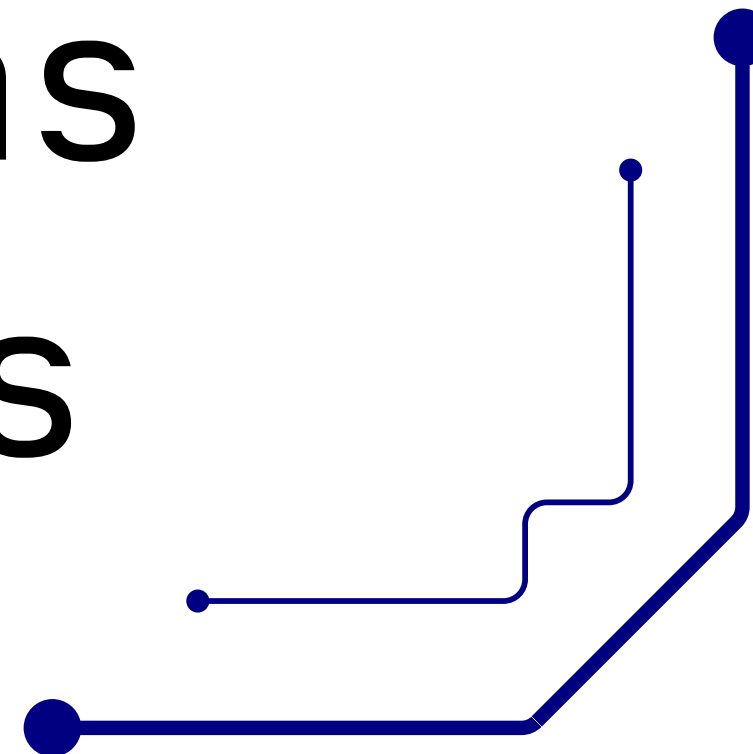
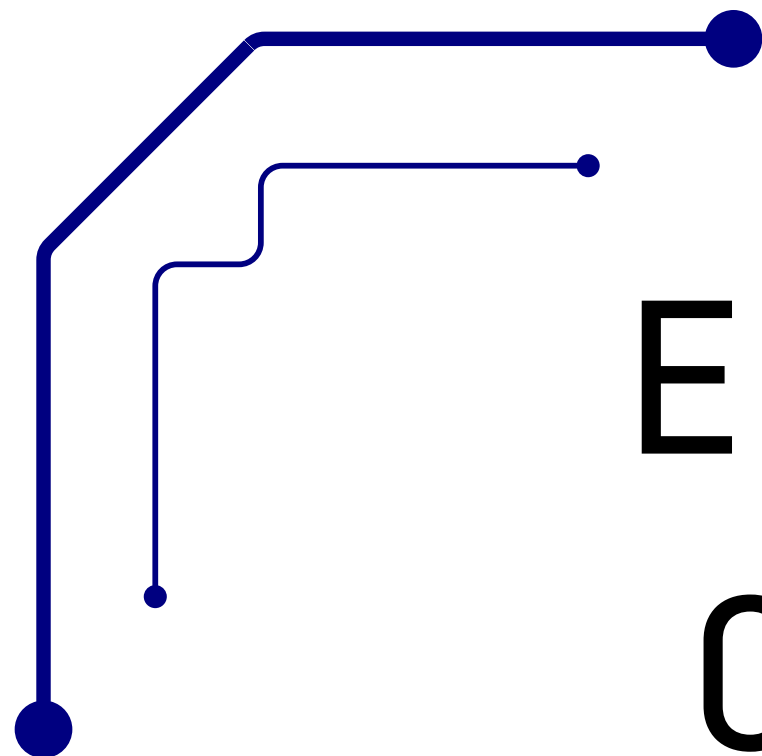




## Estruturas Compostas



TTech Inc



Perfil (Website)



## Introdução

Todos os valores e dados podem ser **locais** ou **globais**.

O seu **local** e espaço de uso determina o seu **escopo**.

### Exemplo:

- Se um dado apenas pode ser utilizado dentro de uma função então o seu escopo é a função.

# Escopo e Retorno



## Exemplo

Utilizando o 'do' podemos definir um espaço no código que possui o seu próprio escopo.

Os dados criados no espaço apenas poderão ser utilizados nele.

```
do
    -- código
end
```



## Funções e Escopo

Funções são estruturas sujeitas a escopo

- Dados declarados em funções **apenas** podem ser utilizados dentro dela

E se quiséssemos utilizar esse dado **fora** da função?

- Precisamos **retornar** esse dado

```
local num = math.ceil(5.4)
```



## Retorno

A função retorna um valor.

```
function somar(n1,n2)
    local soma = n1+n2
    return soma
end

local valor = somar(10,20)
```



## Importante:

- Uma função apenas pode possuir uma instrução de *return* a ser executada
- `0 return` marca o ponto de finalização de uma função



## Retorno Múltiplo

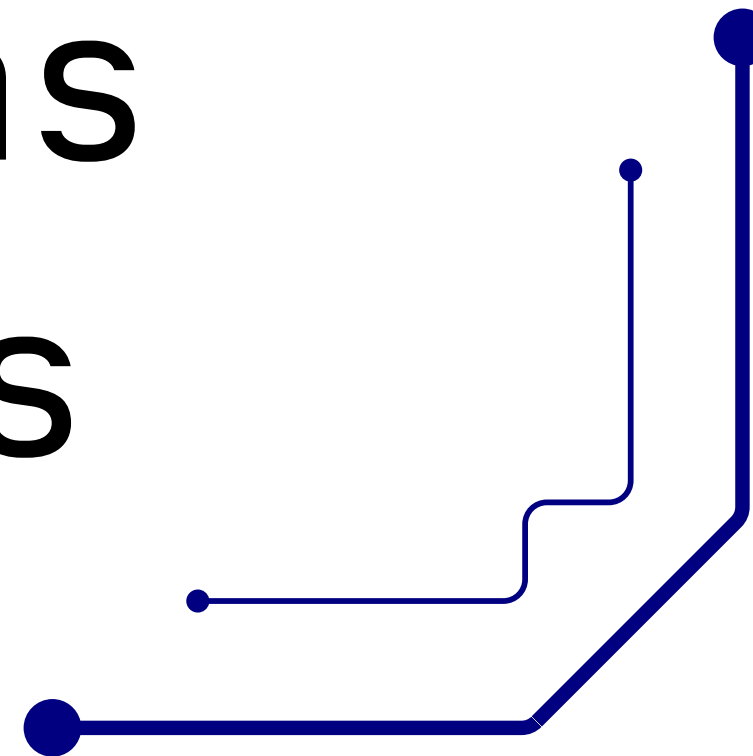
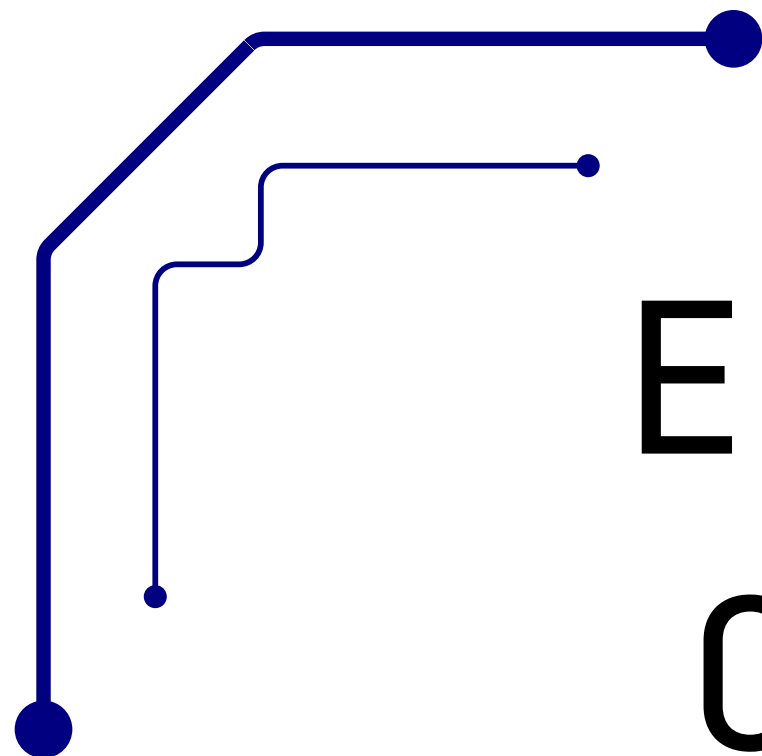
A função retorna vários valores.

```
function calc(n1,n2)
    local soma = n1+n2
    local sub = n1-n2
    return soma, sub
end

local soma, sub = calc(10,20)
```



## Estruturas Compostas



TTech Inc



Perfil (Website)