**var, let và const trong ES6**

Mấy bài viết nói về 3 từ khóa này trong JavaScript thì có nhiều rồi, mình chỉ tổng hợp lại cho ngắn để các bạn lười đọc tham khảo nhanh thôi 

**let**

let tạo ra một biến chỉ có thể truy cập được trong block bao quanh nó, khác với var - tạo ra một biến có phạm vi truy cập xuyên suốt function chứa nó.

Ví dụ:

Sử dụng var:

function foo() {

var x = 10;

if (true) {

var x = 20; *// x ở đây cũng là x ở trên*

console.log(x); *// in ra 20*

}

console.log(x); *// vẫn là 20*

}

Sử dụng let:

function foo() {

let x = 10;

if (true) {

let x = 20; *// x này là x khác rồi đấy*

console.log(x); *// in ra 20*

}

console.log(x); *// in ra 10*

}

Ngoài ra, khi ở global scope (tức là không nằm trong một function nào cả), từ khóa var tạo ra thuộc tính mới cho global object (this), còn let thì không:

var x = 'global';

let y = 'global';

console.log(this.x); *// "global"*

console.log(this.y); *// undefined*

**Callback và let**

Có một trường hợp dùng let rất hiệu quả đó là sử dụng callback trong một vòng lặp.

Ví dụ nếu dùng var:

for (var i = 0; i < 5; i++) {

setTimeout(function(){

console.log('Yo! ', i);

}, 1000);

}

Kết quả sẽ ra gì nào?

Yo! 5

Yo! 5

Yo! 5

Yo! 5

Yo! 5

Giá trị của biến i bên trong hàm callback luôn là giá trị cuối cùng của i trong vòng lặp.

Để giải quyết vấn đề này, chúng ta thay var bằng let:

for (let i = 0; i < 5; i++) {

setTimeout(function(){

console.log('Yo! ', i);

}, 1000);

}

Output sẽ đúng như mong đợi:

Yo! 0

Yo! 1

Yo! 2

Yo! 3

Yo! 4

**Khi nào, dùng gì?**

Lưu ý là chỉ khi làm việc với ES6 nhé:

* Không dùng var trong bất kì mọi trường hợp
* Thay vào đó thì dùng let