

## Aula 3

# Divisão eficiente de código

### ► Unidade

Técnicas computacionais: refletindo sobre Inteligência Artificial a escola

---

### Questão 1 – Referindo uma constante

Maria criou a constante **nomes** no arquivo *nomes.js*.

Para utilizar essa mesma constante no arquivo *script.js*, ela precisa adicionar uma palavra antes de **const**. Essa palavra permitirá que outros arquivos leiam e compilem essa constante.

Qual palavra é adicionada antes da constante **nomes**?

- a) **export**
- b) **import**
- c) **let**
- d) **var**

**Alternativa A**, correta. O *export* é empregado na criação de módulos JavaScript para permitir que funções, objetos ou valores primitivos sejam exportados em tempo real. Assim, eles podem ser utilizados em outros programas por meio de declarações *import*.

**Alternativa B**, incorreta. A declaração estática *import* é usada para importar vínculos.

**Alternativa C**, incorreta. *let* permite que você declare variáveis limitando seu escopo no bloco, instrução ou em uma expressão na qual ela é usada.

**Alternativa D**, incorreta. *var* declara variáveis em um código, porém, no JavaScript não utilizamos esse tipo de sintaxe para declaração.

## Questão 2 – Importando funções

Um projeto estava muito extenso e, para deixá-lo mais organizado, foi necessário dividi-lo em arquivos diferentes. Essa divisão é chamada de módulos.

Agora, é necessário importar as funções e constantes entre arquivos js para que o projeto continue funcionando corretamente.

Ordene **todos os blocos** para importar as funções e constantes entre os arquivos js.

`from``import``'./perguntas.js';``{perguntas}`

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** `import | {perguntas} | from | './perguntas.js';`

**Comentário:** o *import* é uma declaração que indica que vamos utilizar funções, variáveis ou classes externas ao arquivo JavaScript atual, ou seja, elas foram declaradas em outro arquivo e são disponibilizadas com a importação desse conteúdo no código.

## Questão 3 – Conectando o script ao HTML

Continuando nosso projeto, precisamos conectar o *script.js* com o *index.html*.

Sabemos que o script está sendo vinculado dentro de um diretório js, porém, deve-se indicar que é dentro da pasta js e alterar o tipo do script para que tudo fique certo.

Organizando os blocos de forma correta, como é possível efetuar essa operação? Utilize **todos os blocos**.

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

Escreva a sequência correta de blocos nas linhas a seguir:

**Sequência correta:** `<script | type="module" | src="js/perguntas.js"> | </script>`

**Comentário:** o elemento HTML `<script>` é usado para incluir ou referenciar um script executável. Quando criamos projetos nos quais existe apenas um arquivo JavaScript e não atribuímos o tipo, nosso código assume que o tipo é texto. Se utilizarmos arquivos js independentes, que não se complementam, não há a necessidade de declarar o tipo ou *type*. Porém, quando dividimos nosso código JavaScript em módulos, os arquivos js se comunicam por meio do uso de *import* e *export*, e é necessário que declaremos o tipo de arquivo que estamos usando.