

Lógica computacional

JavaScript

Desafio cálculo IMC

$$\text{IMC} = \frac{\text{PESO}}{(\text{ALTURA})^2}$$

CLASSIFICAÇÃO	IMC
Abaixo do Peso	Abaixo 18,5
Peso Normal	18,5 - 24,9
Sobrepeso	25 - 29,9
Obesidade Grau I	30 - 34,9
Obesidade Grau II	35 - 39,9
Obesidade Grau III ou Mórbida	Maior ou Igual 40

Laços e iterações

Servem para executar um trecho de código em um número determinado de vezes até que uma determinada condição seja satisfeita. As estruturas de repetição também são chamadas de laços ou loops.

Sintaxe:

```
var passo;
```

```
for (passo = 0; passo < 5; passo++) {
```

```
    // Executa 5 vezes, com os valores de passos de 0 a 4.
```

```
    console.log('Ande um passo para o leste');
```

```
}
```

Existem várias formas diferentes de laços, mas eles essencialmente fazem a mesma coisa: repetir uma ação múltiplas vezes (inclusive você poderá repetir 0 vezes). Os vários mecanismos diferentes de laços oferecem diferentes formas de determinar quando este irá começar ou terminar. Há várias situações em que é mais fácil resolver um problema utilizando um determinado tipo de laço do que outros.

Declaração for

Um laço for é repetido até que a condição especificada seja falsa. O laço for no JavaScript é similar ao Java e C. Uma declaração for é feita da seguinte maneira:

```
for ([expressaoInicial]; [condicao]; [incremento])
```

```
  declaracao
```

Quando um for é executado, ocorre o seguinte:

1. A expressão expressão Inicial é inicializada e, caso possível, é executada. Normalmente essa expressão inicializa um ou mais contadores, mas a sintaxe permite expressões de qualquer grau de complexidade. Podendo conter também declaração de variáveis.
2. A expressão condição é avaliada, caso o resultado da condição seja verdadeiro, o laço é executado. Se o valor da condição é falso, então o laço terminará.
3. A instrução é executada. Para executar múltiplas declarações, use uma declaração em bloco ({ ... }) para agrupá-las.
4. A atualização da expressão do incremento, se houver, executa, e retorna o controle para o passo 2.

Exemplo

```
for (let i = 0; i < 5; i++) {  
  console.log("Índice atual: "i);  
}
```


Declaração do...while

A instrução será executada uma vez antes da condição ser verificada. Para executar múltiplas instruções utilize uma declaração de bloco (`{ ... }`) para agrupá-las. Caso a condição seja verdadeira, então o laço será executado novamente. Ao final de cada execução, a condição é verificada. Quando a condição contida no while for falsa a execução do laço é terminada e o controle é passado para a instrução seguinte a do...while.

Exemplo

```
let i = 0;
```

```
do {
```

```
  i += 1;
```

```
  console.log(i);
```

```
} while (i < 5);
```

Declaração while

Uma declaração while executa suas instruções, desde que uma condição especificada seja avaliada como verdadeira.

Se a condição se tornar falsa, a declaração dentro do laço para a execução e o controle é passado para a instrução após o laço.

O teste da condição ocorre antes que o laço seja executado. Desta forma se a condição for verdadeira o laço executará e testará a condição novamente. Se a condição for falsa o laço termina e passa o controle para as instruções após o laço.

Para executar múltiplas declarações, use uma declaração em bloco (`{ ... }`) para agrupar essas declarações.

Exemplo 1

```
n = 0;
```

```
x = 0;
```

```
while (n < 3) {
```

```
    n++;
```

```
    x += n;
```

```
}
```

Exemplo 2

Evite laços infinitos. Tenha certeza que a condição do laço eventualmente será falsa; caso contrário, o laço nunca terminará. O while a seguir executará para sempre pois sua condição nunca será falsa:

```
while (true) {  
    console.log("Olá, mundo");  
}
```

Fontes

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Loops_and_iteration