UNINOVE
Universidade Nove de Julho

Front End
Linguagem JavaScript

...

Objetivo: Entender como o Javascript trabalha com estruturas de decisão e estruturas de looping

UNINOVE
Universidade Nove de Julho

Estruturas de Decisão

...

UNINOVE
Universidade Nove de Julho

Estruturas de Decisão

Utiliza-se estruturas de decisão para identificar o conteúdo de uma condição e direcionar o fluxo do script para uma determinada rotina, desvio, função, etc.

Uma expressão condicional sempre retorna um valor booleano, ou seja, verdadeiro ou falso.

UNINOVE
Universidade Nove de Julho

Estruturas de Decisão

Em Javascript temos as seguintes estruturas de decisão:

- if
- if...else
- if...else if...else
- switch - case

UNINOVE
Universidade Nove de Julho

Instrução IF

A estrutura de decisão if (se) tem o objetivo de testar se uma condição é verdadeira.

A sintaxe do if é relativamente simples, como podemos ver abaixo:

```
if (condição) {
    // conjunto de instruções a ser executado,
   //  caso a condição seja verdadeira;
}
```

UNINOVE
Universidade Nove de Julho

Instrução IF

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
 <meta charset="utf-8">
 <title>Aula5</title>
 <script src="script5_01.js" defer></script>
</head>
<body>
    <h1>Aula5</h1>
    <div class="saida"></div>
</body>
</html>
```

UNINOVE
Universidade Nove de Julho

Instrução IF

```
let a,b;
const saida=document.querySelector(".saida");
a = 5;
b = "5";
if (a == b) {
  saida.innerHTML=`a = ${a}, b = ${b}<br>`;
  saida.innerHTML+=`A variável a é igual a variável b`;
}
```

UNINOVE
Universidade Nove de Julho

Instrução IF...ELSE

A instrução if...else é muito semelhante a instrução if, porém se a condição verificada for falsa, damos uma opção de saída para o código.

```
if (condição) {
    // conjunto de instruções a ser executado,
   //  caso a condição seja verdadeira;
}
else {
    // conjunto de instruções a ser executado,
   //  caso a condição seja falsa;
}
```

UNINOVE
Universidade Nove de Julho

Instrução IF...ELSE

```
let a,b;
const saida=document.querySelector(".saida");
a = 5;
b = 6;
saida.innerHTML=`a = ${a}, b = ${b}<br>`;
if (a == b) {
  saida.innerHTML+="A variável a é igual a variável b";
}
else {
  saida.innerHTML+="A variável a não é igual a variável b";
}
```

UNINOVE
Universidade Nove de Julho

Estudo de caso - PAR ou ÍMPAR

Elaborar um código Javascript para dado um número inteiro, verificar se o número é PAR ou ÍMPAR. A saída deverá mostrar o número e mostrar PAR ou ÍMPAR, como mostrado a seguir:

275 - ÍMPAR

28 - PAR

UNINOVE
Universidade Nove de Julho

Estudo de caso - PAR ou ÍMPAR

```
let num,res;
const saida=document.querySelector(".saida");
num=parseInt(prompt("Digite um número"));
res=num%2;
if(res==0){
 saida.innerHTML=`${num} - PAR`;
}
else{
    saida.innerHTML=`${num} - ÍMPAR`;
}
```

UNINOVE
Universidade Nove de Julho

Estudo de caso - PAR ou ÍMPAR

```
let num,res;
const saida=document.querySelector(".saida");
num=parseInt(prompt("Digite um número"));
if(num%2==0){
    saida.innerHTML=`${num} - PAR`;
}
else{
    saida.innerHTML=`${num} - ÍMPAR`;
}
```

UNINOVE
Universidade Nove de Julho

Estudo de caso - PAR ou ÍMPAR

```
let num,res;
const saida=document.querySelector(".saida");
num=parseInt(prompt("Digite um número"));
msg=num%2==0?"PAR":"ÍMPAR";
saida.innerHTML=`${num} - ${msg}`;
```

UNINOVE
Universidade Nove de Julho

Instrução IF...ELSE IF...ELSE

Assim como a instrução if a instrução else if também contém uma declaração condicional, todavia, esta instrução deve obrigatoriamente ser precedida por uma instrução if. Você não pode ter uma instrução elseif sem ter primeiro uma instrução if.

Quando o Javascript avalia uma instrução if...else if, a verificação acontece no primeiro IF, caso a condição se falsa aí então será checada a segunda condição, a condicional do elseif.

UNINOVE
Universidade Nove de Julho

Instrução IF...ELSEIF...ELSE

```
if (condição 1) {
// set de instruções a ser executado, caso a condição seja verdadeira;
}
else if (condição 2){
// set de instruções a ser executado, caso a condição seja verdadeira;
}
else{
// set de instruções a ser executado, caso a condição seja falsa;
}
```

UNINOVE
Universidade Nove de Julho

Instrução IF...ELSEIF...ELSE

```
let a,b;
const saida=document.querySelector(".saida");
a=6;
b=5;
if (a == b) {
 saida.innerHTML="A variável a é igual a variável b";
}
else if (a < b){
 saida.innerHTML="A variável a é menor a variável b";
}
else{
 saida.innerHTML="A variável a é maior a variável b";
}
```

UNINOVE
Universidade Nove de Julho

Instrução switch/case

A instrução switch/case permite verificar todas condições de uma só vez. Portanto, é própria para se testar uma variável em relação a diversos valores pré-estabelecidos.

Todos os tipos primitivos (int, string, decimal, etc) podem ser usados nas instruções switch/case.

A instrução case avalia apenas o argumento que apresenta o mesmo tipo (string para string, int para int, etc) definido na instrução switch

UNINOVE
Universidade Nove de Julho

Instrução switch/case

```
switch (variável) {
   case opção 1:
   // set de instruções a ser executado, caso a condição seja verdadeira;
   break;
   case opção 2:
   // set de instruções a ser executado, caso a condição seja verdadeira;
   break;
   case opção N:
   // set de instruções a ser executado, caso a condição seja verdadeira;
   break;
   default:
   // set de instruções comandos caso nenhuma das opções anteriores tenha sido
escolhida;
}
```

UNINOVE
Universidade Nove de Julho

Instrução switch/case

```
let destino,msg;
const saida=document.querySelector(".saida");
destino=prompt("Digite seu destino: ");
switch (destino) {
  case "São Paulo":
    msg="Seja bem vindo a cidade que nunca para!";
  break;
  case "New York":
    msg="Seja bem vindo a Big Apple!";
  break;
  default:
    msg="Não encontramos seu destino!";
}
saida.innerHTML=msg;
```

UNINOVE
Universidade Nove de Julho

Estruturas de Repetição (Looping)

...

UNINOVE
Universidade Nove de Julho

Estruturas de Repetição (Looping)

As estruturas de repetição são úteis para executar tarefas repetitivas, por exemplo, repetir elementos de uma lista ou de uma tabela de dados ou simplesmente para executar um mesmo processamento até que uma certa condição seja satisfeita ou até atingir um determinado número de vezes.

Cada volta de uma estrutura de repetição damos o nome de iteração.

UNINOVE
Universidade Nove de Julho

Estruturas de Repetição (Looping)

Em Javascript, temos as seguintes estruturas de repetição:

- while (enquanto)
- do while (faça enquanto)
- for (para)

UNINOVE
Universidade Nove de Julho

Instrução WHILE

A instrução while irá repetir a execução de um determinado bloco de instrução enquanto a condição testada seja verdadeira, quando essa condição se tornar falsa, essa repetição será suspensa.

A presença de um contador se torna obrigatória em scripts que utilizem a estrutura de repetição while, caso contrário, esta estrutura de repetição cairá num loop infinito.

UNINOVE
Universidade Nove de Julho

Instrução WHILE

A sintaxe básica da instrução while é:

```
variavel = valor_inicial; //inicialização do nosso contador;
while (condição) {
  // set de instruções a ser executado repetidamente;
  variavel++;// incremento do contador
}
```

UNINOVE
Universidade Nove de Julho

Instrução WHILE

```
let i;
const saida=document.querySelector(".saida");
i=0;
while (i < 5) {
    saida.innerHTML+=i + "<br>";
    i++;
}
```

UNINOVE
Universidade Nove de Julho

Instrução DO...WHILE

As instruções do tipo do...while são semelhantes aos loops while com a diferença de que os loops do...while primeiro executa o bloco de instruções e depois verifica se a expressão é verdadeira.

Portanto, neste tipo de estrutura o bloco de repetição será executado ao menos uma vez, para somente depois a condição ser verificada, caso esta continue verdadeira o bloco de instruções será repetido novamente, caso contrário, o bloco de instruções do...while será encerrado.

UNINOVE
Universidade Nove de Julho

Instrução DO...WHILE

A sintaxe básica da instrução do...while é:

```
$variavel = valor_inicial; //inicialização do nosso contador;
do {
  // set de instruções a ser executado repetidamente;
  $variavel++;// incremento do contador
}while (condição)
```

UNINOVE
Universidade Nove de Julho

Instrução DO...WHILE

```
let i;
const saida=document.querySelector(".saida");
i=0;
do{
    saida.innerHTML+=i + "<br>";
    i++;
}while (i < 3);
```

UNINOVE
Universidade Nove de Julho

Instrução FOR

Utilizamos a instrução for quando já sabemos, previamente, a quantidade de vezes que iremos executar a estrutura de repetição.

Sintaxe da instrução for:

```
for (expressão1;expressão2;expressão3) {
  // set de instruções a ser executado repetidamente;
}
```

Instrução FOR

```
for (expressão1;expressão2;expressão3) {
  // set de instruções a ser executado repetidamente;
}
```

A instrução for é dividida em três expressões, a saber:

expressão1: é a variável de controle inicial;

expressão2: é a condição a ser testada, sempre que a condição for verdadeira a repetição é executada;

expressão3: é o incremento/decremento

UNINOVE
Universidade Nove de Julho

Instrução FOR

```
let i;
const saida=document.querySelector(".saida");
for (i=0; i<6; i++) {
   saida.innerHTML+=i + " ";
}
```

UNINOVE
Universidade Nove de Julho

Instrução FOR (aninhada)

Podemos também ter a instrução for de forma aninhada, ou uma instrução for dentro da outra.

```
for (expressao1;expressao2;expressao3) {
// set de instruções a ser executado repetidamente;
   for (expressao1;expressao2;expressao3) {
// set de instruções a ser executado repetidamente;
   } // finaliza a segunda instrução for
} // finaliza a primeira instrução for
```

UNINOVE
Universidade Nove de Julho

Instrução FOR (aninhada)

```
let i,j;
const saida=document.querySelector(".saida");
for (i=0; i<3; i++) {
 for (j=0; j<4; j++) {
   saida.innerHTML+=`${i} ${j}<br>`;
 }
}
```

UNINOVE
Universidade Nove de Julho

Referências

Referências

MORRISON, M. Use a cabeça JavaScript. 5o Ed. Rio de Janeiro: Alta Books, 2012. 606 p.

OLIVIERO. C. A. J. Faça um site JavaScript orientado por projeto. 6o ed. São Paulo: Érica, 2010. 266 p.

ZAKAS, Nicholas C. JavaScript de alto desempenho. 8o Ed. São Paulo: Novatec, 2010. 245 p.