

Module 模块

ES6

- [import 语句](#)
 - [export 语句](#)
 - [export default 语句](#)
 - [import 语句](#)
 - [import 语句](#)
 - [export default 语句](#)
 - [export 语句 import 语句](#)
 - [import 语句](#)
 - [import 语句](#)
 - [import 语句](#)
 - [import\(\)](#)
 - [import 语句](#)
 - [import 语句](#)
 - [import 语句](#)
 - [import.meta](#)
-

ES6

ES6 模块 JavaScript 模块化的 module

ES6 模块 JavaScript 模块化的 module

ES6 模块 JavaScript 模块化的 module

ES6 模块 JavaScript 模块化的 module

ES6 模块 JavaScript 模块化的 module

CommonJS 模块 AMD

ES6 模块 JavaScript 模块化的 module

ES6 模块 JavaScript 模块化的 module

CommonJS 模块 AMD

ES6 模块 JavaScript 模块化的 module

ES6 模块化

CommonJS 与 AMD 模块化

CommonJS

```
// CommonJS 模块化
let { stat, exists, readFile } = require('fs');

// 使用
let _fs = require('fs');
let stat = _fs.stat;
let exists = _fs.exists;
let readFile = _fs.readFile;
```

模块化 3 种方式 “ ” “ ”

ES6 模块化 export import

```
// ES6 模块化
import { stat, exists, readFile } from 'fs';
```

模块化 fs 3 种方式 “ ”

ES6 模块化 CommonJS 模块化

ES6 模块化

ES6 模块化 JavaScript 模块化 macro type system

ES6 模块化

- UMD 模块化 ES6 模块化
- API navigator
- Math

ES6 模块化 Node ES6 模块化



ES6 模块化 "use strict";

- _____
- _____
- _____ with _____
- _____
- _____ 0 _____
- _____
- _____ delete prop _____ delete global[prop]
- eval _____
- eval _____ arguments _____
- arguments _____
- _____ arguments.callee
- _____ arguments.caller
- _____ this _____
- _____ fn.caller _____ fn.arguments _____
- _____ protected _____ static _____ interface _____

_____ ES5 _____ ES6 _____ ES5

_____ this _____ ES6 _____ this _____ undefined _____
this _____

export _____

_____ export _____ import _____ export _____ import

_____ export _____ JS _____ export _____

```
// profile.js
export var firstName = 'Michael';
export var lastName = 'Jackson';
export var year = 1958;
```

_____ profile.js _____ ES6 _____ export

export _____

```
// profile.js
var firstName = 'Michael';
var lastName = 'Jackson';
var year = 1958;

export { firstName, lastName, year };
```

import export var

export class

```
export function multiply(x, y) {
  return x * y;
};
```

multiply

export as

```
function v1() { ... }
function v2() { ... }

export {
  v1 as streamV1,
  v2 as streamV2,
  v2 as streamLatestVersion
};
```

as v1 v2 v2

export

```
// 
export 1;

// 
var m = 1;
export m;
```

Module 1

1

m

1

```
// 模块 1  
export var m = 1;
```

```
// 模块 2  
var m = 1;  
export {m};
```

```
// 模块 3  
var n = 1;  
export {n as m};
```

Module 1

m

1

Module 2

function class

```
// 模块 1  
function f() {}  
export f;  
  
// 模块 2  
export function f() {};  
  
// 模块 3  
function f() {}  
export {f};
```

export

Functions

Classes

var

let

const

Variables

export

```
export var foo = 'bar';  
setTimeout(() => foo = 'baz', 500);
```

Module 1

foo

bar

500

baz

CommonJS

CommonJS

Module

export
import ES6

```
function foo() {  
  export default 'bar' // SyntaxError  
}  
foo()
```

export

import

export JS import

```
// main.js  
import { firstName, lastName, year } from './profile.js';  
  
function setName(element) {  
  element.textContent = firstName + ' ' + lastName;  
}
```

import profile.js import
profile.js

import as

```
import { lastName as surname } from './profile.js';
```

import
a

```
import {a} from './xxx.js'  
  
a = {}; // Syntax Error : 'a' is read-only;
```

a a a

```
import {a} from './xxx.js'
```

```
a.foo = 'hello'; // 输出
```

```
import {a} from './xxx.js'
// 输出
a.foo
```

```
import {a} from './xxx.js'
// 输出
a.foo
```

```
import { myMethod } from 'util';
```

```
import { myMethod } from 'util';
```

```
import { myMethod } from 'util';
```

```
foo();
```

```
import { foo } from 'my_module';
```

```
import { foo } from 'my_module';
// 输出
foo
```

```
import { foo } from 'my_module';
```

```
// 输出
import { 'f' + 'oo' } from 'my_module';
```

```
// 输出
let module = 'my_module';
import { foo } from module;
```

```
// 输出
if (x === 1) {
  import { foo } from 'module1';
} else {
  import { foo } from 'module2';
}
```

```
if (x === 1) {
  import { foo } from 'module1';
} else {
  import { foo } from 'module2';
}
```

import

```
import 'lodash';
```

lodash

import

```
import 'lodash';
import 'lodash';
```

lodash

```
import { foo } from 'my_module';
import { bar } from 'my_module';

// 
import { foo, bar } from 'my_module';
```

foo bar my_module

import Singleton

Babel CommonJS require ES6 import

import

```
require('core-js/modules/es6.symbol');
require('core-js/modules/es6.promise');
import React from 'React';
```



*

circle.js area circumference

```
// circle.js

export function area(radius) {
  return Math.PI * radius * radius;
```

```
}

export function circumference(radius) {
  return 2 * Math.PI * radius;
}
```

```
// main.js

import { area, circumference } from './circle';

console.log('Area: ' + area(4));
console.log('Circumference: ' + circumference(14));
```

```
import * as circle from './circle';

console.log('Area: ' + circle.area(4));
console.log('Circumference: ' + circle.circumference(14));
```



```
import * as circle from './circle';

// 円周率
circle.foo = 'hello';

circle.area = function () {};
```

export default ☐

[illegible][illegible]

```
export default
```

```
// export-default.js
export default function () {
  console.log('foo');
}
```

□□□□□□□□

export-default.js □□□□□□□□

□□□□□□□□

import □□□□□□□□□□□□

```
// import-default.js
import customName from './export-default';
customName(); // 'foo'
```

□□□□

import □□□□□□□□

export-default.js

□□□□□□□□□□□□□□□□□□□□□□□□□□□□

import □□□□□□□□□□

export default □□□□□□□□□□□□□□

```
// export-default.js
export default function foo() {
  console.log('foo');
}

// 模块

function foo() {
  console.log('foo');
}

export default foo;
```

□□□□□

foo □□□□□

foo □□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□

```
// 模块
export default function crc32() { // 模块
  // ...
}

import crc32 from 'crc32'; // 模块
```

```
// 测试
export function crc32() { // 测试
  // ...
};

import {crc32} from 'crc32'; // 测试
```

Diagram illustrating the evolution of a code snippet across four rows, showing the addition of characters (represented by boxes) to the code:

- Row 1: `export default` (4 characters) `import`
- Row 2: `export default` (4 characters) `import` (8 characters)
- Row 3: `export default` (24 characters) `import` (16 characters) `export default` (3 characters)
- Row 4: `export default` (6 characters) `default` (24 characters)

```
// modules.js

function add(x, y) {
  return x * y;
}

export {add as default};

// app.js

import { default as foo } from 'modules';

// import foo from 'modules';
```

■■■■ export default ■■■■■■■■■■ default ■■■■■■■■■■■■■■■■■■■■

```
// ❶
export var a = 1;

// ❷
var a = 1;
export default a;

// ❸
export default var a = 1;
```

export default

default

export default

```
//   
export default 42;
```

```
//   
export 42;
```

default

export default

lodash

```
import _ from 'lodash';
```

import

```
import _, { each, forEach } from 'lodash';
```

export

```
export default function (obj) {  
  // ...  
}  
  
export function each(obj, iterator, context) {  
  // ...  
}  
  
export { each as forEach };
```

forEach

each

forEach each

export default

```
// MyClass.js  
export default class { ... }  
  
// main.js  
import MyClass from 'MyClass';
```

```
let o = new MyClass();
```

export □ import □□□□

```
import
```

export

--	--	--	--	--	--

```
export { foo, bar } from 'my_module';

// 再export一次

import { foo, bar } from 'my_module';
export { foo, bar };
```

// □□□□□□

```
import { foo, bar } from 'my_module';
```

```
export { foo, bar };
```

export

```
import
```

foo

bar

foo

bar

```
// 名前空間を再エクスポート
export { foo as myFoo } from 'my_module';

// 名前空間を再エクスポート
export * from 'my_module';
```

```
export { foo as myFoo } from 'my_module';
```

//

```
export * from 'my_module';
```

```
export { default } from 'foo';
```

```
export { es6 as default } from './someModule';

// 空行

import { es6 } from './someModule';
export default es6;
```

//

```
import { es6 } from './someModule';
```

```
export default es6;
```

```
export { default as es6 } from './someModule';
```

ES2020

import

```
import * as someIdentifier from "someModule";
```

ES2020

```
export * as ns from "mod";
```

```
//
```

```
import * as ns from "mod";
```

```
export {ns};
```

import

ES2025 “import” import attributes

import

JSON

WebAssembly

CSS

JSON

```
import configData from './config-data.json' with { type: 'json' };
```

import with

type

json

import()

```
try {
  const configData = await import(
    './config-data.json', { with: { type: 'json' } }
  );
} catch(error) {
  //
}
```

import JSON fetch

```
const response = await fetch('./config.json');
const json = await response.json();
```



```
// circleplus.js

export { area as circleArea } from 'circle';
```

```
// circleplus.js

export { area as circleArea } from 'circle';
```

 circle area circleArea

```
// main.js

import * as math from 'circleplus';

import exp from 'circleplus';

console.log(exp(math.e));
```

```
// main.js

import * as math from 'circleplus';

import exp from 'circleplus';

console.log(exp(math.e));
```

```
// main.js

import * as math from 'circleplus';

import exp from 'circleplus';

console.log(exp(math.e));
```

```
// main.js

import * as math from 'circleplus';

import exp from 'circleplus';

console.log(exp(math.e));
```

import exp circleplus exp

import exp circleplus exp

--	--	--	--	--

`const` `const`

`const` `const`

[illegible][illegible]

```
// constants.js
export const A = 1;
export const B = 3;
export const C = 4;

// test1.js
import * as constants from './constants';
console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// constants.js
export const A = 1;
export const B = 3;
export const C = 4;

// test1.js
import * as constants from './constants';
console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// constants.js
export const A = 1;
export const B = 3;
export const C = 4;

// test1.js
import * as constants from './constants';
console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// constants.js
export const A = 1;
export const B = 3;
export const C = 4;

// test1.js
import * as constants from './constants';
console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// test1.js
import * as constants from './constants';

console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';

console.log(A); // 1
console.log(B); // 3
```

```
// test1.js
import * as constants from './constants';

console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';

console.log(A); // 1
console.log(B); // 3
```

```
// test1.js
import * as constants from './constants';

console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';

console.log(A); // 1
console.log(B); // 3
```

```
// test1.js
import * as constants from './constants';

console.log(constants.A); // 1
console.log(constants.B); // 3

// test2.js
import {A, B} from './constants';

console.log(A); // 1
console.log(B); // 3
```

```
// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

```
// test2.js
import {A, B} from './constants';
console.log(A); // 1
console.log(B); // 3
```

constants

constants


```
// constants/db.js
export const db = {
  url: 'http://my.couchdbserver.local:5984',
  admin_username: 'admin',
  admin_password: 'admin password'
};

// constants/user.js
export const users = ['root', 'admin', 'staff', 'ceo', 'chief', 'moderator'];
```

index.js

```
// constants/index.js
export {db} from './db';
export {users} from './users';
```

index.js

```
// script.js
import {db, users} from './constants/index';
```

import()

--	--

import JavaScript import " binding

```
// []  
if (x === 2) {  
  import MyModual from './myModual';  
}
```

```

import module1
module1.function1()
if True:
    module1.function2()

```

The diagram illustrates the memory layout of a Node object. The top bar represents the Node object, which contains various fields. The bottom bar shows the layout of the 'require' property, which is a pointer to another object.


```
const widget = await import('./widget.js');
widget.render(container);
}
}

renderWidget();
```

`then()`

```
import()
```

```
button.addEventListener('click', event => {
  import('./dialogBox.js')
  .then(dialogBox => {
    dialogBox.open();
  })
  .catch(error => {
    /* Error handling */
  })
});
```

```
if [ ]
```

```
if (condition) {
  import('moduleA').then(...);
} else {
  import('moduleB').then(...);
}
```

[illegible]

import()

```
import(f())  
.then(...);
```

f

import()

then

```
import('./myModule.js')  
.then(({export1, export2}) => {  
  // ...  
});
```

export1

export2

myModule.js

default

```
import('./myModule.js')  
.then(myModule => {  
  console.log(myModule.default);  
});
```

```
import('./myModule.js')  
.then(({default: theDefault}) => {  
  console.log(theDefault);  
});
```

```
Promise.all([  
  import('./module1.js'),  
  import('./module2.js'),  
  import('./module3.js'),  
)  
.then([module1, module2, module3]) => {
```

```
...  
});
```

import() async

```
async function main() {  
  const myModule = await import('./myModule.js');  
  const {export1, export2} = await import('./myModule.js');  
  const [module1, module2, module3] =  
    await Promise.all([  
      import('./module1.js'),  
      import('./module2.js'),  
      import('./module3.js'),  
    ]);  
}  
main();
```

import.meta

[ES2020](#) ☐ import

import.meta

import.meta

import.meta

☐ 1 ☐ import.meta.url

import.meta.url URL

https://foo.com/main.js

☐ import.meta.url

data.txt

```
new URL('data.txt', import.meta.url)
```

☐ Node.js ☐ import.meta.url

file:URL

file:///home/user/foo.js ☐

☐ 2 ☐ import.meta.scriptElement

import.meta.scriptElement

<script>

document.currentScript

```
// HTML
```

```
// <script type="module" src="my-module.js" data-foo="abc"></script>
```

```
// my-module.js
```

```
import.meta.scriptElement.dataset.foo
```

```
// "abc"
```

3

Deno

import.meta.filename

import.meta.dirname

CommonJS

__filename

__dirname

- import.meta.filename
- import.meta.dirname

Linux

/dev/my_module.ts Windows

C:\dev\my_module.ts

#1

2020-04-16 10:29:52 UTC

2026-09-24 22:20:22 UTC