ES6模块

export命令: export命令用于规定模块的对外接口。

export输出变量：

Var name1 = ‘yu’;

Var name2 = ‘zhen’;

Var name3 = ‘fan’;

Export {name1, name2, name3};

export命令除了输出变量，还可以输出函数或类（class）。

export函数：

Export function multiply( x,y ){ return x\*y }

export为输出变量改名：

Export { v1 as name1, v2 as name2 }//此时的v1可以通过v1和name1调用。

Import命令：其他 JS 文件就可以通过import命令加载这个模块。

Import 加载变量

Import ｛ name1, name2 ｝ form’./main’;

Import为加载的变量改名

Import { name1 as firstname } form’./main’;

模块的整体加载：即用星号（\*）指定一个对象

Import \* as circle from ’./main’;

Export dedault 命令： 为模块指定默认输出。

Export default function（）{ } //其实相当于default为变量

Import name from ’./main’; //不需要使用{}

Import name { each } from ’./main’; //加载默认模块name 和 each模块

Export 与 import 的复合写法：

模块的继承：

ES6模块加载的实质：ES6 模块是动态引用，并且不会缓存值，模块里面的变量绑定其所在的模块。

浏览器的模块加载：

<script type=”module” src=”foo.js”></script>

NODE的加载处理：如果不输出任何接口，但是希望被 Node 认为是 ES6 模块，可以在脚本中加一行语句。 Ecport {}；

Import命令加载commonjs模块：Node 会自动将module.exports属性，当作模块的默认输出，即等同于export default。

Require命令加载ES6模块：const es\_namespace = require(‘./es’);

Require命令加载ES6模块：

ES6模块的循环加载：

跨模块常量：

export const users = ['root', 'admin', 'staff', 'ceo', 'chief', 'moderator'];

import {db, users} from './constants';

Import（）：有一个[提案](https://github.com/tc39/proposal-dynamic-import)，建议引入import()函数，完成动态加载。