模組化

32

```
<!DOCTYPE html>
<html>
<head>
   <script src=“/global_lib/aaa.js"></script>
   <script src="/global_lib/bbb.js"></script>
   <script src="/global_lib/ccc.js"></script>

   <script src=“/index_lib/001.js"></script>
   <script src="/index_lib/002.js"></script>
   <script src="/index_lib/003.js"></script>

   ...
</head>
<body>
   ...
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
   <script src="/global_lib.js"></script>


   <script src="/index_lib.js"></script>


   ...
</head>
<body>
   ...
</body>
</html>
```

1

將共用js包成一個js

1

2

將特定頁面用之js包成一個js

1

2

模組化

- ES5
  - **Require JS**: 用js的方式來進行模組化

- ES6
  - **import**, **export**: 原生支持模組化
  - 導致require js的熱度降低
  - 不支持ES6之瀏覽器無法使用
  - 模組化後需要進行優化
    => 使得webpack、babel等js興起

模組化

```
<!DOCTYPE html>
<html>
<head>
1  <script src="/global_lib/aaa.js"></script>
   <script src="/global_lib/bbb.js"></script>
   <script src="/global_lib/ccc.js"></script>

2  <script src="/index_lib/001.js"></script>
   <script src="/index_lib/002.js"></script>
   <script src="/index_lib/003.js"></script>

   ...
</head>
<body>
   ...
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
1  <script src="/global_lib.js"></script>

2  <script src="/index_lib.js"></script>

   ...
</head>
<body>
   ...
</body>
</html>
```

1 將共用js包成一個js

2 將特定頁面用之js包成一個js