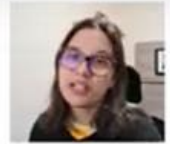




DIGITAL
INNOVATION
ONE

Assistir ma... Compartilhar



< Módulos >

Stephany Nusch
Software Engineer @ QuintoAndar



DIGITAL
INNOVATION
ONE

Objetivo do Curso



<

1. Explicar a necessidade do uso de módulos
2. Ensinar como utilizá-los

>

**Aula 1**

Utilização

Aula 2

Atividade Prática



Aula 1: Utilização

Módulos



1 | Etapa 1

1. Explicar o que são os módulos em Javascript
2. Ensinar como utilizar esses módulos
3. Alertar para algumas particularidades



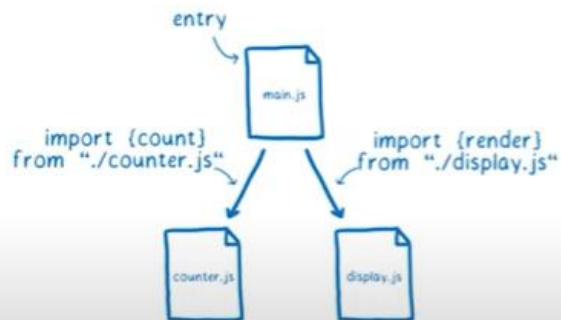
Aula 1 | Etapa 1:

O que são módulos

Módulos



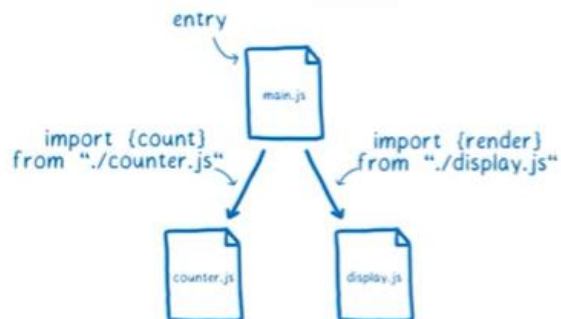
O que são módulos



Módulos arquivos Javascript que tem a capacidade de **exportar** e **importar** informações de **outros arquivos** do mesmo tipo.



O que são módulos



Algumas vantagens são:

- Organização do código;
- Compartilhamento de variáveis em escopos diferentes;
- Explicita as dependências dos arquivos.



Aula 1 | Etapa 2:

Exportar Módulos



Exportar

Named exports



```
export function mostraIdade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.idade}`;  
}  
  
export function mostraCidade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.cidade}`;  
};  
  
export function mostraHobby(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.hobby}`;  
}
```

```
function mostraIdade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.idade}`;  
}  
  
function mostraCidade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.cidade}`;  
};  
  
function mostraHobby(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.hobby}`;  
}  
  
export {  
  mostraIdade,  
  mostraCidade,  
  mostraHobby  
}
```



Exportar

Default exports



```
function mostraIdade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.idade}`;  
}  
  
function mostraCidade(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.cidade}`;  
};  
  
function mostraHobby(pessoa) {  
  return `A idade de ${pessoa.nome} é ${pessoa.hobby}`;  
}  
  
export {  
  mostraIdade,  
  mostraCidade,  
}  
  
export default mostraHobby;
```

- Só pode haver um por arquivo;
- Será o retorno padrão do seu arquivo.



Aula 1 | Etapa 3:

Importar Módulos





Importar



Named exports:

```
import {funcao, variavel, classe} from './arquivo.js'
```

Default exports

```
import valorDefault from './arquivo.js'
```



Importar



Trocando nome de imports

```
import { arquivo as Apelido } from './arquivo.js';  
Apelido.metodo();
```




Importar

Importando todos os dados de um arquivo



```
import * as INFOS from './arquivo.js';  
  
INFOS.metodoA();  
  
console.log(INFOS.variavel);
```



Aula 1 | Etapa 4:



Vinculando ao HTML

Módulos





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

Para fazer testes localmente (de um arquivo no seu computador), será necessário estar rodando um servidor. Isso pode ser feito utilizando a extensão “Live Server”, do VSCode.



Aula 1 | Etapa 5:

< Outras curiosidades >

Módulos



Module Cheatsheet	
Name Export	Name Import
<pre>export const name = 'value'</pre>	<pre>import { name } from '...'</pre>
Default Export	Default Import
<pre>export default 'value'</pre>	<pre>import anyName from '...'</pre>
Rename Export	Name Import
<pre>export { name as newName }</pre>	<pre>import { newName } from '...'</pre>
Export List + Rename	Import List + Rename
<pre>export { name1, name2 as newName2 }</pre>	<pre>import { name1 as newName1, newName2 } from '...'</pre>
<p>@ samanthaming samanthaming.com samantha_ming</p>	

- Módulos sempre estão em “strict mode”;
- Podem ser utilizadas as extensões .js e .mjs
- Para testes locais, é necessário utilizar um servidor;
- Ao importar, sempre lembre da extensão (.js, .mjs);
- Ao importar, sempre utilize “./” como ponto de partida.