INFINITY SCHOOL VISUAL ART CREATIVE CENTER

AULA 04 - LAÇO DE REPETIÇÃO II

O QUE IREMOS APRENDER

01

RESUMO DA AULA PASSADA

02

FOR

03

ESTRUTURA DE REPETIÇÃO FOR

04

FOR...OF

05

MÃOS NO CÓDIGO

RESUMO DA AULA PASSADA

Um Laço de Repetição, ou loop, é uma estrutura em programação que repete uma sequência de instruções até que uma condição específica seja atendida.

Na aula passada, aprendemos um pouco sobre a estrutura de repetição while. O while (enquanto) consiste em executar determinada instrução enquanto determinada condição for verdadeira, a partir do momento em que o teste retornar false, o laço de repetição não será mais executado.



Assim como o while, o for também nos permite repetir um trecho de código enquanto uma condição for atendida. Mas utilizando o while, sempre que precisamos de um contador em nosso laço de repetição para executar um número específico de vezes, precisamos inicializar uma variável antes de começar o nosso while. O for vem para resolver esse problema, nos proporcionando um controle muito maior sobre a quantidade de vezes que aquele código irá rodar.



Usamos o for em ocasiões em que sabemos a quantidade de vezes que aquele código irá rodar, seja por um valor já definido dentro do código ou por um valor inserido pelo usuário.

O for auxilia muito na leitura do nosso código, facilitando a compreensão de como aquele laço vai se comportar.



Para utilizar essa estrutura, precisamos primeiro passar a palavra reservada "for" e, em seguida, entre parênteses e separadas por ponto e vírgula, as expressões para o laço funcionar, sendo o primeiro a declaração de uma variável, depois a condição que deve ser atendida para o laço continuar rodando e, por último, o que vai acontecer após cada repetição.

```
// (declaração da minha variável; condição para o looping rodar; o que acontecer depois de cada ciclo)
  for (let i = 0; i < 10; i++) {
      //Código a ser executado
}</pre>
```

A condição no loop for é uma expressão que determina se o loop deve continuar executando ou ser interrompido. A expressão é avaliada antes de cada iteração. Se for verdadeira, o bloco de código dentro do loop é executado; caso contrário, o loop é encerrado.

```
for (let i = 0; i < 5; i++) {
    //Código a ser executado
}</pre>
```

Na condição i < 5, o loop continuará enquanto o valor de i for menor que 5. A condição pode ser qualquer expressão que resulte em um valor booleano (true ou false).



O componente de incremento/decremento é responsável por alterar o valor da variável de controle em cada iteração do loop.

No exemplo i++, i é incrementado por 1 a cada iteração. O mesmo pode ser feito com i-- para decrementar.

Outras formas de incremento/decremento incluem i += 2 (incrementa por 2) ou i -= 3 (decrementa por 3).

O incremento/decremento ocorre após a execução do bloco de código no loop.



No exemplo, i++ é usado, o que significa que i será incrementado por 1 após cada iteração.

É importante ajustar o incremento/decremento para evitar loops infinitos ou para atender aos requisitos específicos do problema.

```
for (let i = 0; i < 5; i++) {
    //Código a ser executado
}</pre>
```

```
// No exemplo anterior, declaramos i como nossa variável.
// Após isso, dizemos que o laço continuará executando enquanto i for
menor que 10
// E, por último, dizemos que i vai somar 1 ao valor atual.

// Seguindo esse mesmo exemplo:

for (let i = 0; i < 3; i++) {
    console.log(i) // Imprime o valor de i no console
    console.log('Passei por um ciclo') // Imprime a mensagem no consol
e
}</pre>
```

Passei por um ciclo

Passei por um ciclo

Passei por um ciclo

um ciclo



Tanto o break quanto o continue também funcionam no for:

```
for(let i = 0; i < 3; i ++) {
    if(i == 1) {
        continue
    }
        Teste esse código
    console.log(i)
}</pre>
```



ATIVIDADE PRÁTICA

Atividade 01

Escreva um programa que imprima no console os números de 0 a 10.

Atividade 02

Escreva um programa que solicite um número inteiro e imprima a tabuada desse número de 1 a 10.

ATIVIDADE PRÁTICA

Atividade 03

Escreva um programa que solicite um número inteiro positivo e faça uma contagem regressiva a partir desse número até 0.

Atividade 04

Escreva um programa que solicite um número e verifique se ele é um número primo.

For...of

O for of é uma estrutura que nos traz a possibilidade de percorrer algum tipo de coleção de dados, como uma string, array, set ou map (abordaremos mais sobre os últimos três em aulas futuras). Nesse laço, a cada ciclo, o programa vai pegar um dos valores da coleção e guardará em uma variável, permitindo-nos acessar os dados de cada ciclo. Vamos para um exemplo:

```
texto = 'Meu texto'

for (letra of texto) {
    console.log(letra)
    console.log('Passei por um ciclo')
}
```



For...of

Nesse exemplo, o "for" está percorrendo a string "texto", atribuindo a cada ciclo um caractere da string à variável "letra", até que toda a string tenha sido percorrida.

```
texto = 'Meu texto'

for (letra of texto) {
    console.log(letra)
    console.log('Passei por um ciclo')
}
```

```
Passei por um ciclo
```



ATIVIDADE PRÁTICA

Atividade 06

Escreva um programa que irá solicitar ao usuário uma palavra e que imprimirá no console a palavra sem vogais.

Atividade 07

Escreva uma programa que irá receber um texto e mostrará no console se esse texto é um palíndromo ou não. (palíndromos são palavras que são lidas da mesma forma de trás pra frente) Ex: arara

ATIVIDADE PRÁTICA

Atividade 07

Escreva um programa que irá solicitar ao usuário uma palavra e conte o número de vogais presentes.

Atividade 08

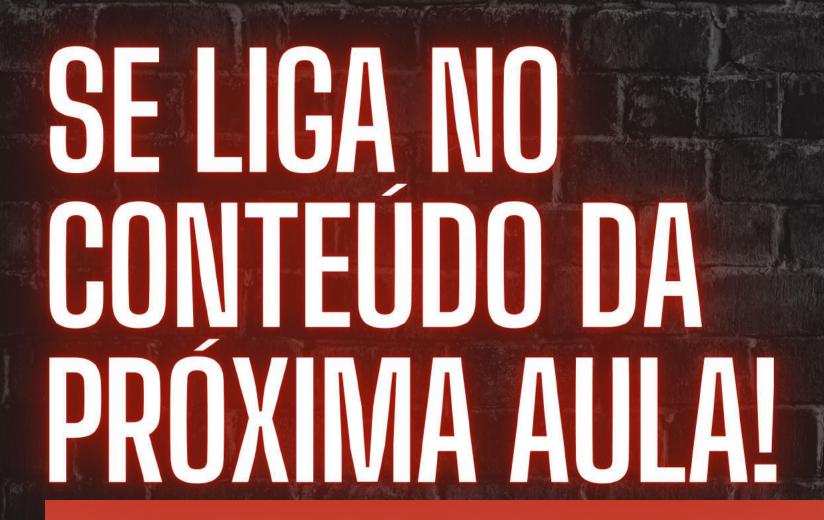
Escreva um programa que imprima os múltiplos de 3 de 1 a 30.

DESAFIO PRÁTICO

Neste desafio, você criará um jogo simples de par ou ímpar usando loops (while), condicionais (if, else), e operadores lógicos (||).

Regras:

- O jogo continuará até que o jogador decida parar. Em cada rodada, um dado será lançado.
 - O jogador escolherá se quer apostar em "par" ou "ímpar".
 - O resultado do dado será verificado e o vencedor será determinado com base na escolha do jogador.



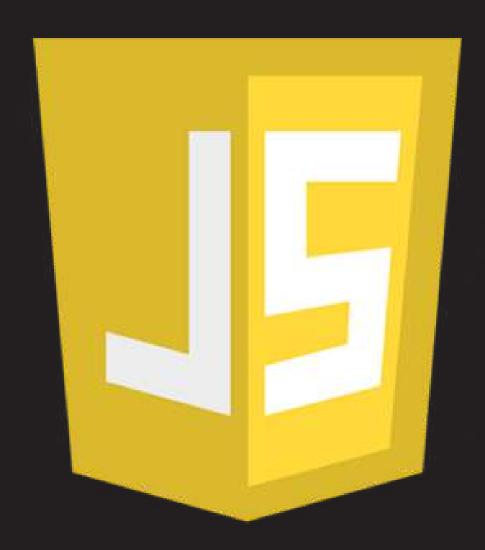
AULA 05 DE JAVASCRIPT. ARRAYS

INFINITY SCHOOL
VISUAL ART CREATIVE CENTER

ARRAY

No JavaScript, o primeiro tipo de variável composta que vamos estudar é chamado de Array.

Para criar um array no JavaScript, utilizamos pares de colchetes []. E então, cada valor dentro dos [] é separado por vírgula.



ARRAY

Exemplos de **Arrays** em JavaScript:

```
let nome = "Pedro" // Variável Simples
let nome2 = "Marcela" // Variável Simples
let nomes = ["Marcela", "Pedro"] // Variável Composta
/*
Perceba que tanto a string Marcela quanto a string
Pedro estão armazenas na variável "nomes"
```

