

Aula 6

Analizando informações

► Unidade

Projeto de vida: organizando objetivos futuros com matemática aplicada

Questão 1 – Múltiplos de três

Mariana comprou ingressos para 4 shows diferentes para ela e seus amigos. Ela quer garantir que cada amigo vá a pelo menos 1 show e que ela tenha 1 ingresso extra para levar um convidado surpresa para qualquer um dos shows. Como ela pode verificar se o número de ingressos que comprou para cada show garante que todos os amigos possam ir e ela ainda tenha um ingresso sobrando?

- a) `(numeroIngressos - 4) % 4 == 0;`
- b) `numeroIngressos / 4 == 1;`
- c) `numeroIngressos % 4 == 1;`
- d) `numeroIngressos == 4 * 4 + 1;`

Alternativa A, incorreta. Subtrair o número de ingressos por 4 não garante que vai sobrar 1 ingresso extra para levar um convidado surpresa.

Alternativa B, incorreta. Isso só seria verdadeiro se ela tivesse comprado 4 ingressos no total, o que não garante que cada amigo vá a um show e ela ainda tenha um ingresso sobrando.

Alternativa C, correta. Se o resto for 1, significa que ela sempre terá um ingresso sobrando e todos os amigos vão a pelo menos um show.

Alternativa D, incorreta. Esta verificação só seria válida para um caso, que são 17 ingressos no total. Cada amigo iria a 4 shows e ela ainda teria 1 ingresso sobrando.

Questão 2 – Adicionando classes com resto de divisão

Giuliana criou o seguinte código em HTML:

```
<p class="numero">0</p>
<p class="numero">1</p>
<p class="numero">2</p>
<p class="numero">3</p>
```

Seu desejo é adicionar classe ímpar para os números que são ímpares e classe par para os números que são pares.

Ordene todos os blocos para criar esse algoritmo.

```
for (let i=0;i<números.length;i++){
const números = document.querySelectorAll('.numero');
if (i%2==0){ números[i].classList.add("impar");
números[i].classList.add("par"); }} else {
```

Escreva a sequência correta de blocos na linha a seguir:

Sequência correta:

```
const números = document.querySelectorAll('.numero');
for (let i=0;i<números.length;i++){
  if (i%2==0){
    números[i].classList.add("par");
  }else{
    números[i].classList.add("impar");
  }
}
```

Comentário: Para sabermos se um número é par, devemos saber se o resto da divisão por 2 é zero. Caso seja, este número é par, senão, será ímpar.

Questão 3 – Adicionando classes em listas não ordenadas

O código abaixo contém uma lista que não está ordenada da maneira correta:

```
<p class="numero">1</p>
<p class="numero">0</p>
<p class="numero">3</p>
<p class="numero">2</p>
```

Sabemos que, para coletar todos os elementos com a classe `.numero`, é utilizado o comando:

```
const numeros = document.querySelectorAll('.numero');
```

Ordene todos os blocos para acessar o conteúdo interno do elemento e verificar se é par ou ímpar. Caso seja par, adicione a classe par, e, se for ímpar, adicione ímpar.

```
numero[1].classList.add("par");
```

```
let conteudoTexto=numeros[i].textContent;
```

```
números[i].classList.add("impar"); } else {
```

```
For (let i=0;i<números.length;i++){
```

```
If (conteudoTexto % 2 == 0){ }}
```

Escreva a sequência correta de blocos na linha a seguir:

Sequência correta:

```
For (let i=0;i<números.length;i++){
let conteudoTexto=numeros[i].textContent;
If (conteudoTexto % 2 == 0){
numero[1].classList.add("par");
} else {
números[i].classList.add("impar");
}}
```

Comentário: Para sabermos se um número é par, devemos saber se o resto da divisão por 2 é zero. Caso seja, este número é par, senão, será ímpar. O loop de verificação ocorrerá até a variável contadora i atingir um valor menor que o comprimento de números.