FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

JAVASCRIPT NÂNG CAO

BẤT ĐỒNG BỘ VÀ XỬ LÝ BẤT ĐỒNG BỘ (PHẦN 1)

THỰC HỌC – THỰC NGHIỆP

http://www.poly.edu.vn

Mục tiêu

- Giải thích được khái niệm bất đồng bộ trong javascript
- Có thể tự xử lý được bài toán bất đồng bộ khi thực hiện công việc (sử dung callback)

NỘI DUNG

- 📖Giải thích được khái niệm bất đồng bộ trong Javascript.
- 📖Nguyên lý xử lý bất đồng bộ trong javascript
- 📖Giải thích được khái niệm Callback
- 📖Sử dụng được Callback

NHẮC LẠI BÀI CŨ

- Sử dụng được Object literals
- Giải thích được Prototyping inheritance
- Sử dụng được Object constructor functions

PHẦN 1:
BẤT ĐỒNG BỘ TRONG JAVASCRIPT

ĐỒNG BỘ VÀ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- Xử lý đồng bộ (Synchronous): Chương trình sẽ chạy theo từng bước và chỉ khi nào bước 1 thực hiện xong thì mới nhảy sang bước 2, khi nào chương trình này chạy xong mới nhảy qua chương trình khác. Chương trình sẽ chạy theo trình tự code từ trên xuống dưới.
- Mặt tốt của xử lý đồng bộ:
  - Dễ kiểm soát quá trình xử lý logic của chương trình
  - Dễ kiểm soát lỗi phát sinh
- Mặt không tốt của xử lý đồng bộ:
  - Chương trình chạy theo thứ tự do đó sẽ có trạng thái chờ, điều này gây chậm việc xử lý, nhiều khi gây tràn bộ nhớ

ĐỒNG BỘ VÀ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- Xử lý bất đồng bộ (Asynchronous): Chương trình cho phép nhiều công việc có thể được thực hiện cùng lúc. Và nếu công việc thứ hai kết thúc trước, nó có thể sẽ cho ra kết quả trước cả câu lệnh thứ nhất. Vì thế, đôi khi kết quả của các câu lệnh sẽ không trả về đúng theo đúng thứ tự như trực quan của nó.
- Mặt tốt của xử lý bất đồng bộ:
  - Tối ưu được sức mạnh của hệ thống
  - Giảm thời gian chờ đợi của người dùng, tạo cảm giác thoải mái hơn khi sử dụng
- Mặt không tốt của xử lý đồng bộ:
  - Không phải chương trình, hệ thống nào cũng sử dung bất đồng bộ được
  - Khó làm quen và kiểm soát lỗi phát sinh

ĐỒNG BỘ VÀ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

Ví dụ 1:

Xử lý đồng bộ

```
<script>

    console.log('message thứ nhất');
    console.log('message thứ hai');

</script>
```

```
message thứ nhất
message thứ hai
```

Xử lý bất đồng bộ

```
<script>

    setTimeout(function (){
        console.log('message thứ nhất');
    }, 1000);
    console.log('message thứ hai');

</script>
```

```
message thứ hai
message thứ nhất
```

NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- Nguyên lý bất đồng bộ trong javascript được dựa trên 4 thành phần sau:
  - **CALL STACK** - là một dạng cấu trúc dữ liệu ghi lại vị trí các lệnh đang được thực hiện trong chương trình. Khi lệnh bắt đầu được thực hiện sẽ được đưa vào đỉnh của stack và sau khi thực hiện xong sẽ được lấy ra khỏi ngăn xếp.
  - **WEB APIs** - về bản chất đây chính là các luồng xử lý mà ta không thể truy cập trực tiếp mà chỉ có thể gọi được đến nó. Các luồng này do trình duyệt cung cấp.
  - **CALLBACK QUEUE** - là một dạng cấu trúc dữ liệu với nguyên tắc First-In-First-Out (vào trước ra trước).
  - **EVENT LOOP** - có nhiệm vụ giám sát tình trạng của CALL STACK và CALLBACK QUEUE. Để hiểu được quá trình thực hiện của cơ chế bất đồng bộ ta sẽ đưa ví dụ thứ hai vào và thực hiện trong mô hình trên.

NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

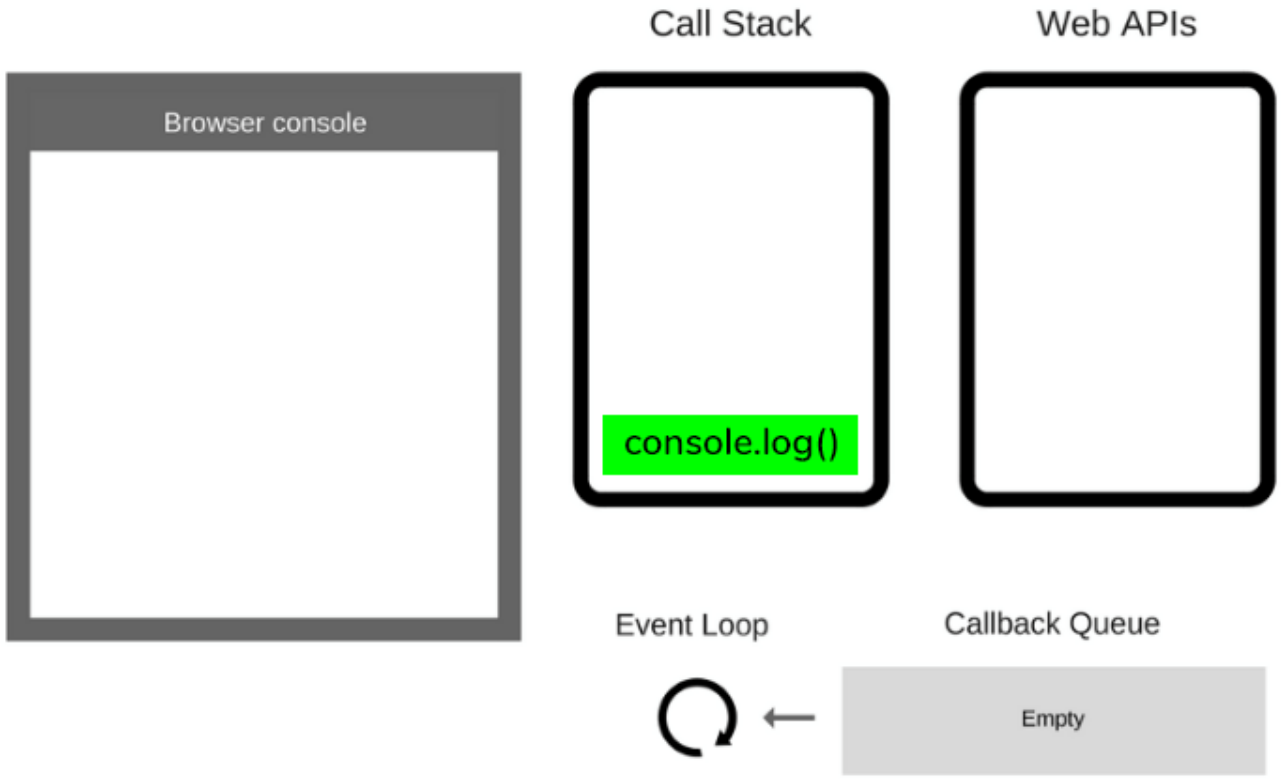
Ví dụ 2:

```
<script>

    console.log('This is the first line');

    setTimeout(function(){
        console.log('This is the second line');
    }, 1000);

    console.log('This is the last line');


</script>
```

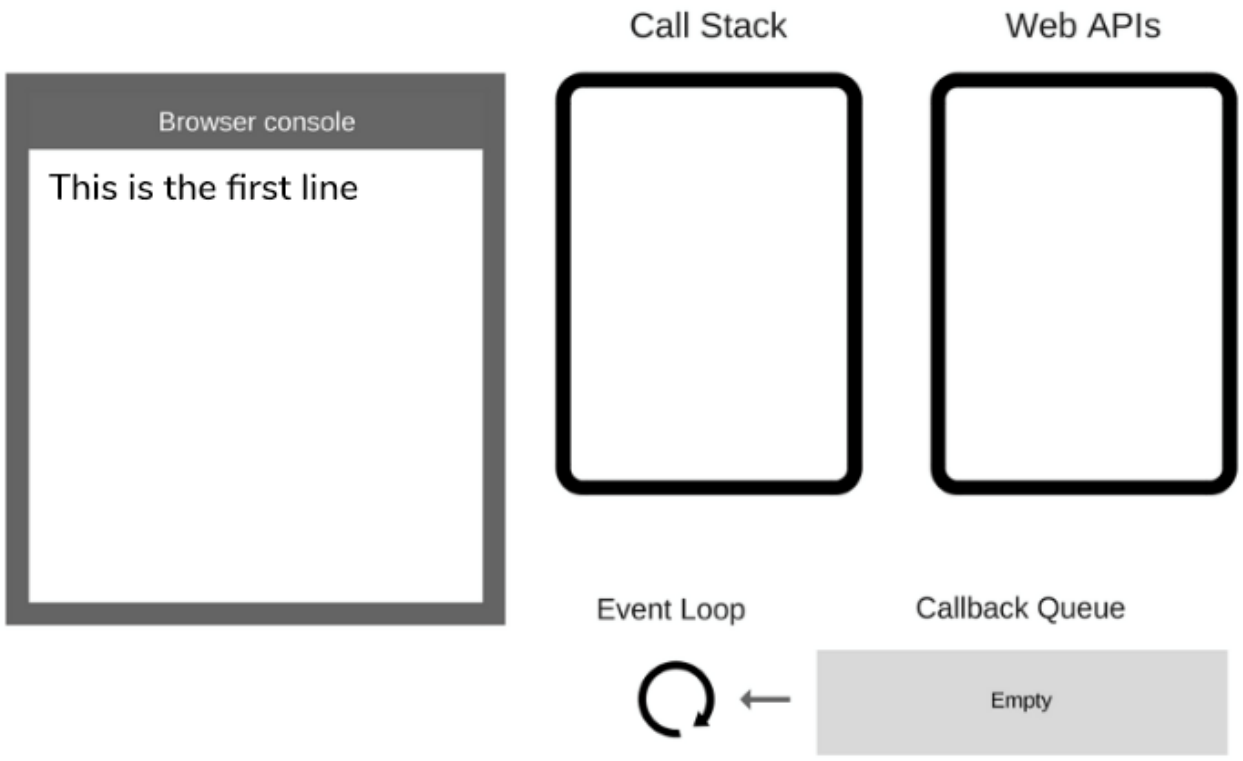
```
This is the first line
This is the last line
This is the second line
```

NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- 1. Khi chương trình bắt đầu chạy lệnh đầu tiên của chương trình (console.log('This is the first line')) sẽ được đưa vào trong **CALL STACK**.

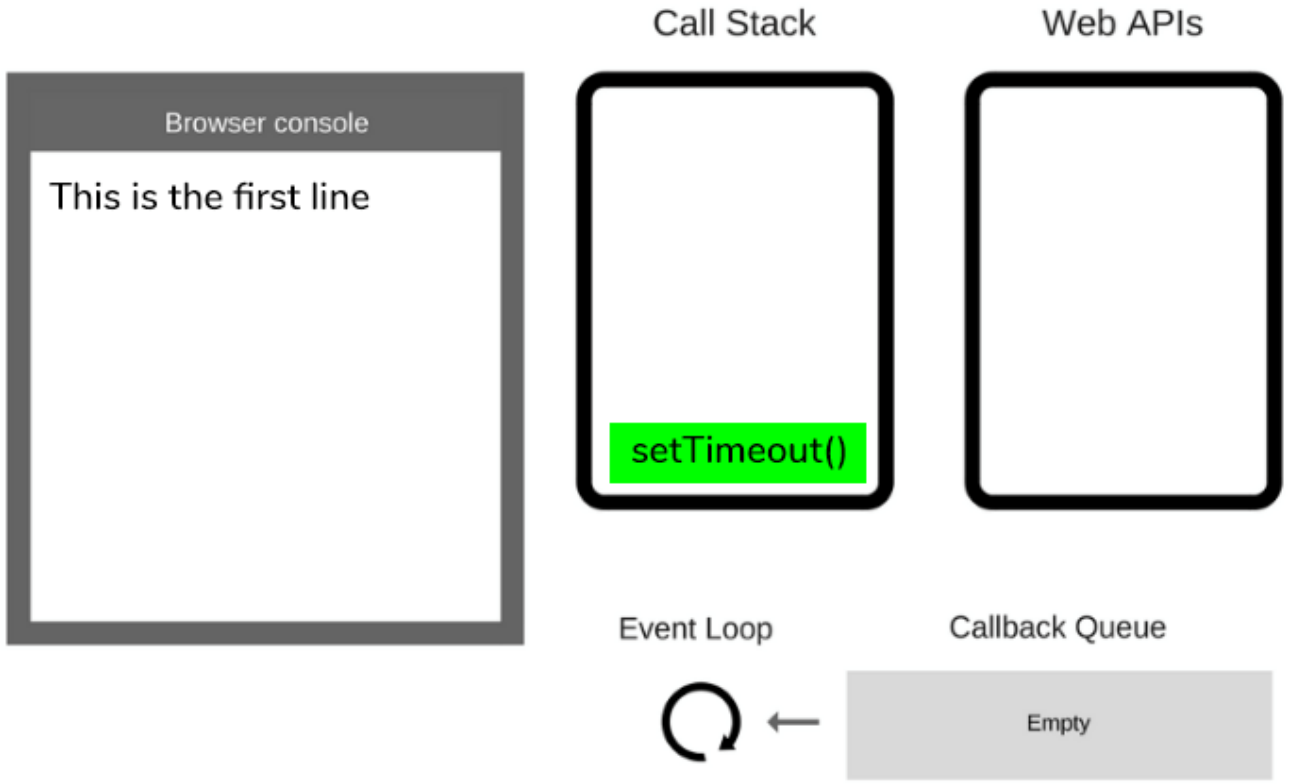


- 2. Lệnh này lập tức trả về dòng chữ This is the first line đồng nghĩa với việc nó đã chạy xong và được đẩy ra khỏi **CALL STACK**.



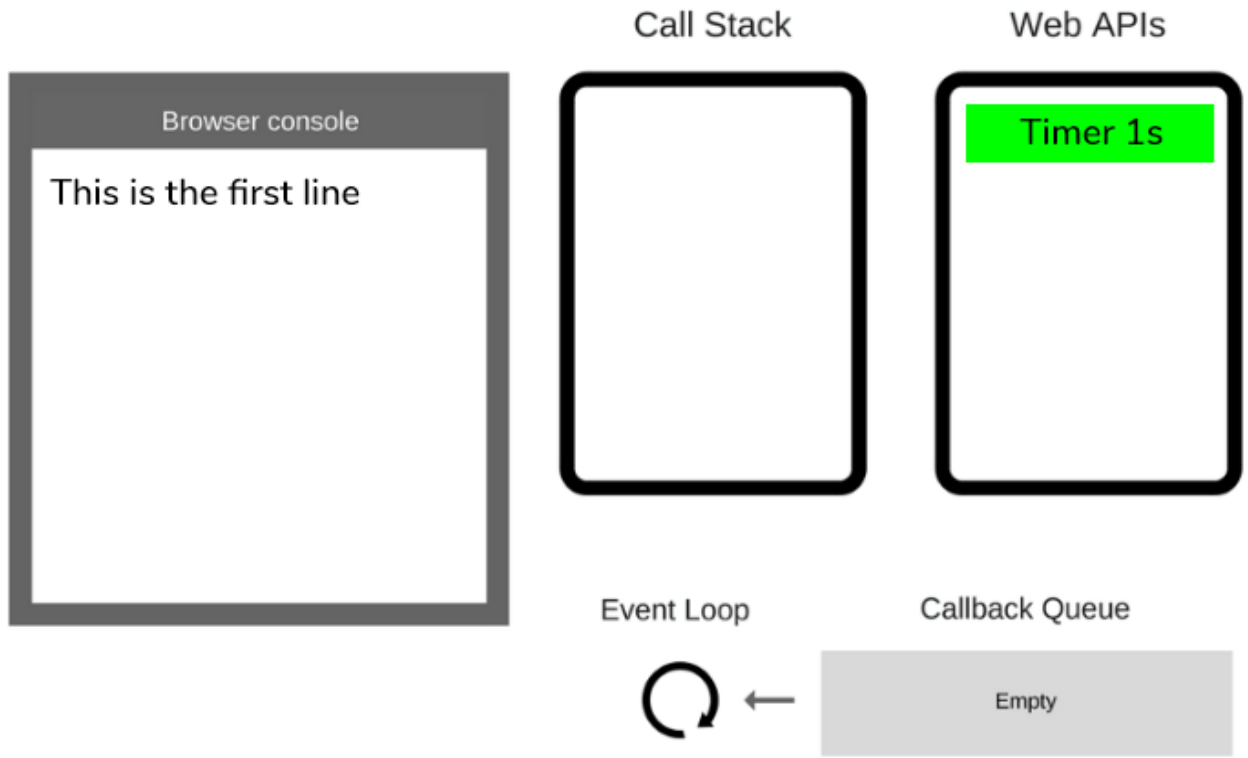
NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- 3. Tiếp đến hàm setTimeout(function() { console.log('This is the second line'); }, 1000); được đưa vào trong **CALL STACK** để thực hiện.



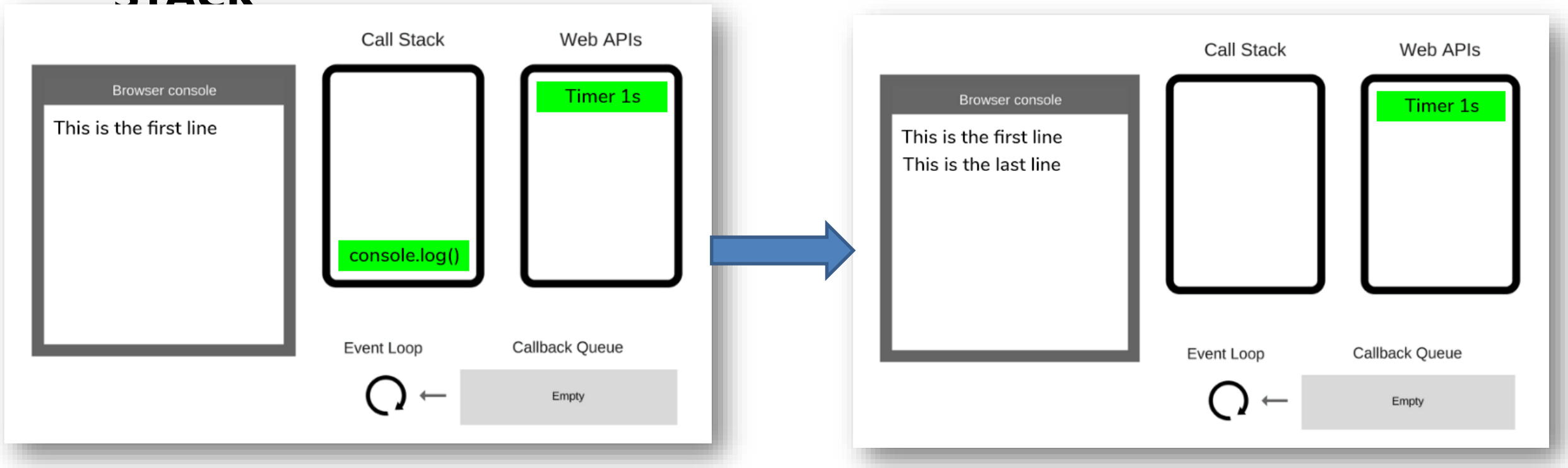
NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- 4. Hàm nay không trả về kết qua ngay mà phải đợi 1 giây. Hàm setTimeout() ở đây chính là một API mà **WEB APIs** cung cấp. Lập tức đoạn code này được chuyển vào trong **WEB APIs** và trình duyệt sẽ tạo ra một bộ hẹn giờ tương ứng với thời gian trên là 1 giây trước khi trả về kế



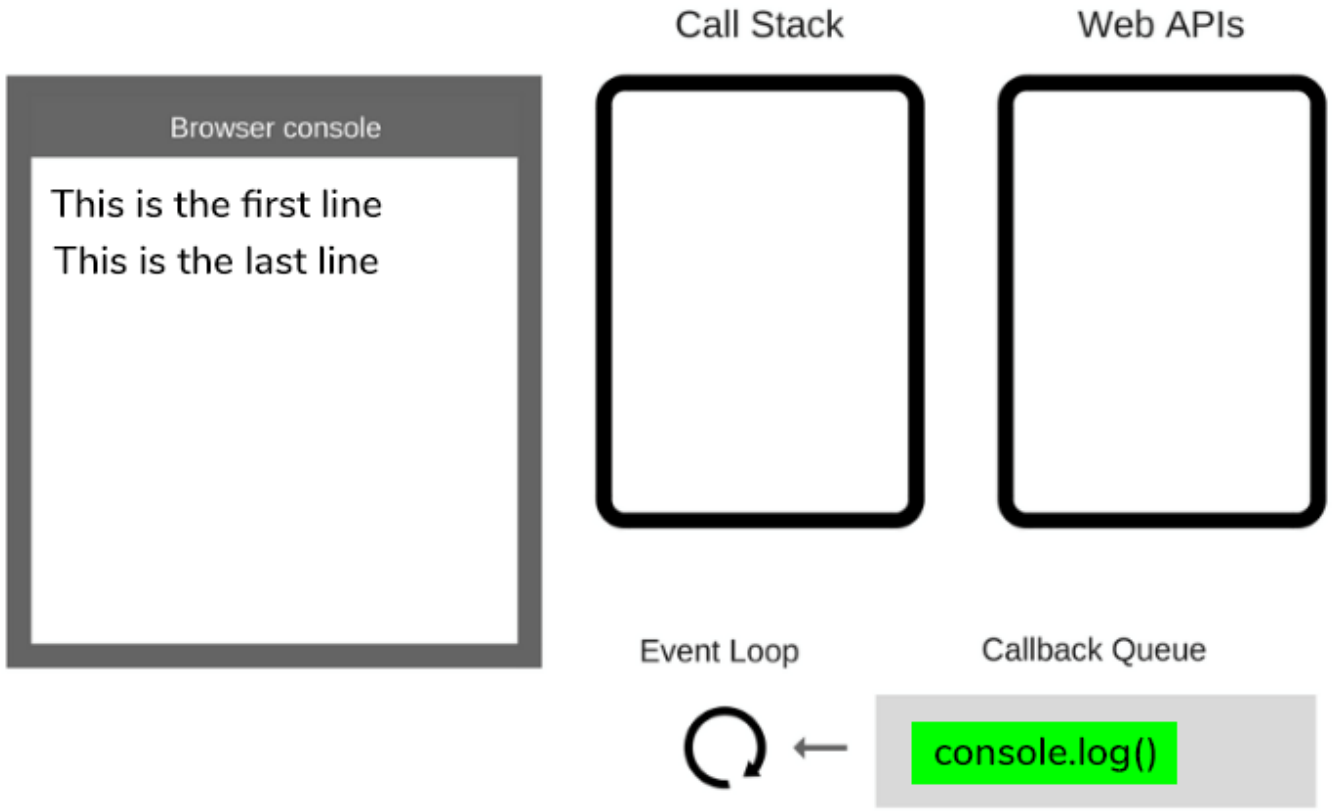
NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- 5. Đoạn code thứ 2 được chuyển sang **WEB APIs** thì lập tức đoạn code cuối cùng console.log('This is the last line ') đã được đưa vào **CALL STACK** để thực hiện và trả về kết quả là dòng chữ This is the last line. Sau đó đoạn code này cũng được đẩy ra khỏi **CALL STACK**



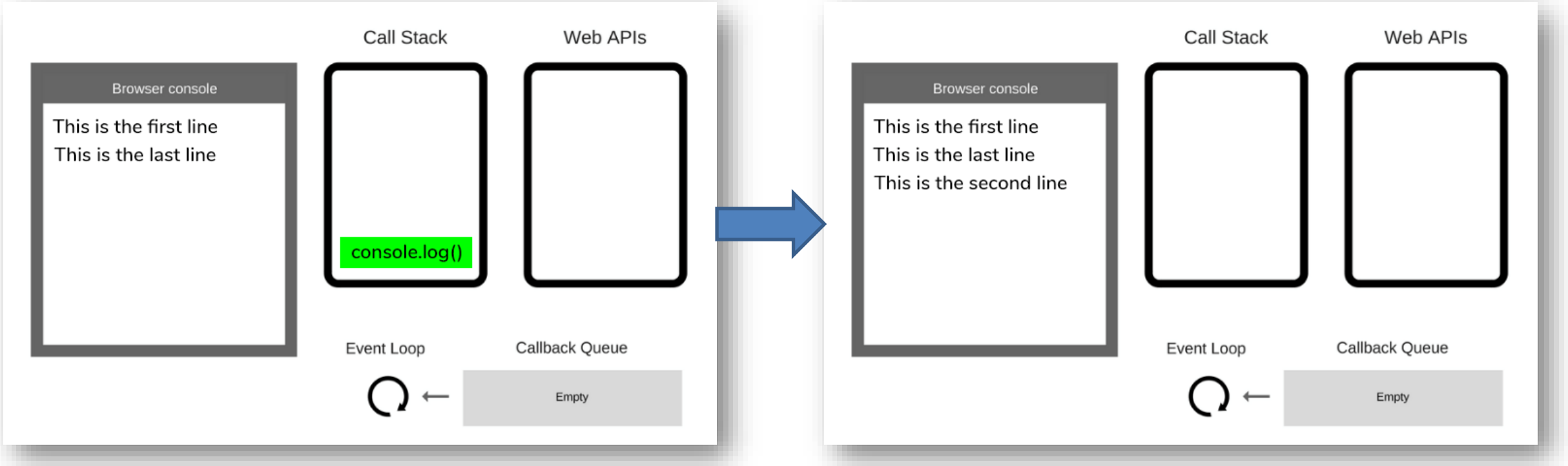
NGUYÊN LÝ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- 6. Sau khi bộ giờ trong **WEB APIs** chạy xong thì kết quả trả về lúc này không được in ngay ra màn hình mà nó được đẩy vào **CALLBACK QUE**



- 7. **EVENT LOOP** với chức năng liên tục giám sát xem **CALL STACK** đã trống chưa và **CALLBACK QUEUE** có gì không. Lúc này **CALLBACK QUEUE** đang chờ kết quả mà **WEB APIs** trả về nên và **CALL STACK** lúc này cũng đã trống do toàn bộ code trong chương trình đã được thực hiện nên nó sẽ đẩy kết quả trong **CALLBACK QUEUE** vào lại **CALL STACK** và đoạn code console.log('This is the second line') được thực hiện và trả kết quả ra màn hình.

Kết quả


Call Stack
Web APIs
Browser console
This is the first line
This is the last line
console.log()
Event Loop
Callback Queue
Empty
Call Stack
Web APIs
Browser console
This is the first line
This is the last line
This is the second line
Event Loop
Callback Queue
Empty

PHẦN 2:
XỬ LÝ BẤT ĐỒNG BỘ BẰNG CALLBACK

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT

- Trong javascript mặc định áp dụng xử lý bất đồng bộ (cả trên trình duyệt lẫn nodejs) do đó rất nhiều trường hợp lập trình viên gặp phải tình huống khó xử khi vô tình rơi vào các trường hợp hệ thống tự động xử lý bất đồng bộ, do đó chúng ta sẽ tìm cách để "khử" bất đồng bộ.
- Các phương án xử lý bất đồng bộ:
  - Sử dụng hàm Callback
  - Sử dụng Promise
  - Sử dụng async/await (của es6)

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT – SỬ DUNG CALLBACK

- Định nghĩa hàm callback: Trong lập trình máy tính, callback là một đoạn code chạy được (thường là một hàm A) được sử dụng như tham số truyền vào của hàm B nào đó. Hàm A được gọi ngay lập tức hoặc trễ một chút sau khi hàm B được gọi. Các ngôn ngữ lập trình khác nhau hỗ trợ callback theo các cách khác nhau, thường được triển khai dưới dạng chương trình con, hàm nặc danh, chuỗi lệnh hoặc con trỏ hàm.

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT – SỬ DUNG CALLBACK

- Rất nhiều phương thức built-in trong javascript sử dụng hàm callback cho việc xử lý nghiệp vụ.
- Ví dụ 3:

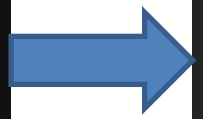
```
// lọc các phần tử là số lẻ
var arr = [4, 3, 6, 2, 5, 11, 17];
arr2 = arr.filter(function(val, index){
    return val%2==1;
});

console.log(arr2);
```

```
▼(4) [3, 5, 11, 17] i
   0: 3
   1: 5
   2: 11
   3: 17
   length: 4
 ▶ __proto__: Array(0)
```

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT – SỬ DUNG CALLBACK

- Các cách để sử dụng hàm callback:
  - 1. Sử dụng anonymous functions (ví dụ 3)
  - 2. Sử dụng 1 hàm đã được đặt tên

```
function getOddNumber(num){
    return num%2==1;
}

// lọc các phần tử là số lẻ
var arr = [4, 3, 6, 2, 5, 11, 17];
arr2 = arr.filter(getOddNumber);

console.log(arr2);
```

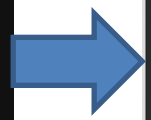


```
▼(4) [3, 5, 11, 17] i
    0: 3
    1: 5
    2: 11
    3: 17
    length: 4
  ▶ __proto__: Array(0)
```

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT – SỬ DUNG CALLBACK

- Tự định nghĩa 1 hàm sử dụng callback: Ta có thể tự tạo 1 hàm sử dụng callback bằng cách nhận tenbien ở tham số sau đó thực thi tenbien() ở nội dung của hàm.
- Ví dụ 4:

```
// định nghĩa hàm xử lý
function getNameCallback(yourName){
    alert("hello, " + yourName);
}

// định nghĩa hàm sẽ nhận hàm callback
function testFunction(callback){
    if(callback) callback();
}

var name = "Tran Thi Loan";
// gọi hàm sử dụng hàm getNameCallback làm tham số
testFunction(getNameCallback(name));
```



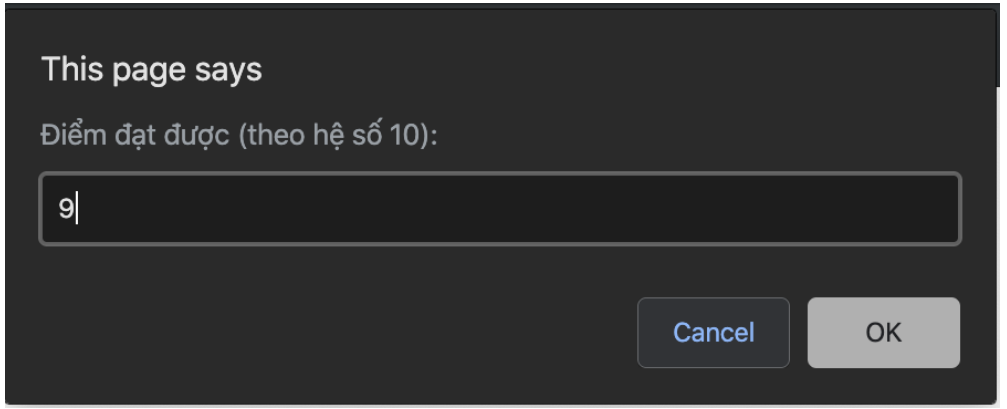
- Ưu điểm của hàm callback:
  - Callback function là một mô hình khá phổ biến nên rất dễ hiểu.
  - Rất dễ triển khai.
- Nhược điểm:
  - Rất dễ xảy ra tình huống các callback lồng nhau dẫn đến tình trạng CallbackHell (sẽ được đề cập đến ở bài tới) – điều này sẽ gây khó khăn khi sửa lỗi và bảo trì.
  - Chỉ có thể truyền một callback cho một sự kiện nhất định, điều này có thể là một giới hạn lớn trong nhiều trường hợp.

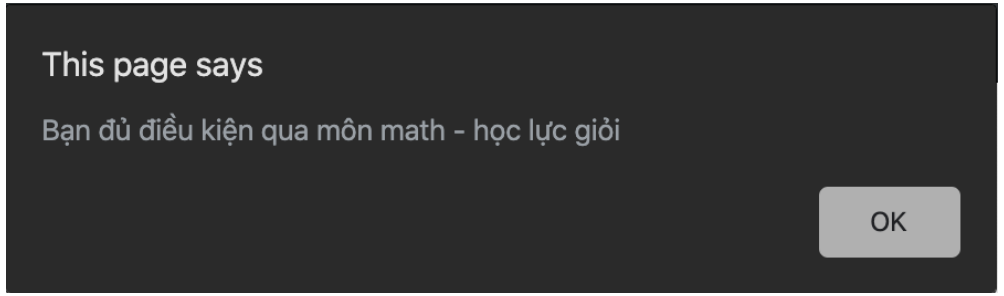
- Ví dụ 5: Viết hàm nhập môn học, sau đó hiển thị alert ra môn đang học, nhập số điểm đạt được, sau đó gọi hàm callback để hiển thị ra học lực của bạn.

FPT Education
FPT POLYTECHNIC

CÁCH XỬ BẤT ĐỒNG BỘ TRONG JAVASCRIPT – SỬ DUNG CALLBACK

```
function doHomework(subject, callback) {
  alert('Môn đang học: ' + subject);
  var score = prompt('Điểm đạt được (theo hệ số 10): ');
  callback(score, subject);
}

function alertFinished(score, subject){
    if(score < 5){
        alert('Bạn không đủ điều kiện qua môn ' + subject);
    }else if(score >= 5 && score < 7){
        alert('Bạn đủ điều kiện qua môn ' + subject
                + ' - học lực trung bình');
    }else if(score >= 7 && score < 9){
        alert('Bạn đủ điều kiện qua môn ' + subject
                + ' - học lực khá');
    }else{
        alert('Bạn đủ điều kiện qua môn ' + subject
                + ' - học lực giỏi');
    }
}
doHomework('math', alertFinished);
```





TỔNG KẾT

- Khái niệm bất đồng bộ là một trong những yếu tố gây khó khăn cho lập trình viên khi làm quen với ngôn ngữ javascript, bài học này đem lại cho chúng ta khái niệm, giải thích cơ chế và cách giải quyết bất đồng bộ cơ bản với callback.

TỔNG KẾT

- ☑ Giải thích được khái niệm bất đồng bộ trong Javascript.
- ☑ Nguyên lý xử lý bất đồng bộ trong javascript
- ☑ Giải thích được khái niệm Callback
- ☑ Sử dụng được Callback


TIN HỌC VĂN PHÒNG QUỐC TẾ THEO CHUẨN ICDL
Syllabus 5 - Office 2007
LUẬT KINH TẾ
LUẬT KINH TẾ
thank you!