- Conhecendo Funções
  - Funções vs Procedimentos
- Funcao Nao Nomeada e Funcao Imediatamente Invocada

Conhecendo Funções

Uma função corresponde a um bloco de código que precisa ser repetidor durante o programa. Pense o seguinte, *vamos supor que toda vez que você prepara comida, é necessário você lavar a louça. Durante todo o dia, você faz 3 refeições, então deveria lavar 3 vezes a louça. Mas imagine o seguinte, ao invés de lavar mais vezes, você poderia escolher lavar tudo ao anoitecer*. Desse modo, você faz apenas uma vez uma acao que precisaria ser repetida.

Bom, uma função serve para escrevermos menos e agilizar o programa. Por exemplo, existem vários programas que possuem a função salvar em várias janelas, imagine usar o mesmo código para todoas as janelas, por isso escrevemos a função e **reutilizamos**. A estrutura básica de uma função é:

```
function nomeFuncao(){
        //condigo
}
```

Uma função também possui parâmetros, que sao os valores passados dentro () dela, confira:

```
function sayMyName(name){
  console.log(`My name is ${name}`);
};
//Output: My name is Gabriel
sayMyName(`Gabriel`);

//Output: My name is Fernando
sayMyName(`Fernando`);
```

A função acima tem uma funcionalizado, mostrar o nome. No entanto, ela é **mais eficiente** do que o código abaixo, já que para mudar de nome, eu posso passar o nome da pessoa para dentro dos () da função.

```
console.log(`My name is Gabriel`)
console.log(`My name is Fernando`)
```

Funções vs Procedimentos

A diferença entre Procedimentos e Funções é que o primeiro **NÃO RETORNAR VALOR**. Pense da seguinte forma, quando desejamos enviar um email para alguém, é preciso obter o retorno daquilo? Frequentemente, não, por isso devemos apenas enviar e nada mais. Quando desejamos mostrar uma informação escrita pelo usuário na tela (como o console.log), é necessário obter algum retorno? Não, apenas precisamos escrever.

Confira a seguinte observação:

> Procedimentos são estruturas que agrupam um conjunto de comandos, que são executados quando o procedimento é chamado.

> Funções são procedimentos que retornam um único valor ao final de sua execução.

```
//Funcao
function quadrado(valor){
  return valor * valor;
};
//Procedimento
function quadrado2(valor2){
  valor2 * valor2;
};

console.log(quadrado(10) + quadrado(10))//Usando os retornos da funcao
```

Confira mais um exemplo:

```
function incrementarJuros(valor, percentualJuros){
  const acrescimo = (percentualJuros / 100) * valor;
  return valor + acrescimo;
};

console.log(incrementarJuros(10,50));
```

Funcao Nao Nomeada e Funcao Imediatamente Invocada

Funções são tratadas dentro do JS como Objetos. Sendo assim, elas podem ser referenciadas (chamadas pelo nome), assim como variaveis.

```
console.log(main())
/*Output expected:
ƒ main(){
  const peso = 85;
  const altura = 1.80;

  const imc = calcular(peso,altura)
  console.log(classificar(imc))
  console.log(imc)
}
*/
```

Existe um conceito de Funcao Nao Nomeada e Funcao. Quando usamos funcoes nao nomeadas, significa que elas sao imediatamente invocadas, ou seja, sao criadas e executadas imediatamente apos sua criacao.

Confira um exemplo de **FUNCAO IMEDIATAMENTE INVOCADA (TAMBEM NAO NOMEADA):**

```
(function (){
  const peso = 85;
  const altura = 1.80;

  const imc = calcular(peso,altura)
  console.log(classificar(imc))
  console.log(imc)
})();
```