

Exemplo de Calculadora com TypeScript ↩️

1. INDEX

1. Introdução

1. [Objetivo.](#)
2. [Pre-requisitos.](#)
3. [benefícios.](#)

2. [Descrição.](#)

3. [Exemplos.](#)

4. [Conteúdo01](#)

5. [Referências.](#)

6. [Histórico.](#)

2. CONTEÚDO

1. Introdução

1. Objetivo:

1. Este documento tem como objetivo descrever todos os comandos typescript necessários para criar uma calculadora.
2. [↩️](#)

2. Pre-requisitos:

1. Programas instalados:
 1. nodejs;
 2. Vscod;
 3. Compilador typescript.
 4. tsc-init
- 2.
3. [↩️](#)

3. Benefícios:

1. Aprender os comandos da linguagem typescript.
2. [↩️](#)

2. Descrição

1. Escreve em detalhe o conteúdo deste conteúdo.

2. 

3. Exemplos.

1. Calculadora:

1. Código typescript

```
export class Calculator {
  private current = 0;
  private memory = 0;
  private operator: string;
  protected processDigit(digit: string, currentValue: number) {
    if (digit >= "0" && digit <= "9") {
      return currentValue * 10 + (digit.charCodeAt(0) -
"0".charCodeAt(0));
    }
  }
  protected processOperator(operator: string) {
    if (["+","-","*","/"].indexOf(operator) >= 0) {
      return operator;
    }
  }
  protected evaluateOperator(
    operator: string,
    left: number,
    right: number
  ): number {
    switch (this.operator) {
      case "+":
        return left + right;
      case "-":
        return left - right;
      case "*":
        return left * right;
      case "/":
        return left / right;
    }
  }
  private evaluate() {
    if (this.operator) {
      this.memory = this.evaluateOperator(
        this.operator,
        this.memory,
        this.current
      );
    } else {
      this.memory = this.current;
    }
    this.current = 0;
  }
}
```

```
}
public handleChar(char: string) {
  if (char === "=") {
    this.evaluate();
    return;
  } else {
    let value = this.processDigit(char, this.current);
    if (value !== undefined) {
      this.current = value;
      return;
    } else {
      let value = this.processOperator(char);
      if (value !== undefined) {
        this.evaluate();
        this.operator = value;
        return;
      }
    }
  }
  throw new Error(`Unsupported input: '${char}'`);
}
public getResult() {
  return this.memory;
}
}
export function test(c: Calculator, input: string) {
  for (let i = 0; i < input.length; i++) {
    c.handleChar(input[i]);
  }
  console.log(`result of '${input}' is '${c.getResult()}'`);
}
```

2. item 02.

3. 

4. Conteúdo01

1. item 01.

2. item 02.

3. 

5. REFERÊNCIAS

1. #

2. #

3. #

4. #

5. #

6. 

6. HISTÓRICO

1. dd/mm/2021

- 

2. dd/mm/2021

- ☐ Criar este documento baseado no Calculator.md ;
- ☐ Escrever tópico Objetivos;
- ☐ Escrever tópico Pre-requisitos
- ☐ Escrever tópico Benefícios
- ☐ Escrever tópico Descrição
- ☐ Escrever tópico Conteúdo 02
- ☐ Escrever tópico Referências
- ☐ Atualizar o histórico deste documento.
- ☐ Ler no dia seguinte este documento para checar os erros de português.
- 

