IE Masters in Finance
Specialize in FinTech and build a profile that stands out

JavaScript Const

❮ Previous Next ❯

ECMAScript 2015

ES2015 introduced two important new JavaScript keywords: `let` and `const`.

Variables defined with `const` behave like `let` variables, except they cannot be reassigned:

Example

```
const PI = 3.141592653589793;
PI = 3.14;      // This will give an error
PI = PI + 10;   // This will also give an error
```

Try it Yourself »

Block Scope

Declaring a variable with `const` is similar to `let` when it comes to **Block Scope**.

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

Example

```
var 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 previous chapter: JavaScript Let.

Assigned when Declared

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

Incorrect

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

Correct

```
const PI = 3.14159265359;
```

Not Real Constants

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, we cannot change constant primitive values, but we can change the properties of constant objects.

Primitive Values

If we assign a primitive value to a constant, we cannot change the primitive value:

Example

```
const PI = 3.141592653589793;
PI = 3.14;      // This will give an error
PI = PI + 10;   // This will also give an error
```

Try it Yourself »

Constant Objects can Change

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";

// You can add a property:
car.owner = "Johnson";
```

Try it Yourself »

But you can NOT reassign a constant object:

Example

```
const car = {type:"Fiat", model:"500", color:"white"};
car = {type:"Volvo", model:"EX60", color:"red"};    // ERROR
```

Try it Yourself »

Constant Arrays can Change

You can change the elements of a constant array:

Example

```
// You can create a constant array:
const cars = ["Saab", "Volvo", "BMW"];

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

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

Try it Yourself »

But you can NOT reassign a constant array:

Example

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

```
cars = ["Toyota", "Volvo", "Audi"];    // 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 / Edge 11	Firefox 36	Safari 10	Opera 36
Mar, 2016	Oct, 2013	Feb, 2015	Sep, 2016	Mar, 2016

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 or reassigning an existing `var` or `let` variable to `const`, in the same scope, or in the same block, is not allowed:

Example

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

Redeclaring or reassigning an existing `const` variable, in the same scope, or in the same block, is not allowed:

Example

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

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

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

Example

```
const x = 2;       // Allowed

{
  const x = 3;   // Allowed
}

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

Hoisting

Variables defined with `var` are **hoisted** to the top (if you don't know what Hoisting is, read our Hoisting Chapter).

You can use a `var` variable before it is declared:

Example

```
carName = "Volvo";    // You CAN use carName here
var carName;
```

Try it Yourself »

Variables defined with `const` are not hoisted to the top.

A `const` variable cannot be used before it is declared:

Example

```
carName = "Volvo";    // You can NOT use carName here
const carName = "Volvo";
```

❮ Previous

Next ❯

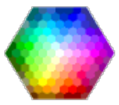
Qixing Welding Material

coowor

Produce copper-based, silver-based, aluminum-based brazing & soldering materials, flux .

OPEN

COLOR PICKER



HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes