

Lógica de Programação

Aula 04

Estrutura Condicional

Prof. Anderson Passos de Aragão



Agenda

- Revisão
- Estrutura Condicional



Revisão

- Entrada de Dados: prompt
- Saída de Dados: console.log



Estrutura



Introdução

Todos os algoritmos que fizemos até agora são puramente sequenciais:

Todas as instruções são executadas na ordem em que foram definidas, uma após a outra

Veremos agora como resolver um outra classe de problemas, na qual uma ou mais ações podem ser executadas, dependendo da avaliação prévia de condições

Introdução

JavaScript oferece suporte a um conjunto compacto de instruções, especificamente instruções de controle de fluxo, que você pode usar para incorporar muita interatividade em seu aplicativo

Introdução

Lembram do problema da Média Aritmética?

E se quisermos adicionar a informação de que o aluno está aprovado ou reprovado?

- Como fazer?

Com o que aprendemos até o momento não é possível executar esta ação, uma vez que requer a análise de uma condição



Introdução

Neste caso precisamos dispor de ferramentas que permitam que uma ou mais instruções estejam associadas ao valor de uma informação

No caso do exemplo da média aritmética:

- Aprovado caso a média seja maior ou igual a 6
- Reprovado caso contrário (media menor do que 6)

Estrutura Condicional Simples

O comando só será executado se a condição for **verdadeira**

Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso

```
if(condicao){  
    comando  
}
```

Estrutura Condicional Simples

O comando só será executado se a condição for **verdadeira**

Uma condição é uma comparação que possui **dois valores possíveis**: **verdadeiro** ou **falso**

```
1  let numero = 10
2
3  if(numero > 0){
4      console.log("Condição Verdadeira")
5  }
```

Estrutura Condicional Composta

— — —

Se a condição for **verdadeira**, será executado o **comando1**; **caso contrário**, será executado o **comando2**

```
if(condicao){  
    comando1  
} else {  
    comando2  
}
```

Estrutura Condicional Composta

Se a condição for **verdadeira**, será executado o **comando1**; **caso contrário**, será executado o **comando2**

```
1  let numero = -10
2
3  if(numero > 0){
4      console.log("Condição Verdadeira")
5  } else {
6      console.log("Condição Falsa")
7  }
```

Estrutura Condicional Aninhada

Em situações em que se há um leque de opções é possível utilizar os comandos `if ... else` em sequência

Estrutura Condicional Aninhada

```
1  let opcao = 2
2
3  if(opcao == 1){
4      console.log("Opção Selecionada é a 1")
5  } else if(opcao == 2) {
6      console.log("Opção Selecionada é a 2")
7  } else if (opcao == 3){
8      console.log("Opção Selecionada é a 3")
9  } else {
10     console.log("Opção Inválida")
11 }
```

Estrutura Condicional Aninhada

```
1  let opcao = 2
2
3  if(opcao == 1){
4      console.log("Opção Selecionada é a 1")
5  } else if(opcao == 2) {
6      console.log("Opção Selecionada é a 2")
7  } else if (opcao == 3){
8      console.log("Opção Selecionada é a 3")
9  } else {
10     console.log("Opção Inválida")
11 }
```

Valores Falsos

Os seguintes valores são avaliados como **false** (também conhecidos como valores Falsy):

- false
- undefined
- null
- 0
- NaN
- a string vazia ("")

Todos os outros valores são avaliados **true** quando passados para uma instrução condicional

Valores Falsos

Condição com o número zero (0)

```
1 if(0){  
2     console.log("Condição Verdadeira")  
3 } else {  
4     console.log("Condição Falsa")  
5 }
```

Valores Falsos

Condição com variável undefined

```
13 let nome
14 if(nome){
15     console.log("Condição Verdadeira")
16 } else {
17     console.log("Condição Falsa")
18 }
```

Valores Falsos

Condição com null

```
7 if(null){  
8     console.log("Condição Verdadeira")  
9 } else {  
10     console.log("Condição Falsa")  
11 }
```

Operadores Relacionais

— — —

São utilizados em **expressões lógicas** para se testar a relação entre **dois valores do mesmo tipo**

Operador	Relação	Exemplo
==	Igualdade	qtd == 10
===	Estritamente igual	qtd === 10
<	Menor que	qtd < 10
>	Maior que	qtd > 10
<=	Menor ou Igual que	qtd <= 10
>=	Maior ou Igual que	qtd >= 10
!=	Diferente	qtd != 10

Estrutura Condicional Mutualmente Exclusiva

É um comando que permite que um programa avalie uma expressão e tente corresponder o valor da expressão a um rótulo

Se uma correspondência for encontrada, o programa executa a instrução associada

Estrutura Condicional Mutualmente Exclusiva

É um comando que permite que um programa avalie uma expressão e tente corresponder o valor da expressão a um rótulo

Se uma correspondência for encontrada, o programa executa a instrução associada

```
1 switch (expressao) {  
2     case valor1:  
3         comando1  
4         break  
5     case valor2:  
6         comando2  
7         break  
8     // ...  
9     default:  
10        comandoDefault  
11 }
```

Estrutura Condicional Mutualmente Exclusiva

```
1  let opcao = 3
2  switch (opcao) {
3    case 1:
4      console.log("Opção 1 selecionada")
5      break
6    case 2:
7      console.log("Opção 2 selecionada")
8      break
9    // ...
10   default:
11     console.log("Opção Inválida")
12 }
```

Estrutura Condicional Mutualmente Exclusiva

JavaScript avalia a instrução switch acima da seguinte forma:

- O programa primeiro procura uma cláusula case com um rótulo que corresponda ao valor da expressão e depois transfere o controle para essa cláusula, executando as instruções associadas.

```
1 switch (expressao) {  
2     case valor1:  
3         comando1  
4         break  
5     case valor2:  
6         comando2  
7         break  
8     // ...  
9     default:  
10        comandoDefault  
11 }
```


Estrutura Condicional Mutualmente Exclusiva



Instruções de interrupção

A instrução **break** opcional associada a cada cláusula **case** garante que o programa seja interrompido assim que a instrução correspondente for executada e, em seguida, continue a execução na instrução seguinte ao **switch**.

Se o **break** for omitido, o programa continua a execução dentro da instrução **switch** (e avaliará o próximo **case** e assim por diante).



```
1 let tipoFruta = "Laranja"
2 switch (tipoFruta) {
3   case "Laranja":
4     console.log("Laranjas são R$ 0.59 a unidade.")
5     break
6   case "Maças":
7     console.log("Maças são R$ 0.40 a unidade.")
8     break
9   case "Bananas":
10    console.log("Bananas são R$ 0.45 a unidade.")
11    break
12   case "Uvas":
13    console.log("Maças são R$ 2.00 o cacho.")
14    break
15   case "Mangas":
16    console.log("Maças são R$ 0.70 a unidade.")
17    break
18   default:
19    console.log(`Desculpe, estamos sem a fruta ${tipoFruta}.`);
20 }
```

Encerramento

- Revisão
- Exercício Complementar
- Próxima Aula