

# Aulas Senac

Desenvolvimento Web para Adolescentes

Pablo Garcia

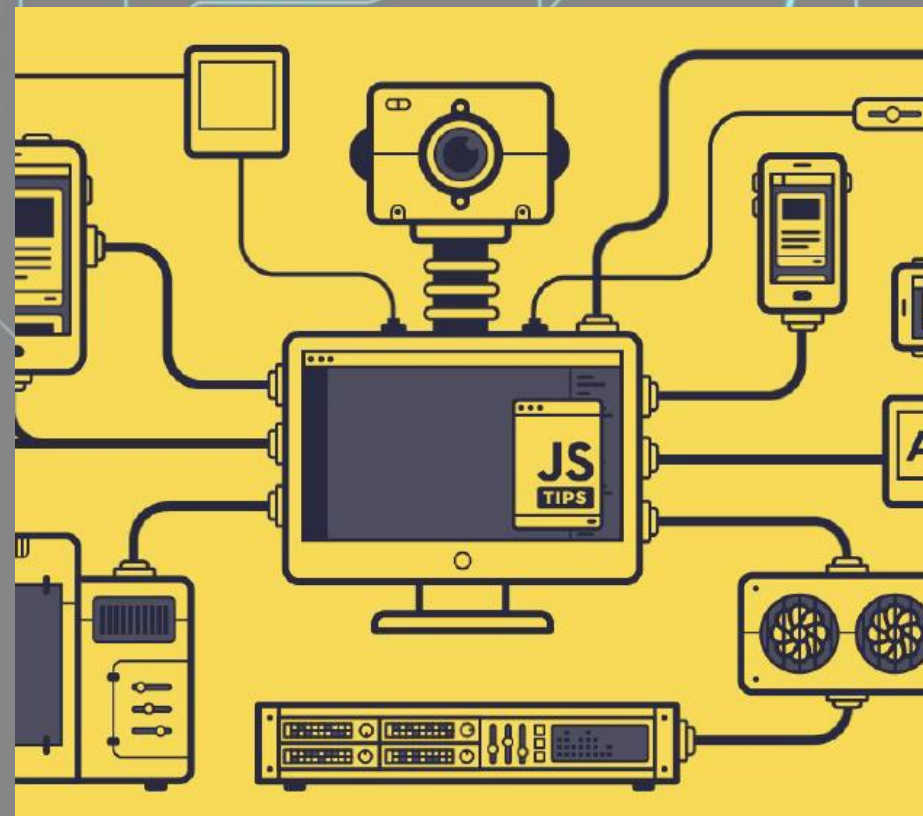
JavaScript



# JavaScript

Nasceu para ser utilizada no lado do cliente (navegador), para dar “vida” às páginas, até então estáticas.

Evoluiu ao longo dos anos e hoje é utilizada tanto no lado do cliente (front), quanto do servidor (back).



# JavaScript

O diferencial dela está no uso do Engine V8, um motor gráfico de código aberto conhecido por ter um alto desempenho nos sistemas e aplicações em que estiver instalado.

O navegador Google Chrome, por exemplo, tem ele por padrão.





# JavaScript



## Interno

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Hello World Example</title>
  <script>
    alert('Hello, World!');
  </script>
</head>
<body>
</body>
</html>
```

## Externo

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Hello World Example</title>
</head>
<body>

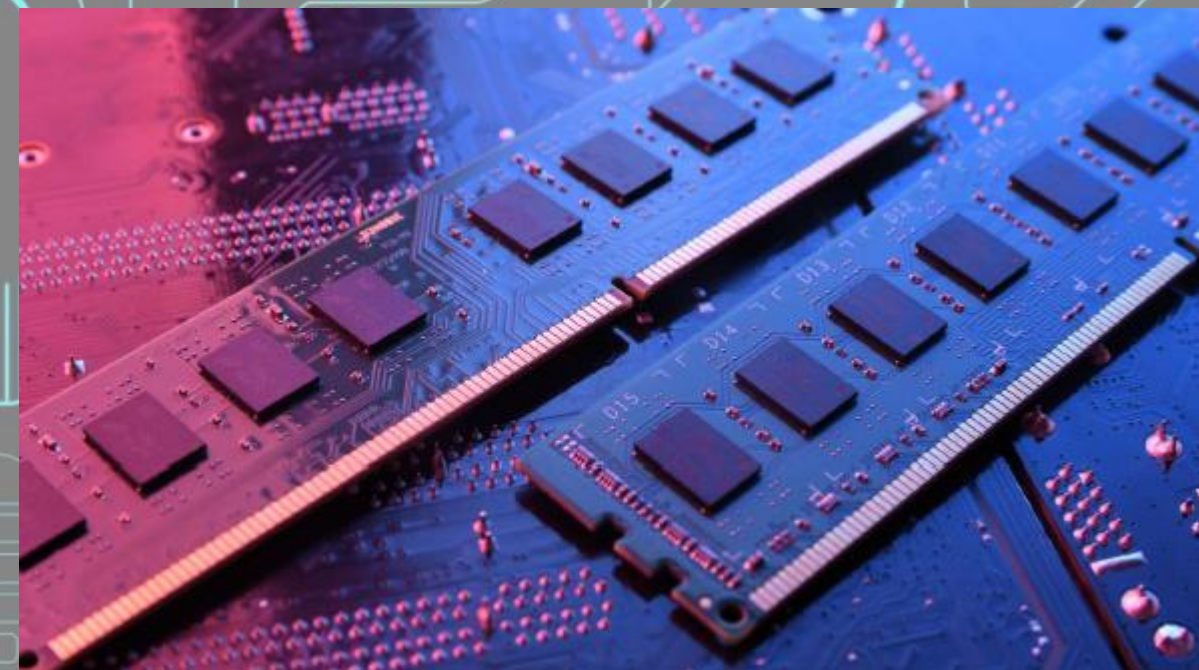
  <!-- end of page content here-->
  <script src="js/service.js"></script>
  <script src="js/app.js"></script>

</body>
</html>
```

# Variáveis



- O computador armazena os dados que são utilizados nos algoritmos na memória.
- A memória do computador é sequencial e dividida em posições
- Cada posição possui um número que indica a sua posição





# Variáveis e Tipos de Dados



Dentro dos tipos de dados primitivos, temos quatro categorias:

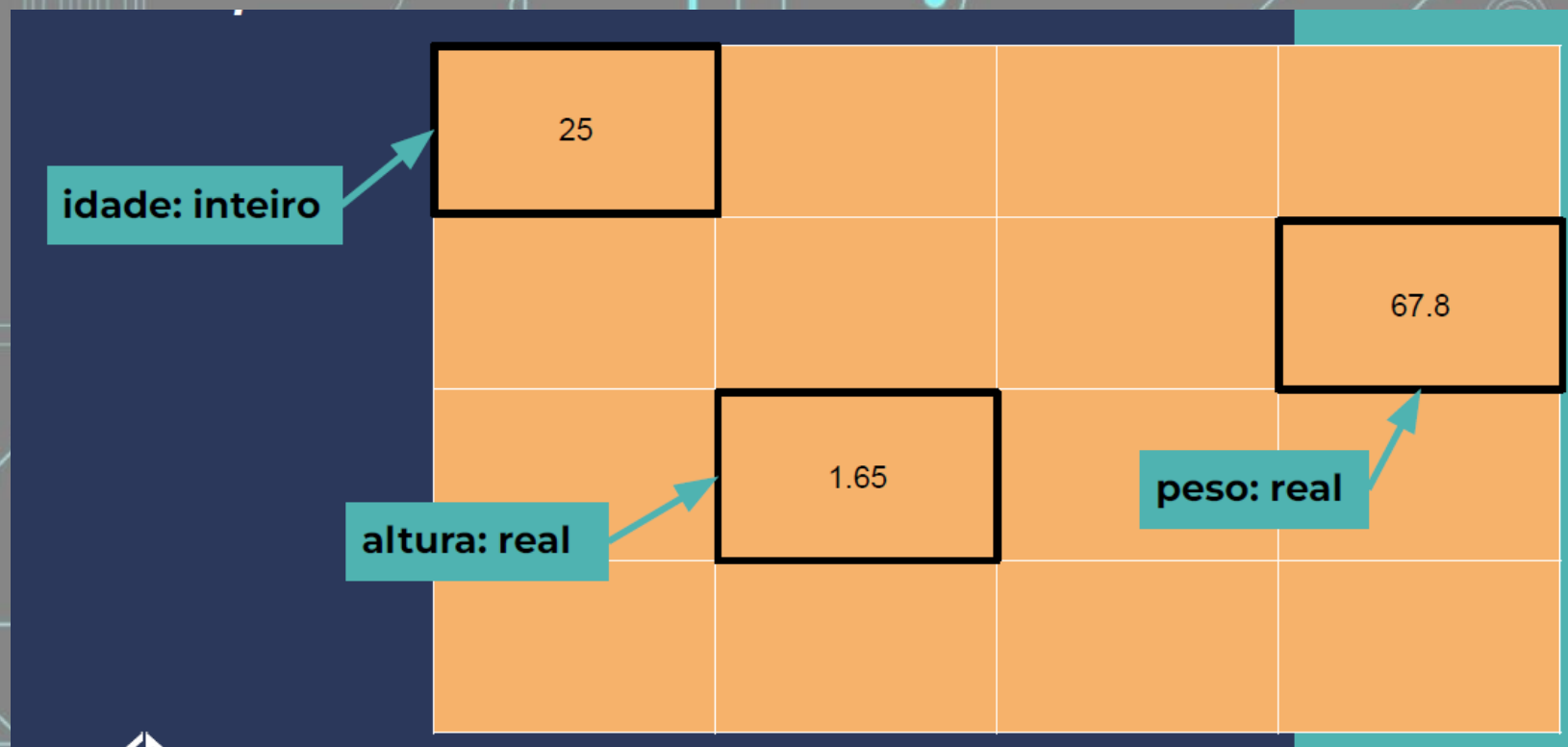
- Texto: possibilita trabalhar com uma sequência de caracteres. Ex: “Programa Starter”
- Inteiro: usado apenas para números inteiros negativos ou positivos. Ex: 1, 5, 8.
- Real: serve para qualquer número real com valores decimais. Ex: 1.21, 5.74, 8.14
- Lógico: com alternativas, sim ou não, verdadeiro ou falso.



# Variáveis e Tipos de Dados



Representação básica da memória de um computador



# Variáveis

## Regras para nomeação de variáveis:

1. Deve começar com uma letra
2. Os próximos caracteres podem ser letras ou números
3. Não pode utilizar nenhum símbolo, exceto \_
4. Não pode conter espaços em branco
5. Não pode conter letras com acentos
6. Não pode ser uma palavra reservada

- ## Regras para nomeação de variáveis:
1. Deve começar com uma letra
  2. Os próximos caracteres podem ser letras ou números
  3. Não pode utilizar nenhum símbolo, exceto \_
  4. Não pode conter espaços em branco
  5. Não pode conter letras com acentos
  6. Não pode ser uma palavra reservada

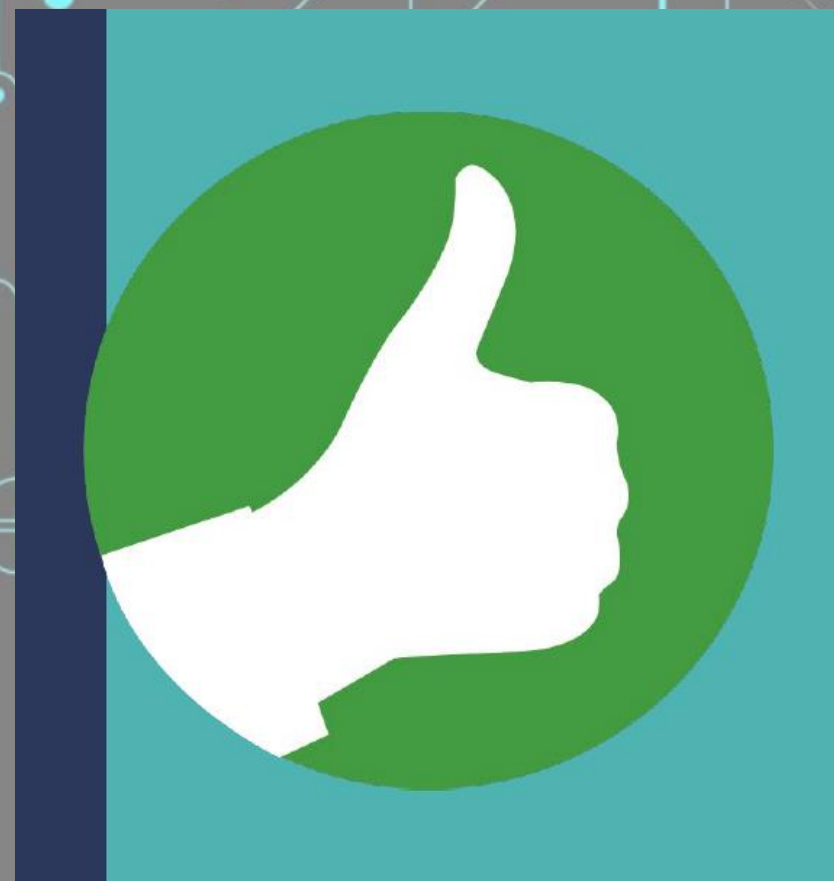




# Variáveis

## Exemplos de identificadores válidos

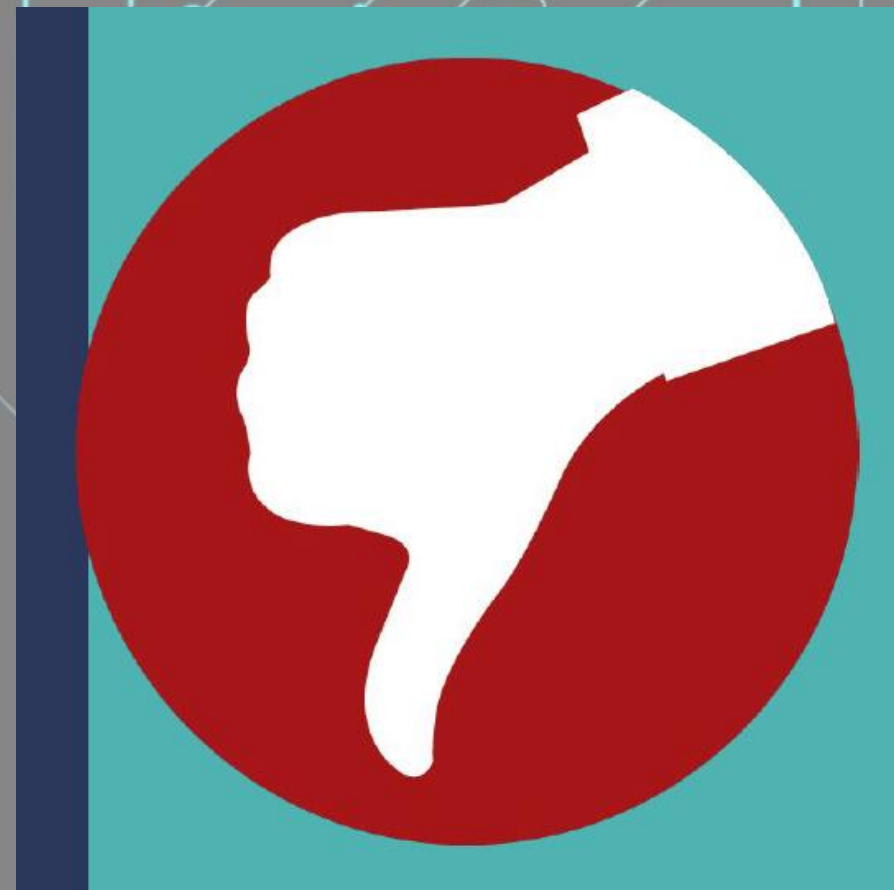
- idade
- Nome
- altura\_pessoa
- alturaPessoa
- documento1
- numero\_inteiro



# Variáveis

## Exemplos de identificadores inválidos

- relógio
- 9cpf
- inteiro
- verdadeiro
- nome de usuário
- compu&tador



# Operadores Aritméticos



|    |                  |
|----|------------------|
| +  | Adição           |
| -  | Subtração        |
| *  | Multiplicação    |
| /  | Divisão Inteira  |
| %  | Resto da divisão |
| ++ | Incremento       |
| -- | Decremento       |
| ** | Exponenciação    |



# Operadores de Atribuição



|    |                             |
|----|-----------------------------|
| =  | Atribuição                  |
| += | Atribuição de adição        |
| -= | Atribuição de subtração     |
| *= | Atribuição de multiplicação |
| /= | Atribuição de divisão       |
| %= | Atribuição de resto         |

# Operadores Relacionais

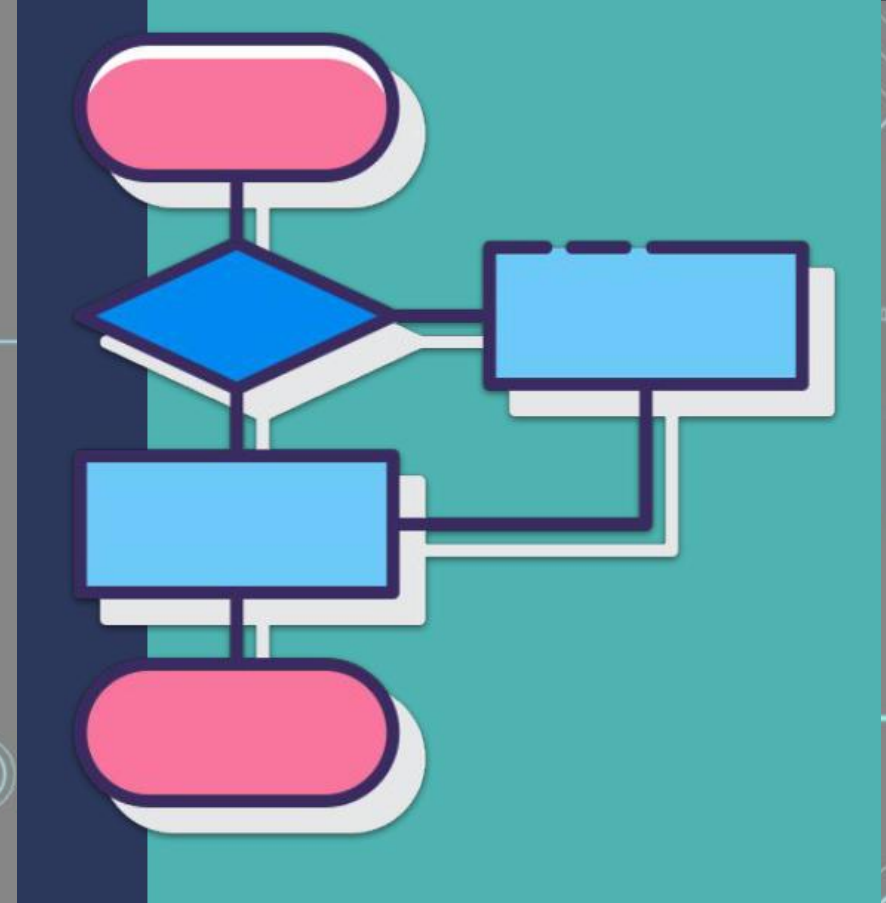


|    |                  |
|----|------------------|
| >  | Maior que        |
| <  | Menor que        |
| >= | Maior ou igual a |
| <= | Menor ou igual a |
| == | Igual a          |
| != | diferente de     |

# Condicionais

Utilizamos as estruturas de condição quando o computador precisa tomar decisões e seguir determinados fluxos. Por exemplo:

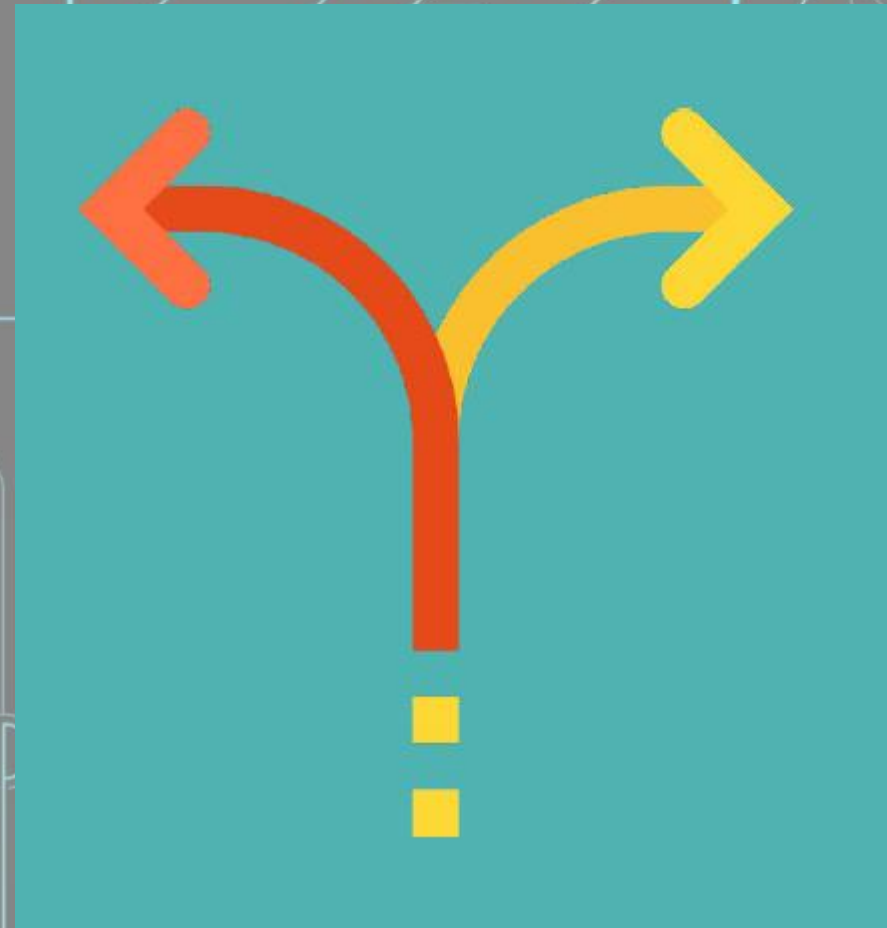
- Se a luz do semáforo estiver verde, então eu posso continuar dirigindo. Se a luz estiver vermelha, então eu devo parar o carro antes da faixa;
- Se estudei bastante, então posso fazer a prova tranquilamente.
- Se estiver chovendo, então eu vou sair com o guarda-chuva;





# Estruturas Condicionais

- Estruturas condicionais servem para desviar o fluxo normal de um programa.
- Os desvios do fluxo normal de um programa existem devido a condições impostas.



# Estruturas Condicionais - Exemplo



Se chover não vou ao parque

Condicional

Condição

Ação a ser  
realizada, caso a  
condição seja  
VERDADEIRA

# Estruturas Condicionais - Exemplo



```
const chover = true

if (chover) {
  console.log("não vou ao
parque")
}
```

No código ao lado a mensagem **“não vou ao parque”** apenas será exibida caso a condição imposta no condicional if seja verdadeira. E nesse caso é verdadeira.



# Estruturas Condicionais - Exemplo



```
const idade = 4

if (idade < 12) {
  console.log("criança")
} else if (idade < 18) {
  console.log("adolescente")
} else if (idade < 60) {
  console.log("adulto")
} else {
  console.log("idoso")
}
```

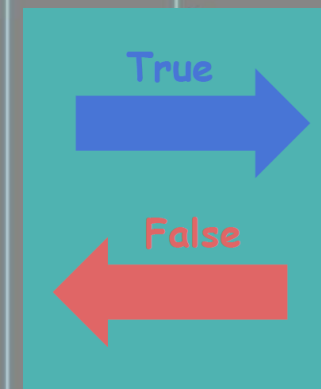
Ainda é possível existir  
vários testes diferentes em  
um mesmo conjunto de  
condicionais

# Operadores Lógicos



- Operadores que atuam entre expressões lógicas
- Realizam **conjunção**, **disjunção** ou **negação**
- Nos ajudam a analisar mais de uma expressão por vez em um condicional
- Existem basicamente 3 operadores lógicos

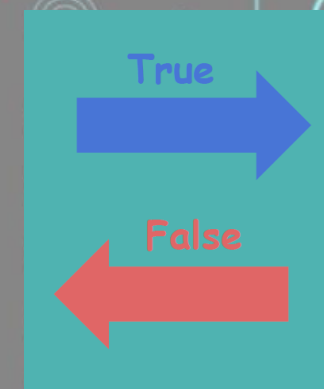
| Operador                        | Nome | Operação  |
|---------------------------------|------|-----------|
| <b>and</b> ou <b>&amp;&amp;</b> | e    | Conjunção |
| <b>or</b> ou <b>  </b>          | ou   | Disjunção |
| <b>not</b> ou <b>!</b>          | não  | Negação   |



# Operadores Lógicos



| Operador                        | Nome | Operação  |
|---------------------------------|------|-----------|
| <b>and</b> ou <b>&amp;&amp;</b> | e    | Conjunção |
| <b>or</b> ou <b>  </b>          | ou   | Disjunção |
| <b>not</b> ou <b>!</b>          | não  | Negação   |





# Operadores Lógicos – AND &&



| Teste   | Resultado |
|---------|-----------|
| T and T | T         |
| T and F | F         |
| F and T | F         |
| F and F | F         |

Esse operador é utilizado para avaliar se duas condições são verdadeiras.  
F = False, T = True

# Operadores Lógicos – AND &&



Verificar se o valor armazenado em uma certa variável X está entre o intervalo [10,25]

```
if(x >= 10) {  
  if(x <= 25) {  
    console.log("está no  
    intervalo")  
  }  
}
```

```
if(x >= 10 && x <= 25) {  
  console.log("está no intervalo")  
}
```

# Operadores Lógicos – OR ||



| Teste  | Resultado |
|--------|-----------|
| T ou T | T         |
| T ou F | T         |
| F ou T | T         |
| F ou F | F         |

Esse operador é utilizado para avaliar se ao menos uma expressão é verdadeira.  
F = False, T = True



# Operadores Lógicos – OR ||



Verificar se o valor armazenado em uma certa variável X fora do intervalo [10,25]

```
if (x < 10) {  
  console.log("fora do intervalo")  
} else if(x > 25) {  
  console.log("fora do intervalo")  
} else {  
  console.log("dentro do intervalo")  
}
```

```
if (x < 10 || x > 25) {  
  console.log("fora do intervalo")  
} else {  
  console.log("dentro do  
intervalo")  
}
```

# Operadores Lógicos – NOT !



| Teste | Resultado |
|-------|-----------|
| !T    | F         |
| !F    | T         |

Esse operador é utilizado para negar o valor lógico de uma expressão  
F = False, T = True

# Ordem de precedência



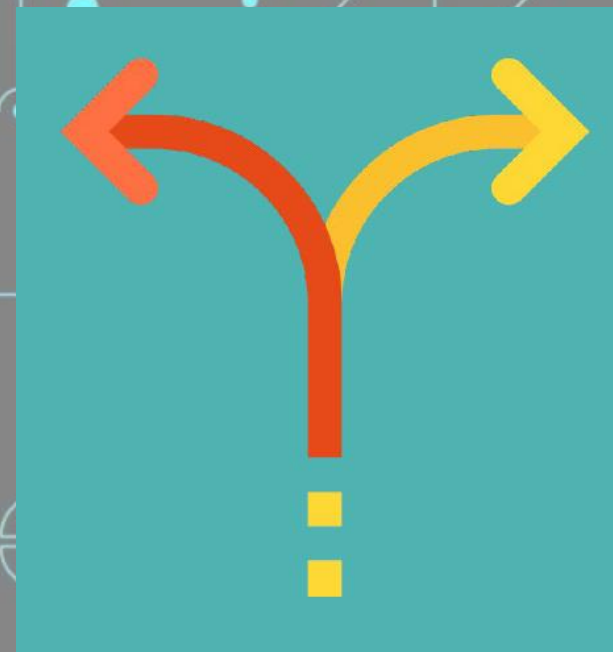
|             |                                          |
|-------------|------------------------------------------|
| Aritméticos | ( )                                      |
|             | **                                       |
|             | * / %                                    |
|             | + -                                      |
| Relacionais | Todos<br>Sempre da esquerda para direita |
| Lógicos     | NOT !                                    |
|             | AND &&                                   |
|             | OR                                       |



# Estruturas Condicionais - Switch



- A estrutura condicional switch permite executar um bloco de código diferente de acordo com cada opção especificada.
- Seu uso é indicado quando os valores a serem analisados nessas condições são pré-definidos

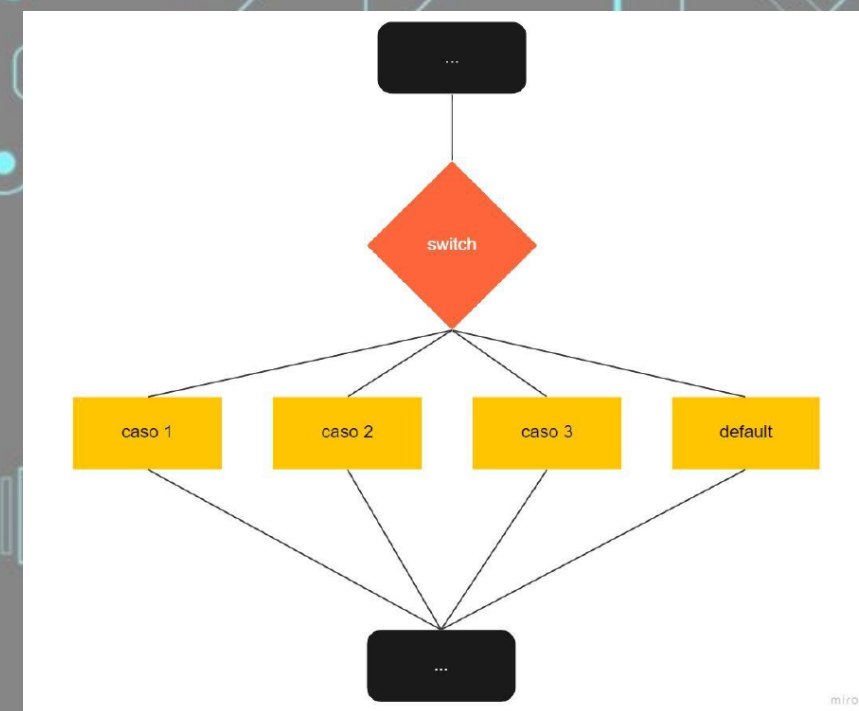


# Estruturas Condicionais - Switch



```
const operacao = 'somar'

switch (operacao) {
  case 'somar':
    console.log('somando...')
    break;
  case 'multiplicar':
    console.log('multiplicando...')
    break;
  default:
    console.log('default...')
    break;
}
```



# Obrigado

Desenvolvimento Web para Adolescentes

Pablo Garcia

JavaScript

