

## Модулі у JS

Модуль – це окрема програмна одиниця (окремий файл), яка може бути використана (підключена) при реалізації інших модулів

## Проблема використання глобальних змінних

JS mod1.js global\_issue X

global\_issue > JS mod1.js > ...

```
1 var someVar = 10
```

```
2
```

JS mod2.js global\_issue X

global\_issue > JS mod2.js > ...

```
1 var someVar = 'Hello'
```

```
2
```

<> index.html global\_issue X

global\_issue > <> index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible"
6       content="IE=edge" />
7     <meta name="viewport"
8       content="width=device-width, initial-scale=1.0" />
9   </head>
10  <body>
11    <script src="./mod1.js"></script>
12    <script>
13      console.log(someVar)
14    </script>
15  </body>
16</html>
```

someVar = 10

# Проблема використання глобальних змінних

JS mod1.js global\_issue X

global\_issue > JS mod1.js > ...

```
1 var someVar = 10
2
```

У двох файлах  
описано змінні з  
однаковими  
іменами

JS mod2.js global\_issue X

global\_issue > JS mod2.js > ...

```
1 var someVar = 'Hello'
2
```

<> index.html global\_issue X

global\_issue > <> index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible"
6       content="IE=edge" />
7     <meta name="viewport"
8       content="width=device-width, initial-scale=1.0" />
9   </head>
10  <body>
11    <script src="./mod1.js"></script>
12    <script src="./mod2.js"></script>
13    <script>
14      console.log(someVar)
15    </script>
16  </body>
17 </html>
```

someVar = 10

someVar = 'Hello'

*Проблема використання глобальних змінних. Раніше вирішували з використанням функцій з самовикликом*

**Опис локальних в межах модуля змінних і функцій**

Загальна форма опису функцій з самовикликом	Приклад
<pre>;(function() {      // Опис локальних змінних та     // та функцій, які доступні     // тільки всередині цієї функції  })();</pre>	<pre>===== MyScript.js ===== ; (function() {      //Локальні змінні та функції     var privateVariable;     function privateFunc() { ... }     ...      //Виклик локальних змінних     privateFunc()     . . .  })();</pre> <pre>===== MyScript.js ===== //Змінні та функції не будуть //доступні у цьому файлі  &lt;script src=" MyScript.js"&gt; &lt;/script&gt;</pre>

*Проблема використання глобальних змінних. Раніше вирішували з використанням функцій з самовикликом*

*Повернення змінних та функцій з модуля через глобальний об'єкт window*

<div>Повернення змінних та об'єктів з модуля, як властивостей глобального об'єкта window</div> <div>===== MyScript.js =====</div> <div><pre>; (function() {   //Локальні змінні та функції   var privateVariable;   function privateFunc() { ... }   ...    // - - - - -   //Те, що повертаємо   window.var1=value1;   window.var1=value1;   . . .    window.func1= function() { },   window.func2= function() { },   . . . })();</pre></div>	<div>Підключення модуля та використання доданих у глобальний об'єкт window змінних та функцій</div> <div>===== index.html =====</div> <div><pre>&lt;script src=" MyScript.js"&gt;&lt;/script&gt;  &lt;script&gt;   // Викорст.глобальної змінної   var a= var1;   //Виклик глобальної функції   func1();  &lt;/script&gt;</pre></div>
---	---

*Проблема використання глобальних змінних. Раніше вирішували з використанням функцій з самовикликом*

*Повернення змінних та функцій з модуля через властивості об'єкта*

Повернення змінних та об'єктів з модуля, як властивостей експортованого об'єкта	Підключення модуля та використання доданих у об'єкт змінних та функцій
<pre>===== MyScript.js =====  var expObject = (function() {   //Локальні змінні та функції   var privateVariable;   function privateFunc() { ... }   ...    // - - - - -   //Об'єкт, що повертаємо   return {     var1:value1,     var2:value2,     . . .     func1: function() { },     func2: function() { },     . . .   } })();</pre>	<pre>===== index.html =====  &lt;script src=" MyScript.js"&gt;&lt;/script&gt;  &lt;script&gt;  // Викорст. змінної var a= expObject.var1;  //Виклик функції expObject.func1();  &lt;/script&gt;</pre>

Модуль – це окрема програмна одиниця (окремий файл), яка може бути використана (підключена) при реалізації інших модулів

Особливості сучасних модулів

- створюють власну область видимості
- усі змінні, функції – локальні
- для використання ззовні треба експортувати
- виконуються в режимі *strict mode*
- завантажуються тільки один раз
- завантажуються асинхронно
- можуть підключати інші модулі
- *this* невизначено
- для підключення треба використати `<script type="module" src="...шлях..."></script>`
- сторінка повинна бути на сервері (можна використати розширення Live Server)

```
JS module1.js module_example X ... JS main.js module_example X
module_example > JS module1.js > ...
1 export function sayHi(user) {
2   return `Hello, ${user}!`
3 }
4

module_example > JS main.js
1 import { sayHi } from './module1.js'
2
3 document.body.innerHTML = sayHi('John')
4

<> index.html module_example ●
module_example > <> index.html > html > head > script
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <title>Document</title>
8     <script type="module" src="./main.js"></script>
9   </head>
10  <body></body>
11 </html>
```

# Імпорт модулів

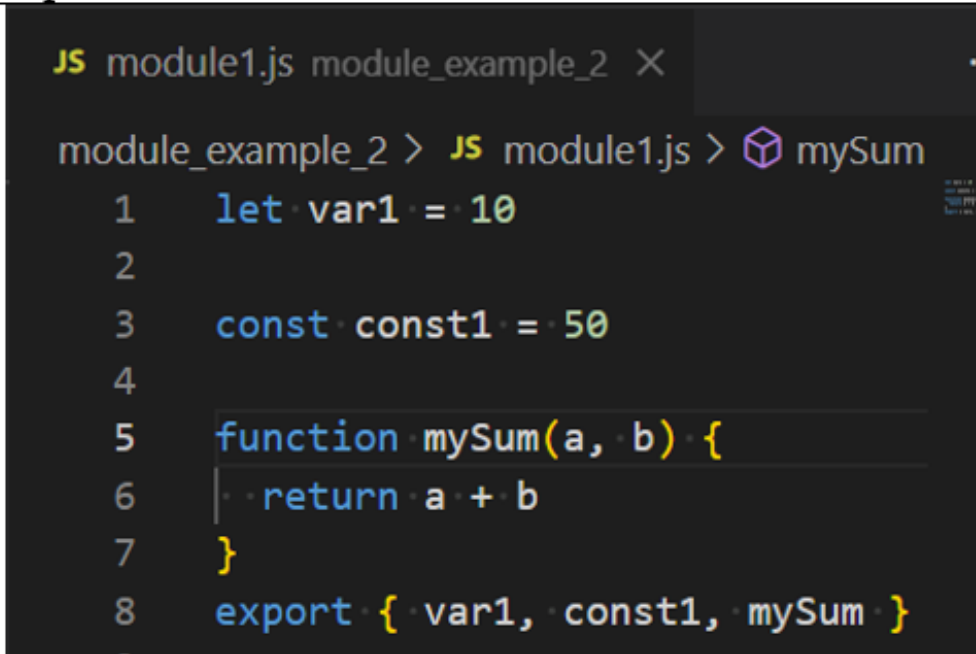
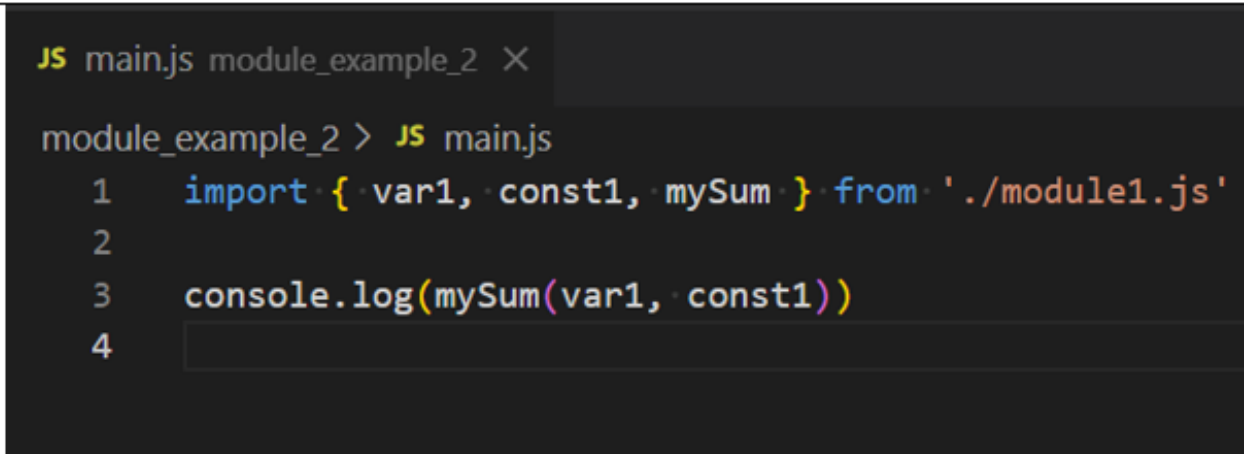
Загальна форма	Приклад
<pre>import 'шлях_до_модуля'</pre>	<div data-bbox="723 158 2104 682"><pre>JS main.js module_example X  module_example &gt; JS main.js 1 document.body.innerHTML = 'Hello' 2</pre></div> <div data-bbox="723 682 2104 1076"><pre>JS main.js module_example_2 ●  module_example_2 &gt; JS main.js 1 import './module1.js' 2</pre></div>



## Експорт при описі

Загальна форма	Приклад
<p>=== експорт величини ===</p> <p><i>export</i> <span>опис_змінної</span></p> <p><i>export</i> <span>опис_константи</span></p> <p><i>export</i> <span>опис_функції</span></p>	<pre>JS module1.js module_example_2 × .. module_example_2 &gt; JS module1.js &gt; mySum 1   export let var1 = 10 2 3   export const const1 = 50 4 5   export function mySum(a, b) { 6     return a + b 7   }</pre>
<p>=== імпорт величин ===</p> <p>import {   назва_змінної ,    назва_константи ,    назва_функції , } from ' шлях_до_модуля '</p>	<pre>JS main.js module_example_2 × module_example_2 &gt; JS main.js 1   import { var1, const1, mySum } from './module1.js' 2 3   console.log(mySum(var1, const1)) 4</pre>

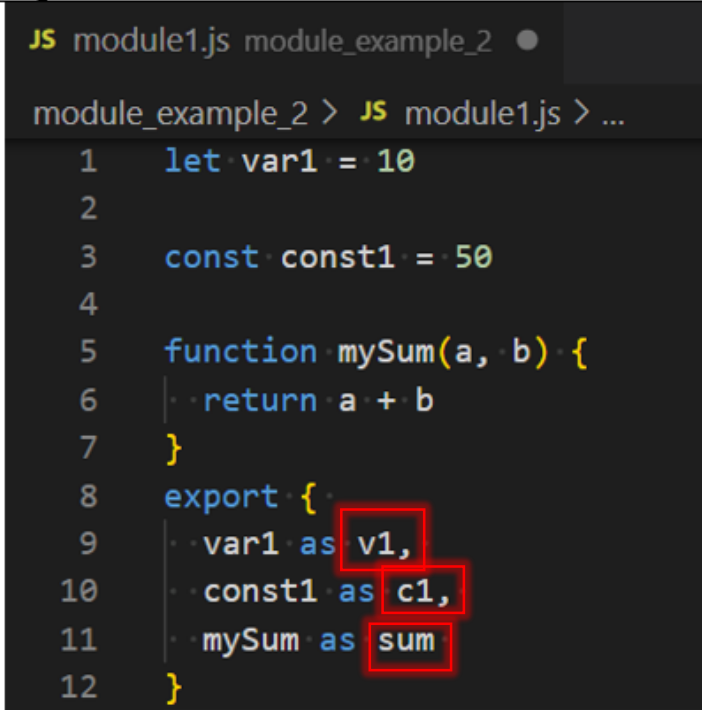
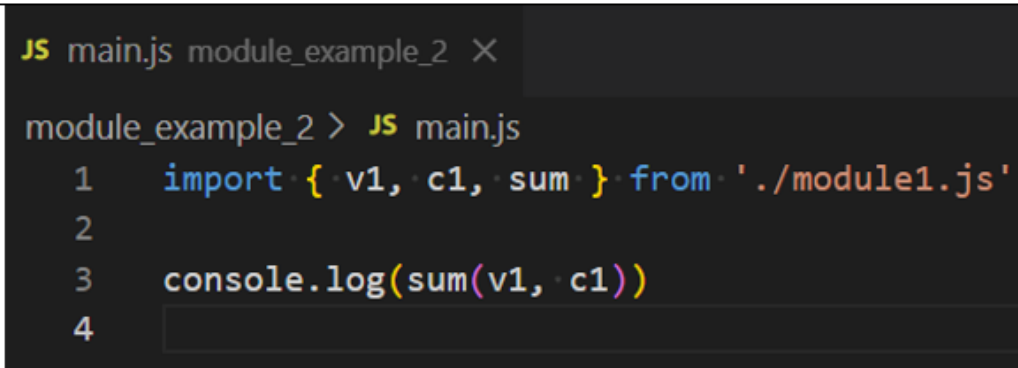
# Експорт величин після опису

Загальна форма	Приклад
<p>=== експорт величини ===</p> <p><code>опис_змінної</code></p> <p><code>опис_константи</code></p> <p><code>опис_функції</code></p> <p><code>export { змінна, константа, функція }</code></p>	 <pre>JS module1.js module_example_2 X module_example_2 &gt; JS module1.js &gt; mySum 1 let var1 = 10 2 3 const const1 = 50 4 5 function mySum(a, b) { 6   return a + b 7 } 8 export { var1, const1, mySum }</pre>
<p>=== імпорт величин ===</p> <p><code>import {</code>     <code>назва_змінної ,</code>      <code>назва_константи ,</code>      <code>назва_функції ,</code> <code>}</code> <code>from ' шлях_до_модуля '</code></p>	 <pre>JS main.js module_example_2 X module_example_2 &gt; JS main.js 1 import { var1, const1, mySum } from './module1.js' 2 3 console.log(mySum(var1, const1)) 4</pre>

## Використання псевдонімів під час імпорту

Загальна форма	Приклад
<p>=== експорт величини ===</p> <p>опис_змінної</p> <p>опис_константи</p> <p>опис_функції</p> <pre>export {   змінна,    константа,    функція,   . . . . }</pre>	<pre>JS module1.js module_example_2 ×  module_example_2 &gt; JS module1.js &gt; ... 1   let var1 = 10 2 3   const const1 = 50 4 5   function mySum(a, b) { 6     return a + b 7   } 8   export { var1, const1, mySum }</pre>
<p>=== імпорт величин ===</p> <pre>import {   змінна as псевдонім_змінної ,    константа as псевдонім_константи ,    функція as псевдонім_функції ,   . . . . } from ' шлях_до_модуля '</pre>	<pre>JS main.js module_example_2 ●  module_example_2 &gt; JS main.js 1   import { 2     var1 as v1, 3     const1 as c1, 4     mySum as sum 5   } from './module1.js' 6 7   console.log(sum(v1, c1))</pre>

# Використання псевдонімів під час експорту

Загальна форма	Приклад
<p>=== експорт величини ===</p> <p>опис_змінної</p> <p>опис_константи</p> <p>опис_функції</p> <pre>export {   змінна as псевдонім_змінної ,   константа as псевдонім_константи ,   функція as псевдонім_функції ,   . . . . . }</pre>	
<p>=== імпорт величин ===</p> <pre>import {   псевдонім_змінної ,   псевдонім_константи ,   псевдонім_функції ,   . . . . . } from ' шлях_до_модуля '</pre>	

## Експорт всього модуля

### Загальна форма

=== експорт величини ===

опис\_змінної

опис\_константи

опис\_функції

*export* { змінна, константа, функція }

### Приклад

```
JS module1.js module_example_2 X
module_example_2 > JS module1.js > mySum
1   let var1 = 10
2
3   const const1 = 50
4
5   function mySum(a, b) {
6     return a + b
7   }
8   export { var1, const1, mySum }
```

=== імпорт величин ===

import \* as псевдонім\_модуля

from ' шлях\_до\_модуля '

//--- звертання до величини ---

псевдонім\_модуля . назва\_величини

```
JS main.js module_example_2 X
module_example_2 > JS main.js
1   import * as mod1 from './module1.js'
2
3   console.log(mod1.mySum(mod1.var1, mod1.const1))
4
```

# Експорт за замовчуванням (тільки одна велчина з використанням *default*)

Загальна форма	Приклад
<pre>=== експорт величини ===  export default <span>опис_величини_1</span>  export <span>опис_величини_2</span>  export <span>опис_величини_3</span> . . . . .</pre>	<pre>JS module1.js module_example_2 X module_example_2 &gt; JS module1.js &gt; mySum 1   export let var1 = 10 2 3   export const const1 = 50 4 5   export default function mySum(a, b) { 6     return a + b 7   }</pre>
<pre>=== імпорт величин ===  import <span>величина_1</span> ,  { <span>величина_2</span> ,  <span>величина_3</span> , . . . . . }  from ' шлях_до_модуля '</pre>	<pre>JS main.js module_example_2 X module_example_2 &gt; JS main.js 1   import mySum, { var1, const1 } from './module1.js' 2 3   console.log(mySum(var1, const1))</pre>

Експорт величин після опису. Експорт за замовчуванням

Загальна форма	Приклад
<p>=== експорт величини ===</p> <div>опис_величини_1</div> <div>опис_величини_2</div> <div>опис_величини_3</div> <pre>export {    величина_1 as default,    величина_2 ,    величина_3 ,   . . . . . }</pre>	<div>JS module1.js module_example_2 X</div> <div>module_example_2 &gt; JS module1.js &gt; ...</div> <pre>1  let var1 = 10 2 3  const const1 = 50 4 5  function mySum(a, b) { 6    return a + b 7  } 8  export { var1, const1, mySum as default }</pre>
<p>=== імпорт величин ===</p> <pre>import    величина_1 ,    {     величина_2 ,      величина_3 ,     . . . . .   } from ' шлях_до_модуля '</pre>	<div>JS main.js module_example_2 X</div> <div>module_example_2 &gt; JS main.js</div> <pre>1  import mySum, { var1, const1 } from './module1.js' 2 3  console.log(mySum(var1, const1)) 4</pre>



Динамічне завантаження модулів (`import модуль` – задає проміс, тому можна використати *then* або *await*)

Загальна форма	Приклад
<pre>=== експорт величини ===  export default <span>опис_величини_1</span>  export <span>опис_величини_2</span>  export <span>опис_величини_3</span>  . . . . .</pre>	<div><div><pre>JS sum_module.js module_example_2 X  module_example_2 &gt; JS sum_module.js &gt; ... 1   export let var1 = 10 2 3   export const const1 = 50 4 5   export default function mySum(a, b) { 6     return a + b 7   }</pre></div><div><pre>JS product_module.js module_example_2 X  module_example_2 &gt; JS product_module.js &gt; ... 1   export let var1 = 10 2 3   export const const1 = 50 4 5   export default function product(a, b) { 6     return a * b 7   }</pre></div></div>
<pre>=== імпорт величин ===  import(from 'модуль')   .then ( module =&gt;{     ... використання модуля ...     ... module . <span>величина</span> ...   })  module = await import from ('модуль')</pre>	<p>Питаємо, що хочемо знаходити і завантажувемо відповідний модуль</p> <div><pre>JS main.js module_example_2 ●  module_example_2 &gt; JS main.js &gt; ... 1   const ans = prompt('1) sum \n 2) product') 2   let module 3   if (ans === '1') 4     module = await import('./sum_module.js') 5   else 6     module = await import('./product_module.js') 7 8   console.log(module.default(module.var1, 5))</pre></div>