe", dob: "Thu Apr 03 1980 00:00:00 GMT-0500 (Eastern Standard Time)}

这些基本就是面向对象的基础
so that that's pretty much the basics of object-oriented programming

index.html # style.css JS main.js x

```

1 console.log(window);
2
3 // Single element
4
5
6 // Multiple element
7
8 
```

JS For Beginners

Add User
Name:
Email:
Submit

window对象是浏览器的父对象
and the window object is the parent object of the browser

Object.getOwnPropertyNames(window)

- alert: f alert()
- applicationCache: ApplicationCache
- atob: f atob()
- blur: f ()
- btoa: f btoa()
- caches: CacheStorage
- cancelAnimationFrame: f cancelAnimationFrame()
- cancelIdleCallback: f cancelIdleCallback()
- captureEvents: f captureEvents()

window是最高级别一切对象皆属于window

The screenshot shows a browser developer tools console with two tabs: 'index.html' and 'main.js'. The 'main.js' tab contains the following code:

```
1 // Single element
2 console.log(document.getElementById('my-form'));
3 console.log(document.querySelector('h1'));
4
5 // Multiple element
6 console.log(document.querySelectorAll('.item'));
```

The output of this code in the console is:

```
▶<form id="my-form">...
</form>
▶<h1>JS For Beginners
</h1>
▶ NodeList(3) [li.item, li.item, li.item]
```

Below the console, there is a UI component titled 'Add User' with fields for Name and Email, and a 'Submit' button. There are also three items labeled 'Item 1', 'Item 2', and 'Item 3'.

querySelector类似jQuery选择器

The screenshot shows a browser developer tools console with two tabs: 'index.html' and 'main.js'. The 'main.js' tab contains the following code:

```
1 const items = document.querySelectorAll('.item');
2
3 items.forEach((item) => console.log(item));
4
```

The output of this code in the console is:

```
▶<li class="item">Item 1
</li>
▶<li class="item">Item 2
</li>
▶<li class="item">Item 3
```

Below the console, there is a UI component titled 'Add User' with fields for Name and Email, and a 'Submit' button. There are also three items labeled 'Item 1', 'Item 2', and 'Item 3'.

会看到遍历并获取了每一项

and you'll see that we're just looping through and grabbing each of the list items

The screenshot shows a browser window titled "JS For Beginners". On the left, there is a code editor with two tabs: "ex.html" and "JS main.js". The "JS main.js" tab contains the following code:

```
1 const ul = document.querySelector('.items');
2
3 ul.remove();
```

The third line, `ul.remove();`, is highlighted with a red rectangular box. To the right of the code editor is the browser's developer tools, specifically the Network tab, which shows a list of resources being loaded.

Below the browser window, there is a large text overlay in Chinese: "会看到整个ul都不见了" (You will see the entire ul disappear) and its English translation: "notice that the the UL is now gone the whole thing is gone".

The screenshot shows a browser window titled "JS For Beginners". On the left, there is a code editor with two tabs: "index.html" and "JS main.js". The "JS main.js" tab contains the following code:

```
1 const ul = document.querySelector('.items');
2
3 // ul.remove();
4 ul.lastElementChild.remove();
```

The fourth line, `ul.lastElementChild.remove();`, is highlighted with a green rectangular box. To the right of the code editor is the browser's developer tools, specifically the Network tab, which shows a list of resources being loaded.

Below the browser window, there is a large text overlay in Chinese: "现在最后一项就木有了" (Now the last item is gone) and its English translation: "and now item three is gone".

main.js — js_crash_course

```
1 const ul = document.querySelector('.items');
2
3 // ul.remove();
4 // ul.lastElementChild.remove();
5 ul.firstChild.textContent = 'Hello';
6 ul.children[1].innerText = 'Brad';
7 ul.lastElementChild.innerHTML = '<h1>Hello</h1>';
```

JS For Beginners

Add User

Name:

Email:

Submit

Hello

Brad

Hello

如果想动态添加html内容
so if you want to add HTML dynamically

用innerHTML动态添加html内容

表单验证

main.js — js_crash_course

```
1 const myForm = document.querySelector('#my-form');
2 const nameInput = document.querySelector('#name');
3 const emailInput = document.querySelector('#email');
4 const msg = document.querySelector('.msg');
5 const userList = document.querySelector('#users');
6
7 myForm.addEventListener('submit', onSubmit);
8
9 function onSubmit(e) {
10   e.preventDefault();
11
12   if(nameInput.value === '' || emailInput.value === '') {
13     msg.classList.add('error');
14     msg.innerHTML = 'Please enter all fields';
15
16     setTimeout(() => msg.remove(), 3000);
17   } else {
18     console.log('success');
19   }
20 }
```

JS For Beginners

Add User

Name:

Email:

Submit

top

```
index.html  JS main.js  # style.css
1 const nameInput = document.querySelector('#nameInput');
2 const emailInput = document.querySelector('#emailInput');
3 const msg = document.querySelector('.msg');
4 const userList = document.querySelector('#users');
5
6
7 myForm.addEventListener('submit', onSubmit);
8
9 function onSubmit(e) {
10   e.preventDefault();
11
12   if(nameInput.value === '' || emailInput.value === '') {
13     msg.classList.add('error');
14     msg.innerHTML = 'Please enter all fields';
15
16     setTimeout(() => msg.remove(), 3000);
17   } else {
18     const li = document.createElement('li');
19     li.appendChild(document.createTextNode(`${nameInput.value} : ${emailInput.value}`));
20
21     userList.appendChild(li);
22
23   // Clear fields
24   nameInput.value = '';
25   emailInput.value = '';
26 }
27 }
```

这就是如何操纵DOM的例子

all right so that's an example of how we can manipulate the DOM