

JavaScript Let

- description : JavaScript Let
- author :
- email : shlim@repia.com
- lastupdate : 2021-04-13

ECMAScript 2015

ES2015 let const 가 JavaScript .
 JavaScript (Block Scope) (constant) .
 ES2015 , JavaScript (Global Scope) (Function Scope) 가

Global Scope

(Globally) (Global Scope) 가

```

<!DOCTYPE html>
<html>

<body>
  <h2>JavaScript Scope</h2>
  <p>A Global variable can be accessed from any script or function.</p>
  <p id="demo"></p> <!-- I can display All New
Rexton -->
  <script>
    var carName = "All New Rexton";
    myFunction();

    function myFunction() {
      document.getElementById("demo").innerHTML = "I can display " +
carName;
    }
  </script>
</body>

</html>

```

(Global variables) JavaScript

Function Scope

(Locally)

(Function Scope) 가

```
<!DOCTYPE html>
<html>
<body>
  <h2>JavaScript Scope</h2>
  <p>Outside myFunction() carName is undefined.</p>
  <p id="demo1"></p> // string Volvo
  <p id="demo2"></p> // undefined
  <script>
    myFunction();
    function myFunction() {
      var carName = "Volvo";
      document.getElementById("demo1").innerHTML = typeof carName + "
" + carName;
    }
    document.getElementById("demo2").innerHTML = typeof carName;
  </script>
</body>
</html>
```

(Local variables)

JavaScript Block Scope

var (Block Scope) 가

```
{}
```

```
{
  var x = 2;
}
// x CAN be used here
```

ES2015 JavaScript 가

let 가

```
{}
```

```
{
  let x = 2;
}
// x can Not be used here.
```

Redeclaring Variables

var

가

:

```

<!DOCTYPE html>
<html lang="en">
<body>

  <h2>Declaring a Variable Using var</h2>
  <p id="demo"></p>
  <script>
    var x = 10; // Here x is 10
    {
      var x = 2; // Here x is 2
    }
    document.getElementById("demo").innerHTML = x; // Here x is 2
  </script>
</body>
</html>

```

let

,

.

```

<!DOCTYPE html>
<html lang="en">
<body>
  <h2>Declaring a Variable Using let</h2>
  <p id="demo"></p>
  <script>
    var x = 10;
    {
      let x = 2;
    }
    document.getElementById("demo").innerHTML = x; // Here x is 10
  </script>
</body>
</html>

```

let

Internet Explorer 11

.

Loop Scope

loop

var

.


```
function myFunction() {
  let carName = "Volvo"; // Function Scope
```

Global Scope

var let

(Global Scope)

```
var x = 2; // Global scope
```

```
let x = 2; // Global scope
```

Global Variables in HTML

JavaScript

JavaScript

HTML

window

var

window

```
<!DOCTYPE html>
<html>
<body>
  <h2>JavaScript Global Variables</h2>
  <p>In HTML, global variables defined with <b>var</b> will become window
variables.</p>
  <p id="demo"></p>  <!-- I can display Carnival -->
  <script>
    var carName = "Carnival";
    // code here can use window.carName
    document.getElementById("demo").innerHTML = "I can display " +
window.carName;
  </script>
</body>
</html>
```

let

window

```
<!DOCTYPE html>
<html lang="en">
<body>
  <h2>JavaScript Global Variables</h2>
  <p>In HTML, global variables defined with <b>let</b> will not become
window variables.</p>
```

```
<p id="demo"></p>      <!-- I can not display undefined -->
<script>
  let carName = "Tusan";
  // code here can not use window.carName
  document.getElementById("demo").innerHTML = "I can not display " +
window.carName;
</script>
</body>
</html>
```

Redeclaring

var JavaScript

let var

```
var x = 2;      // Allowed
let x = 3;      // Not allowed

{
  var x = 4;      // Allowed
  let x = 5;      // Not Allowed
}
```

let let

```
let x = 2;      // Allowed
let x = 3;      // Not allowed

{
  let x = 4;      // Allowed
  let x = 5;      // Not allowed
}
```

let

```
<!DOCTYPE html>
<html lang="en">
<body>
  <h2>JavaScript let</h2>
  <p>Reclaring a variable with <b>let</b> in another scope or in another
block is allowed.</p>
  <p id="demo"></p>
  <script>
```

```

    let x = 2; // Allowed

    {
      let x = 3; // Allowed
    }

    {
      let x = 4; // Allowed
    }
    document.getElementById("demo").innerHTML = x;
  </script>
</body>
</html>

```

Hoisting

var

hoisting:

```

<!DOCTYPE html>
<html>
<body>
  <h2>JavaScript Hoisting</h2>
  <p>With <b>var</b>, you can use a variable before it is declared:</p>
  <p id="demo"></p>
  <script>
    carName = "Avante";
    document.getElementById("demo").innerHTML = carName;
    var carName;
  </script>
</body>
</html>

```

let

:

let

ReferenceError가

“TDZ(Temporal Dead Zone)”

ReferenceError

```

<!DOCTYPE html>

```

```
<html>
<body>
  <h2>JavaScript Hoisting</h2>
  <p>With <b>let</b>, you can use a variable before it is declared:</p>
  <p id="demo"></p>
  <script>
    try {
      carName = "Avante";
      let carName;
      document.getElementById("demo").innerHTML = carName;
    }
    catch (err) {
      document.getElementById("demo").innerHTML = err.name + " : " +
err.message;
    }
  </script>
</body>
</html>
```

„ Javascript, Let

From:
<https://125.132.25.164/dokuwiki/> -
2023.12

Permanent link:
https://125.132.25.164/dokuwiki/doku.php?id=wiki:javascript:javascript_note:js_let

Last update: **2023/01/13 18:44**

