

REVISÃO DE ESTRUTURA DE DADOS E LÓGICA DE PROGRAMAÇÃO

Prof. Ms. José Antonio Gallo Junior

ALGORITMOS E PROGRAMAS

Sempre pense e analise o problema a ser resolvido antes de começar a tentar resolvê-lo.

Divida o problema a ser solucionado em três fases fundamentais:

- Entrada
- Processamento
- Saída

criação de variáveis e constantes

Podemos criar variáveis com os comandos `let` e `var`. Para criar constantes utilizamos o `const`.

Exemplos:

```
let nome = "José Antonio Gallo Junior"
```

```
let idade = 42
```

```
var altura = 1.68
```

```
const PI = 3.141592653589;
```

TIPOS DE DADOS

Tipo	Descrição	Exemplos
String	Texto	<code>let nome = "Gallo"; let nome = 'Gallo'</code>
Number	Números inteiros e reais	<code>let idade = 42; let altura = 1.68;</code>
Bigint	Números inteiros maiores que 64bits	<code>let x = BigInt('123456789012345678901234567890');</code>
Boolean	Lógicos: true ou false	<code>let x = true; let y = false;</code>
Undefined	Indefinido	<code>let x; let y = undefined;</code>
Null	Nulo	<code>let x = null;</code>
Object	Pode ser um objeto, vetor ou data	<code>let pessoa = {nome: "Gallo", idade: 42}; let semana = ["Dom", "Seg", "Ter", "Qua", "Qui", "Sex", "Sab"]; let dataNascimento = new Date("1985-08-05")</code>

OPERADORES ARITMÉTICOS

Operador	Descrição	Exemplo de Uso	Resultado
+	Adição	<code>let x = 10 + 5; console.log(x);</code>	15
-	Subtração	<code>let x = 10 - 5; console.log(x);</code>	5
*	Multiplicação	<code>let x = 10 * 5; console.log(x);</code>	50
**	Exponenciação	<code>let x = 10 ** 5; console.log(x);</code>	100000
/	Divisão	<code>let x = 10 / 5; console.log(x);</code>	2
%	Resto	<code>let x = 10 % 5; console.log(x);</code>	0
++	Incremento	<code>let x = 5; x++; console.log(x);</code>	6
--	Decremento	<code>let x = 5; x--; console.log(x);</code>	4

OPERADORES ARITMÉTICOS - ORDEM DE EXECUÇÃO

- () Parênteses
- ** Exponenciação
- *
- /
- %
- +
- -

OPERADORES DE ATRIBUIÇÃO

Operador	Descrição	Exemplo de Uso	Resultado
=	Atribuição	<code>x = 5; console.log(x)</code>	5
+=	Atribuição de Adição	<code>x += 5; console.log(x);</code>	15
-=	Atribuição de Subtração	<code>x -= 5; console.log(x);</code>	5
*=	Atribuição de Multiplicação	<code>x *= 5; console.log(x);</code>	50
**=	Atribuição de Exponenciação	<code>x **= 5; console.log(x);</code>	100000
/=	Atribuição de Divisão	<code>x /= 5; console.log(x);</code>	2
%=	Atribuição de Resto	<code>x %= 5; console.log(x);</code>	0
Considerando para os exemplos o seguinte: <code>let x = 10;</code>			

OPERADORES DE COMPARAÇÃO

Operador	Descrição	Exemplo: let x = 5	Resultado	Exemplo: let x = 5	Resultado
==	Igualdade	x == 5	true	x == "5"	true
===	Igualdade de valor e tipo	x === 5	true	x === 5	false
!=	Diferença	x != 5	false	x != "5"	false
!==	Diferença de valor e tipo	x !== 5	false	x !== "5"	true
>	Maior	x > 4	true	x > 8	false
<	Menor	x < 10	true	x < 3	false
>=	Maior ou igual	x >= 5	true	x >= 3	false
<=	Menor ou igual	x <= 10	true	x <= 4	false

OPERADORES LÓGICOS

Operador	Descrição	Exemplo: let x = 5; let y = 10	Resultado	Exemplo: let x = 5; let y = 10	Resultado
&&	e	(x < 10 && y > 1)	true	(x <= 5 && y > 12)	false
	ou	(x < 10 y < 5)	true	(x > 5 y != 10)	false
!	não	!(x == y)	true	!(x != y)	false

Tabela Verdade - &&		
Valor 1	Valor 2	Resultado
true	true	true
true	false	false
false	true	false
false	false	false

Tabela Verdade -		
Valor 1	Valor 2	Resultado
true	true	true
true	false	true
false	true	true
false	false	false

Tabela Verdade - !	
Valor	Resultado
true	false
false	true

A stylized illustration of a road with a signpost and two large, textured, dome-like structures on either side. The road is dark with white dashed lines, leading towards a signpost in the distance. The signpost has two signs: a rectangular one pointing left and a triangular one pointing right. The two large, dome-like structures are light gray with dark, textured lines, resembling a stylized landscape or perhaps large pieces of machinery. The background is a light gray gradient.

ESTRUTURAS DE DESVIO CONDICIONAL

ESTRUTURAS DE DESVIO CONDICIONAL

A Estrutura Condicional possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas. A Estrutura Condicional pode ser Simples ou Composta.

- if
- else
- else if
- switch

CONDICIONAL IF

```
if (condição) {  
    // Código executado se a condição for verdadeira  
}
```

Exemplo:

```
if (hora < 12) {  
    console.log("Bom Dia");  
}
```

CONDICIONAL IF ELSE

```
if (condição) {  
    // Código executado se a condição for verdadeira  
} else (condição) {  
    // Código executado se a condição for falsa  
}
```

Exemplo:

```
if (hora < 12) {  
    console.log("Bom Dia");  
} else {  
    console.log("Boa tarde");  
}
```

CONDICIONAL IF ELSE IF

```
if (condição1) {  
    // Código executado se a condição 1 for verdadeira  
} else if (condição2) {  
    // Código executado se a condição 1 for falsa e a  
    // condição 2 for verdadeira  
} else {  
    // Código executado se a condição1 e a condição 2  
    //forem falsas  
}
```

CONDICIONAL IF ELSE IF

```
if (hora < 12) {  
    console.log("Bom Dia");  
} else if (hora < 18) {  
    console.log("Bom Tarde");  
} else {  
    console.log("Boa Noite");  
}
```

CONDICIONAL IF ELSE IF

```
if (hora < 6) {  
    console.log("Boa Noite");  
} else if (hora < 12) {  
    console.log("Bom Dia");  
} else if (hora < 18) {  
    console.log("Boa tarde");  
} else {  
    console.log("Boa noite");  
}
```


CONDICIONAL SWITCH

```
switch(variável ou expressão) {  
    case x:  
        // bloco de código  
        break;  
    case y:  
        // bloco de código  
        break;  
    default:  
        // bloco de código  
}
```

CONDICIONAL SWITCH

```
switch (new Date().getDay()) {  
  case 6:  
    console.log("Hoje é Sábado");  
    break;  
  case 0:  
    console.log("Hoje é Domingo");  
    break;  
  default:  
    console.log("Esperando pelo fim de semana");  
}
```

An aerial photograph of a large, multi-lane roundabout. The roundabout has a central green island and several smaller green islands at the entry points. The road is dark asphalt with white dashed lines. Several cars are visible on the road. The surrounding area is green grass.

ESTRUTURAS DE REPETIÇÃO

ESTRUTURAS DE REPETIÇÃO

Permitem que uma ou mais instruções sejam executadas por um número determinado de vezes, ou enquanto uma determinada condição não seja satisfeita.

- for
- while
- do ...while
- BREAK
- CONTINUE

REPETIÇÃO FOR

```
for (expressão 1; expressão 2; expressão 3) {  
    // Bloco de código a ser executado  
}
```

Expressão1: definição e inicialização da variável a ser utilizada no laço de repetição (é executado uma vez)

Expressão2: definição da condição de execução do bloco de código

Expressão3: é executado toda vez, ao final do bloco de código, determinar o passo da execução

REPETIÇÃO FOR

```
let n = 5;  
for (let i = 0; i <= 10; i++) {  
    console.log(`${n} x ${i} = ${n*i}`);  
}
```

Resultado

5 x 0 = 0

5 x 1 = 5

.....

5 x 10 = 50

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

REPETIÇÃO WHILE

```
while (condição) {  
    // Bloco de código a ser executado  
}
```

Condição: expressão lógica que mantém o laço em repetição enquanto for verdadeira

REPETIÇÃO WHILE

```
let n = 5;
let i = 0;
while (i <= 10) {
    console.log(`${n} x ${i} = ${n*i}`);
    i++;
}
```

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

REPETIÇÃO DO WHILE

```
do {
```

```
    // Bloco de código a ser executado
```

```
} while (condição)
```

Condição: expressão lógica que mantém o laço em repetição enquanto for verdadeira

REPETIÇÃO DO WHILE

```
let n = 5;  
let i = 0;  
do {  
    console.log(`${n} x ${i} = ${n*i}`);  
    i++;  
} while (i <= 10)
```

5 x 0 = 0

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

REPETIÇÃO - BREAK

Em algumas situações específicas, pode ocorrer a necessidade de se "quebrar" um laço de repetição, forçando assim a sua parada, para essas situações existe o comando `break`.

`break;`

Exemplo:

```
for (let i = 0; i <= 10; i++) {  
    if (i === 3) { break; }  
    console.log(`Número ${i}`);  
}
```

```
> for (let i = 0; i <= 10; i++) {  
    if (i === 3) { break; }  
    console.log(`Número ${i}`);  
}  
Número 0 VM5588:3  
Número 1 VM5588:3  
Número 2 VM5588:3  
< undefined  
>
```

REPETIÇÃO - CONTINUE

Em algumas situações específicas, pode ocorrer a necessidade de se "pular" um loop de um laço de repetição, para essas situações existe o comando `continue`.

`continue;`

Exemplo:

```
for (let i = 0; i <= 10; i++) {  
    if (i === 3) { continue; }  
    console.log(`Número ${i}`);  
}
```

```
> for (let i = 0; i <= 10; i++) {  
    if (i === 3) { continue; }  
    console.log(`Número ${i}`);  
}  
  
Número 0 VM5594:3  
Número 1 VM5594:3  
Número 2 VM5594:3  
Número 4 VM5594:3  
Número 5 VM5594:3  
Número 6 VM5594:3  
Número 7 VM5594:3  
Número 8 VM5594:3  
Número 9 VM5594:3  
Número 10 VM5594:3  
  
< undefined  
> |
```

MUITO OBRIGADO!!!

