

INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO

JULIANO SILVA

DEFINIÇÃO E CONCEITOS

- A lógica de programação a técnica de encadear pensamentos para atingir determinado objetivo.
- Lógica de programação é a maneira como organizamos pensamentos para conseguir executar uma tarefa.
- Por exemplo, como fazemos para verificar se uma pessoa é maior de idade ou não?

ALGORITMOS

- Um algoritmo é formalmente uma sequência finita de passos que levam à execução de uma tarefa.
- Um exemplo é o manual de um equipamento, como o de um gravador de DVD, que explica passo a passo como realizar uma gravação.
- Como você escreveria um algoritmo para fazer uma ligação?

VARIÁVEIS E CONSTANTES

- Variáveis são espaços reservados na memória RAM do computador para guardar informações que serão utilizadas durante o código do programa. Elas podem ter valores de diversos tamanhos e tipos.
- Constante é um valor fixo que não se modifica ao longo da execução do programa.

```
2  var nome = 'Juliano'
3  let sobrenome = 'Silva'
4
5  const PI = 3.141592
```

TIPOS DE DADOS PRIMITIVOS

- Boolean - possuem apenas dois valores: verdadeiro ou falso;
- Undefined - indica que não foi definido um valor;
- Null - indica que um valor é nulo;
- Number - armazena valores numéricos;
- String - armazena textos;

```
2  let boleano = true
3  let indefinido = undefined
4  let nulo = null
5  let numero = 27
6  let texto = 'Juliano'
7
```

VALORES VERDADEIROS E FALSOS

- 0;
- -0;
- null;
- false;
- NaN;
- undefined;
- "" (string vazia).

```
2  0;  
3  -0;  
4  null;  
5  false;  
6  NaN;  
7  undefined;  
8  ""; // (string vazia).
```


OPERADORES ARITMÉTICOS E ATRIBUIÇÃO

- + soma e concatenação
- - subtração
- / divisão
- * multiplicação
- % módulo
- ** potência
- ++ incremento
- -- decremento

- = atribuição
- += soma e concatenação
- -= subtração
- /= divisão
- *= multiplicação
- %= módulo
- **= potência

OPERADORES LÓGICOS E RELACIONAIS

- && - e
- || - ou
- == - igual
- === - idêntico
- != - diferente
- !== não idêntico
- < - menor
- <= - menor ou igual
- > - maior
- >= - maior ou igual

OPERADORES UNÁRIOS E TERNÁRIOS

- ! – diferente ou negação
- typeof – pega o tipo de dados
- - – inverter operação
- (se) ? (então) : (senão) - operador ternário

```
if (!true) {  
  console.log('Juliano');  
}  
  
let numero = 42;  
console.log(typeof numero);  
console.log(-numero)  
console.log(  
  numero > 42 ? 'A vida o universo e tudo mais' : 'nada importante'  
)
```

ESTRUTURAS DE DECISÃO

- if
- else
- elseif
- switch
- case
break

```
if (true) {  
    console.log('Juliano');  
} else if(false) {  
    console.log('Silva');  
} else {  
    console.log('(-_-)');  
}  
  
let idade = 16;  
switch (idade) {  
    case idade > 18:  
        console.log("Maior de Idade");  
        break;  
    case idade < 18:  
        console.log("Menor de Idade");  
        break;  
    default:  
        console.log("Não sei dizer")  
}
```

ESTRUTURAS DE REPETIÇÃO

```
for (let i = 0; i <= 10; i++) {  
  console.log(i);  
}
```

```
let numero = 0;  
while (numero < 10) {  
  console.log(numero);  
  numero++;  
}
```

FUNÇÕES

```
function somar(numero1, numero2)
{
    return numero1 + numero2;
}

let resultado = somar(2, 3);
console.log(resultado);

let multiplicar = (numero1, numero2) {
    return numero1 * numero2;
}

resultado = multiplicar(5, 3);
console.log(resultado);
```

INTERAÇÃO COM O NAVEGADOR

The background is a light gray with a subtle pattern of large, faint question marks. Overlaid on this are several white circles of varying sizes and thin, dark gray lines that resemble circuit board traces or network connections, particularly concentrated in the corners.