

### ### Avaliação sobre Variáveis, Condições e Loops em JavaScript Básico

#### #### Parte 1: Questões com Afirmativas (V ou F)

##### Questão 1:

Analise as afirmativas abaixo sobre variáveis em JavaScript e marque a alternativa correta.

1. A declaração ``let`` permite que uma variável seja reatribuída, mas não declarada novamente no mesmo escopo.
2. A declaração ``const`` permite a reatribuição de valores à variável.
3. A declaração ``var`` tem escopo de bloco, assim como ``let`` e ``const``.

##### Alternativas:

- a) V, F, V
- b) F, V, F
- c) V, F, F
- d) F, F, F
- e) V, V, F

##### Questão 2:

Sobre as estruturas condicionais em JavaScript, analise as afirmativas a seguir:

1. O operador ``===`` compara tanto o valor quanto o tipo dos operandos.
2. A estrutura ``switch`` pode ser usada para comparar diferentes tipos de valores, como strings e números.
3. A declaração ``if`` sempre exige um bloco ``else`` para ser válida.

##### Alternativas:

- a) V, F, F
- b) V, V, F
- c) F, V, F
- d) F, V, V
- e) V, V, V

#### #### Parte 2: Questões de Múltipla Escolha

##### Questão 3:

Qual das seguintes declarações é verdadeira sobre variáveis em JavaScript?

- a) ``var`` é uma palavra-chave que define uma variável com escopo de bloco.
- b) ``let`` permite a criação de variáveis que podem ser redeclaradas no mesmo escopo.
- c) ``const`` deve sempre ser inicializada no momento da declaração.
- d) Variáveis declaradas com ``let`` têm escopo global por padrão.
- e) Variáveis declaradas com ``var`` têm escopo de bloco.

##### Questão 4:

O que acontece se uma variável ``let`` é declarada fora de qualquer função?

- a) Ela terá escopo de bloco.

**b) Ela terá escopo global.**

- c) Ela será acessível apenas dentro da função onde foi declarada.
- d) Ela não pode ser reatribuída.
- e) Ela terá escopo local.

**Questão 5:**

**Sobre loops em JavaScript, qual das afirmações é verdadeira?**

- a) O loop `for` é usado apenas quando o número de iterações é conhecido.
- b) O loop `while` executa pelo menos uma vez, mesmo que a condição inicial seja falsa.
- c) O loop `for...of` pode ser usado para iterar sobre propriedades de um objeto.
- d) O loop `do...while` executa o bloco de código pelo menos uma vez antes de verificar a condição.**
- e) O loop `for...in` é usado exclusivamente para iterar sobre arrays.

**Questão 6:**

**Qual é o resultado do seguinte código JavaScript?**

```
````javascript
let x = 0;
for (let i = 1; i <= 5; i++) {
  x += i;
}
console.log(x);
````
```

- a) 5
- b) 10
- c) 15**
- d) 20
- e) 0

**#### Parte 3: Questões Abertas**

**Questão 7:**

**Explique a diferença entre `var`, `let` e `const` em termos de escopo e reatribuição de variáveis.**

Resposta:

- `var` tem escopo global ou de função e permite reatribuição e redeclaração dentro do mesmo escopo.
- `let` tem escopo de bloco e permite reatribuição, mas não redeclaração no mesmo escopo.
- `const` também tem escopo de bloco, mas não permite reatribuição nem redeclaração no mesmo escopo.

**Questão 8:**

**Descreva como o operador ternário funciona em JavaScript.**

Resposta:

O operador ternário é uma forma simplificada de escrever uma condicional `if-else`. Ele funciona assim: `condição ? valorSeVerdadeiro : valorSeFalso`. Se a condição for verdadeira, o primeiro valor é retornado; caso contrário, o segundo valor é retornado.

#### Questão 9:

Como o loop `for...in` funciona em JavaScript, e qual é a sua principal utilização?

Resposta:

O loop `for...in` é usado para iterar sobre todas as propriedades enumeráveis de um objeto. Para cada propriedade, o loop executa um bloco de código. Ele é particularmente útil quando se deseja acessar todas as propriedades de um objeto, incluindo as herdadas, mas não deve ser usado para iterar sobre arrays, pois pode não preservar a ordem dos elementos.

#### Questão 10:

Dê um exemplo de um loop `while` e explique como ele funciona.

Resposta:

Exemplo de loop `while`:

```
```\javascript
let i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```
```

Neste exemplo, o loop `while` continua executando o bloco de código (`console.log(i); i++;`) enquanto a condição `(i < 5)` for verdadeira. Quando `i` atinge o valor 5, a condição se torna falsa e o loop para. O loop `while` é útil quando não se sabe de antemão quantas vezes o código precisará ser repetido.