

لها خصوصياتها كذلك `let` و `const`

نجد إمكانية إنشاء المتغيرات باستخدام الكلمتين ES6 من المزايا الجديدة التي جاءت مع إصدار جافا سكريبت. كما كان عليه الحال دائماً في إصدارات جافا سكريبت القديمة `var` وليس فقط `let` و `const` المفتاحيتين.

Hoisting في موضوع `var` لهما سلوك مختلف عن `let` و `const`.

يتم `let` و `const` كقيمة افتراضية في الذاكرة، بينما متغيرات `undefined` كما قلنا يتم إعطاؤها `var` متغيرات وإذا حاولنا الوصول إليها أعلى مكان التصريح بها فإن جافا، `uninitialized` التأشير عليها على أنها غير مهياة أو يتوقف بموجبها البرنامج عن التنفيذ `ReferenceError` سكريبت سيظهر رسالة خطأ

```
console.log(sayHello()); // "hello world"
```

```
console.log(myVar); // undefined
```

```
console.log(letVar); // ReferenceError: Cannot access 'letVar' before initialization
```

```
var myVar = "i am a variable";
```

```
let letVar = "let var";
```

```
function sayHello() {
```

```
    return "hello world";
```

```
}
```

المنطقة قبل مكان التصريح بمتغيرات `const` و `let` في نطاق التنفيذ الخاص بها تسمى `Temporel dead zone`، حيث لا يمكنك استعمال أي من تلك المتغيرات.