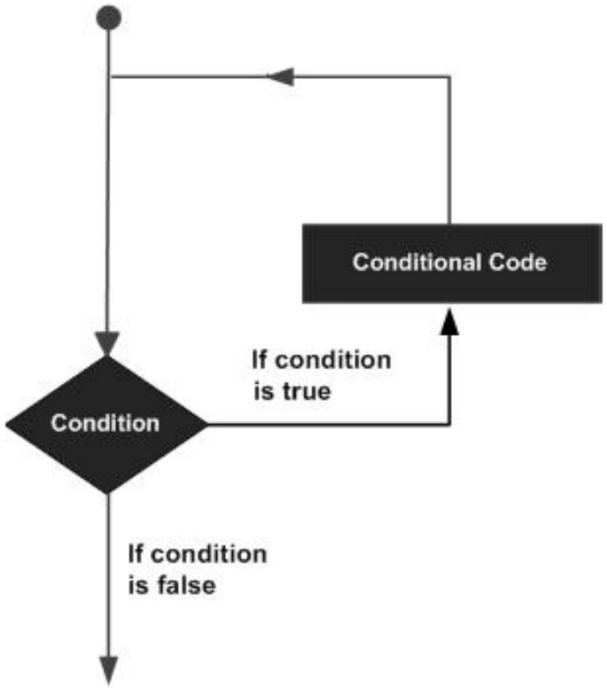
Programação para Design

01 - Introdução ao TypeScript
Prof. Jefferson de Carvalho Silva

Decisões, Loops e Funções
Array, Tuplas, Enum e Union

Decisões, Loops e Funções

- Decisão (if-else)



Decisões, Loops e Funções

- Decisão (if-else)

TS Decisao.ts

```ts
if (true) {
    console.log('This will always executed.');
}

if (false) {
    console.log('This will never executed.');
}
```

Decisões, Loops e Funções

- Decisão (if-else)

TS Decisao.ts > ...

```ts
var x: number = 10, y = 20;

if (x < y) {
    console.log('x is less than y');
}
```

Decisões, Loops e Funções

- Decisão (if-else)

TS Decisao.ts > ...

```ts
var x: number = 10, y = 20;

if (x > y) {
    console.log('x is greater than y.');
}
else {
    console.log('x is less than or equal to y.'); //This will be executed
}
```

Decisões, Loops e Funções

- Decisão (if-else)

TS Decisao.ts

```ts
if (true) {
    console.log('This will always executed.');
}

if (false) {
    console.log('This will never executed.');
}
```

Decisões, Loops e Funções

- Decisão (if-else)

TS Decisao.ts > ...

```ts
let x: number = 10, y = 20;

x > y ?
    console.log('x is greater than y.') :
    console.log('x is less than or equal to y.')
```

Exercícios

Usando o TypeScript, faça:

1) Um programa que calcule o IMC, dado o peso e a altura.
2) Um programa que calcule se um número é par ou ímpar.
3) Um programa que indique o maior e o menor de três números.
4) Um programa que recebe duas notas parciais e calcula se o aluno irá ou não para a prova final, de acordo com as regras da UFC. Caso o aluno vá pra final, também calcule se o aluno irá ser aprovado dependendo da nota da final.

Decisões, Loops e Funções

- Switch

TS Decisao.ts > ...

```ts
let day: number = 4;

switch (day) {
    case 0:
        console.log("It is a Sunday.");
        break;
    case 1:
        console.log("It is a Monday.");
        break;
    case 2:
        console.log("It is a Tuesday.");
        break;
    case 3:
        console.log("It is a Wednesday.");
        break;
    case 4:
        console.log("It is a Thursday.");
        break;
    case 5:
        console.log("It is a Friday.");
        break;
    case 6:
        console.log("It is a Saturday.");
        break;
    default:
        console.log("No such day exists!");
        break;
}
```

Exercício

1) Implemente uma calculadora de quatro operações básicas onde um switch irá escolher a operação a ser efetuada sobre dois operandos.

Decisões, Loops e Funções

- For

TS Decisao.ts > ...

```ts
for (let i = 0; i < 3; i++) {
    console.log ("Block statement execution no." + i);
}
```

Decisões, Loops e Funções

- For

```ts
let arr = [10, 20, 30, 40];

for (var val of arr) {
  console.log(val); // prints values: 10, 20, 30, 40
}
```

```ts
let str = "Hello World";

for (var char of str) {
  console.log(char); // prints chars: H e l l o  W o r l d
}
```

Exercícios

1) Faça um programa que ao ler uma array de inteiros indique qual é o menor e o maior valor do array.
2) Faça um programa que calcule a média de um array de inteiros.
3) Faça um programa que calcule o enésimo termo da série Fibonacci.
4) Faça o programa que dado dois arrays, calcule a soma do elemento-i do primeiro array com o elemento-i do segundo array e armazene a soma na iésima posiçao de um array resultante.

Decisões, Loops e Funções

- While

TS Decisao.ts > ...

```ts
let i: number = 2;

while (i < 4) {
    console.log( "Block statement execution no." + i )
    i++;
}
```

Decisões, Loops e Funções

- While

TS Decisao.ts > ...

```ts
let i: number = 2;

while (i < 4) {
    console.log( "Block statement execution no." + i )
    i++;
}
```

Decisões, Loops e Funções

- While

TS Decisao.ts > ...

```ts
let i: number = 2;
do {
    console.log("Block statement execution no." + i )
    i++;
} while ( i < 4)
```

Exercício

1) Faça um programa que leia um array de inteiros e calcule a média dos elementos lidos até ler um número negativo dentro do array.

2) Refaça o programa anterior com um do-while.

Decisões, Loops e Funções

- Funções Clássicas

```
1  function Sum(x: number, y: number) : number {
2      return x + y;
3  }
4
5  Sum(2,3); // returns 5
```

```
1  let Sum = function(x: number, y: number) : number
2  {
3      return x + y;
4  }
5
6  Sum(2,3); // returns 5
```

Decisões, Loops e Funções

- Funções Arrow

```
1  let sum = (x: number, y: number): number => {
2      return x + y;
3  }
4
5  sum(10, 20); //returns 30
```

```
1  let sum = (x: number, y: number) => x + y;
2
3  sum(3, 4); //returns 7
```

Decisões, Loops e Funções

- Funções (métodos de classes)

```
class Employee {
    empCode: number;
    empName: string;

    constructor(code: number, name: string) {
        this.empName = name;
        this.empCode = code;
    }

    display = () => console.log(this.empCode +' ' + this.empName)
}
let emp = new Employee(1, 'Ram');
emp.display();
```

Exercícios

1) Crie uma função clássica que receba uma array de inteiros e retorne a média dos números do array.

2) Cria uma função arrow que recebe duas notas e retorna se o aluno está aprovado ou não.

3) Cria uma classe Empregado e uma método de cálculo de salário de acordo com a titulação. Se o empregado for nível técnico, ele receber um salário base de R$ 1000,00. Caso seja graduado, um aumento de 30% sobre o salário base. Mestre, uma aumento de 50%. Doutor, o dobro do salário base + 20% sobre esse dobro.

Programação para Design

02 - Introdução ao TypeScript
Prof. Jefferson de Carvalho Silva

Array, Tuplas, Enum e Union

Tipo Array

- Declarando

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];
```

```
let fruits: Array<string> = ['Apple', 'Orange', 'Banana'];
```

```
let arr = [1, 3, 'Apple', 'Orange', 'Banana', true, false];
```

```
let fruits: Array<string>;
fruits = ['Apple', 'Orange', 'Banana'];

let ids: Array<number>;
ids = [23, 34, 100, 124, 44];
```

Tipo Array

- Arrays de tipos múltiplos

```
let values1: (string | number)[] = ['Apple', 2, 'Orange', 3, 4, 'Banana'];
// or
let values2: Array<string | number> = ['Apple', 2, 'Orange', 3, 4, 'Banana'];
```

- Acessando elementos:

```
let fruits: string[] = ['Apple', 'Orange', 'Banana'];
fruits[0]; // returns Apple
fruits[1]; // returns Orange
fruits[2]; // returns Banana
fruits[3]; // returns undefined
```

Tipo Tuplas

- Tuplas podem conter dois valores de diferentes tipos.

```
var empId: number = 1;
var empName: string = "Steve";

// Tuple type variable
var employee: [number, string] = [1, "Steve"];
```

```
var employee: [number, string] = [1, "Steve"];
var person: [number, string, boolean] = [1, "Steve", true];

var user: [number, string, boolean, number, string];// declare tuple variable
user = [1, "Steve", true, 20, "Admin"];// initialize tuple variable
```

Tipo Tuplas

- Array de Tuplas

```
var employee: [number, string][];
employee = [[1, "Steve"], [2, "Bill"], [3, "Jeff"]];
```

- Acessando os elementos da Tupla

```
var employee: [number, string] = [1, "Steve"];
employee[0]; // returns 1
employee[1]; // returns "Steve"
```

Tipo Tuplas

- Usando o push()

```
var employee: [number, string] = [1, "Steve"];
employee.push(2, "Bill");
console.log(employee); //Output: [1, 'Steve', 2, 'Bill']
```

- Métodos do Array

```
var employee: [number, string] = [1, "Steve"];

// retrieving value by index and performing an operation
employee[1] = employee[1].concat(" Jobs");
console.log(employee); //Output: [1, 'Steve Jobs']
```

Tipo Enum

- **Permite declarar um conjunto de constantes nomeadas;**
- **Podem ser:**
  - Numéricos
  - Strings ou
  - Heterogêneos.

Tipo Enum

- Tipo Numérico

```
enum PrintMedia {
    Newspaper = 1,
    Newsletter,
    Magazine,
    Book
}

function getMedia(mediaName: string): PrintMedia {
    if (  mediaName === 'Forbes' || mediaName === 'Outlook') {
        return PrintMedia.Magazine;
    }
 }

let mediaType: PrintMedia = getMedia('Forbes'); // returns Magazine
```

Tipo Enum

- Tipo String

```
enum PrintMedia {
    Newspaper = "NEWSPAPER",
    Newsletter = "NEWSLETTER",
    Magazine = "MAGAZINE",
    Book = "BOOK"
}
// Access String Enum
PrintMedia.Newspaper; //returns NEWSPAPER
PrintMedia['Magazine'];//returns MAGAZINE
```

Tipo Enum

- Tipo Heterogêneo

```
enum Status {
    Active = 'ACTIVE',
    Deactivate = 1,
    Pending
}

console.log(Status.Pending)
```

Tipo Enum

- Mapeamento reverso

```
enum PrintMedia {
    Newspaper = 1,
    Newsletter,
    Magazine,
    Book
}

PrintMedia.Magazine;    // returns  3
PrintMedia["Magazine"];// returns  3
PrintMedia[3];          // returns  Magazine
```

Tipo Union

- Simplesmente permitir o uso de mais de um tipo de dado para uma variável

```
let code: (string | number);
code = 123;   // OK
code = "ABC"; // OK
code = false; // Compiler Error

let empId: string | number;
empId = 111; // OK
empId = "E111"; // OK
empId = true; // Compiler Error
```

Tipo Union

- Usando em Funções

```
function displayType(code: (string | number))
{
    if(typeof(code) === "number")
        console.log('Code is number.')
    else if(typeof(code) === "string")
        console.log('Code is string.')
}

displayType(123); // Output: Code is number.
displayType("ABC"); // Output: Code is string.
displayType(true); //Compiler Error: Argument of type 'true' is not assignable to a parameter of type string | number
```

Referências

- https://www.tutorialspoint.com/typescript/index.htm

- https://www.tutorialsteacher.com/typescript