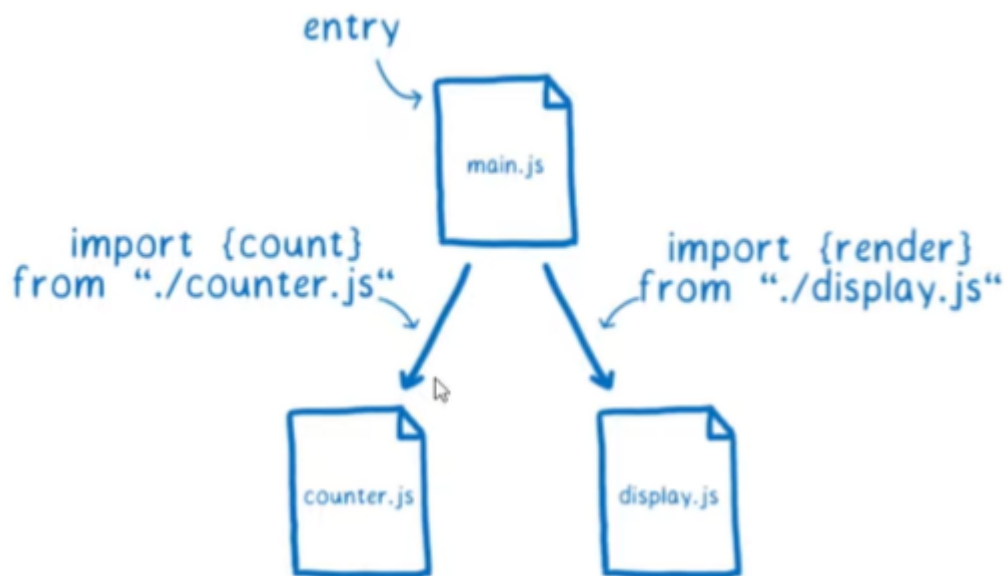


Módulos arquivos javascript que tem a capacidade de exportar e importar informações de outros arquivos do mesmo tipo.



Existe algumas vantagens ao usar os Módulos:

Algumas vantagens são:

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

A gente pode *exportar* de 2 formas:

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  
}
```

Tem uma técnica chamada “Named explore” que é da imagem esquerda e a outra é “Default explore”.

A 1º é quando eu quero exportar mais de um arquivo e eu exporto elas individualmente então eu posso fazer isso usando a palavra “export” antes da function na hora de declarar a função ou eu posso declarar a função normalmente como está na imagem da direita, porém escrevendo “export” e entre {} eu coloco todas as funções que eu quero exportar.

Esses dois tipos de export são equivalentes mas são duas maneiras de escrever e aí vai do gosto de cada pessoa.

Default Exports

Eles são um pouco diferentes, pois:

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

```
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;
```

Etapa 3: Importar

Named exports:

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

Default exports

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

Outra coisa que podemos fazer e que ajuda bastante é dar um apelido para uma função:

Trocando nome de imports

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

É bem legal porque se eu tiver 2 funções com o mesmo nome em arquivos diferentes eu posso dar um nome diferente para elas no arquivo que eu estou importando então elas não vão conflitar entre si.

Another way:

Importando todos os dados de um arquivo

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

* vai importar tudo do arquivo
INFOS é o apelido pra esse “tudo”

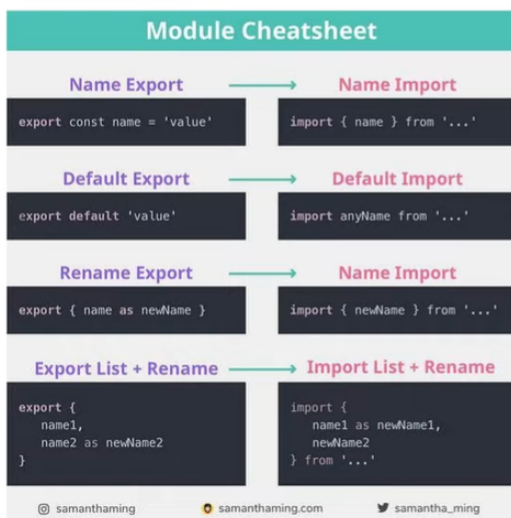
Aí você pode utilizar funções ou variáveis após o . enfim qualquer informação que eu queira utilizar do arquivo que eu importei.

Etapa 3: Vinculando ao HTML

```
<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.

Etapa 5: Outras Curiosidades



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

© samanthaming samanthaming.com samantha_ming

- 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.

Ver mais sobre “strict mode”;