

JavaScript Moduler

13 OKTOBER 2024

JavaScript – utan tjafs
Författare: Acke Strömberg



För en kanin

Moduler

Huvudfil och type

När man skriver interaktiva webbsidor som har mer än en sida, tenderar den JavaScript (= JS) man lägger till, att svälla och bli svår att hantera. För att undvika det här problemet ska man använda sig av moduler. Det är egentligen bara ett sätt att dela in koden i olika filer så att man får lite bättre ordning på sin JavaScript. Ett sätt att dela in dem på är filer som gör specifika saker som att hämta information från ett REST API (= Representational State Transfer Application Programming Interface) eller lägga till listelement till en lista. Man kan också dela in JS så att de är direkt kopplade till en html-fil. Hur man gör indelningen beror lite grann på hur projektet ser ut. Men ett vanligt scenario som rekommenderas är att använda sig av en huvudfil, till exempel main.js och koppla till andra filer från den filen. Se bild 1 för exempel på hur kopplingen till en huvudfil bör se ut.

```
<script src="main.js" type="module" defer></script>
```

Bild 1 Huvudfilen på en webbsidas html-fil.

Lägg noga märke till att attributet type är tillagt. Det ska ha värdet "module" för att man ska kunna använda sig av moduler.

Export och import

För att kunna dela upp koden så behöver man exportera funktioner, variabler och andra värden till andra filer. Man har bestämt sig för att jobba huvudsakligen i main.js och hämtar in det man behöver till den filen. Det innebär att man behöver importera de delarna till main.js. Man kan antingen använda sig av "named exports" eller "default exports". Det finns en liten skillnad mellan dessa, men huvudsakligen klarar man sig alldeles utmärkt genom att bara använda sig av "named exports", som visas i bild 2.

```
// Filen math.js
export function add(a, b) {
  return a + b
}
```

```
// Filen main.js
import { add } from './math.js'

const result = add(5, 3)
```

Bild 2 Named export från filen math.js och en import till filen main.js.

På bild 3 kan man se hur en default export av samma funktion går till.

<pre>// Filen math.js function add(a, b) { return a + b } export default add</pre>	<pre>// Filen main.js import addTwoNumbers from './math.js' const addTwoNumbers = add(5, 3)</pre>
---	--

Bild 3 Default export från filen math.js och en import till filen main.js.

När man använder sig av "named export" så sätter man bara export framför det man vill exportera. Därefter kan man importera det till den fil man vill med hjälp av klammerparenteser, eller måsvingar som man använder i svenskan. När man importerar en "named export" måste man använda samma namn som den har i den andra filen. Man kan exportera flera värden från en och samma fil när man använder den här typen av export.

En "default export" görs genom att längst ner på den filen man vill exportera ifrån sätter export default framför. Man kan endast använda sig av en export default per fil. Till den filen man importerar så kan man importera den utan parenteser, och man har dessutom friheten att sätta ett helt valfritt namn på det man importerar. Det vill säga man måste inte använda samma namn som den hade i export-filen.