JavaScript Const

❮ Previous Next ❯

The `const` keyword was introduced in ES6 (2015).

Variables defined with `const` cannot be Redeclared.

Variables defined with `const` cannot be Reassigned.

Variables defined with `const` have Block Scope.

Cannot be Reassigned

A `const` variable cannot be reassigned:

Example

```
const PI = 3.141592653589793;
```

Try it Yourself »

Must be Assigned

JavaScript `const` variables must be assigned a value when they are declared:

Correct

```
const PI = 3.14159265359;
```

Incorrect

```
const PI;
PI = 3.14159265359;
```

When to use JavaScript const?

As a general rule, always declare a variable with `const` unless you know that the value will change.

Use `const` when you declare:

- A new Array
- A new Object
- A new Function
- A new RegExp

Constant Objects and Arrays

The keyword `const` is a little misleading.

It does not define a constant value. It defines a constant reference to a value.

Because of this you can NOT:

- Reassign a constant value
- Reassign a constant array
- Reassign a constant object

But you CAN:

- Change the elements of constant array
- Change the properties of constant object

Constant Arrays

You can change the elements of a constant array:

Example

```
// You can change an element:
cars[0] = "Toyota";

// You can add an element:
cars.push("Audi");
```

Try it Yourself »

But you can NOT reassign the array:

Example

```
const cars = ["Saab", "Volvo", "BMW"];

cars = ["Toyota", "Volvo", "Audi"];    // ERROR
```

Try it Yourself »

Constant Objects

You can change the properties of a constant object:

Example

```
// You can create a const object:
const car = {type:"Fiat", model:"500", color:"white"};

// You can change a property:
car.color = "red";
```

```
car.owner = "Johnson";
```

Try it Yourself »

But you can NOT reassign the object:

Example

```
const car = {type:"Fiat", model:"500", color:"white"};

car = {type:"Volvo", model:"EX60", color:"red"};    // ERROR
```

Try it Yourself »

Browser Support

The `const` keyword is not supported in Internet Explorer 10 or earlier.

The following table defines the first browser versions with full support for the `const` keyword:

Chrome 49	IE 11 / Edge	Firefox 36	Safari 10	Opera 36
Mar, 2016	Oct, 2013	Feb, 2015	Sep, 2016	Mar, 2016

Block Scope

Scope.

The x declared in the block, in this example, is not the same as the x declared outside the block:

Example

```
const x = 10;
// Here x is 10

{
const x = 2;
// Here x is 2
}

// Here x is 10
```

Try it Yourself »

You can learn more about block scope in the chapter JavaScript Scope.

Redeclaring

Redeclaring a JavaScript `var` variable is allowed anywhere in a program:

Example

```
var x = 2;      // Allowed
var x = 3;      // Allowed
x = 4;          // Allowed
```

Redeclaring an existing `var` or `let` variable to `const`, in the same scope,

Example

```
var x = 2;     // Allowed
const x = 2;   // Not allowed

{
let x = 2;     // Allowed
const x = 2;   // Not allowed
}

{
const x = 2;   // Allowed
const x = 2;   // Not allowed
}
```

Reassigning an existing `const` variable, in the same scope, is not allowed:

Example

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

{
  const x = 2;   // Allowed
  x = 2;         // Not allowed
  var x = 2;     // Not allowed
  let x = 2;     // Not allowed
  const x = 2;   // Not allowed
}
```

Redeclaring a variable with `const` , in another scope, or in another block, is

Example

```
const x = 2;       // Allowed

{
  const x = 3;   // Allowed
}

{
  const x = 4;   // Allowed
}
```

Const Hoisting

Variables defined with `var` are **hoisted** to the top and can be initialized at any time.

Meaning: You can use the variable before it is declared:

Example

This is OK:

```
carName = "Volvo";
var carName;
```

Try it Yourself »

If you want to learn more about hoisting, study the chapter JavaScript Hoisting.

Variables defined with `const` are also hoisted to the top, but not initialized.

`ReferenceError`:

Example

```
alert (carName);
const carName = "Volvo";
```

Try it Yourself »

❮ Previous Next ❯