



06 { ..

# Rotinas e Funções



**Formador:** Ricardo Mourão

} ..



# Rotinas e Funções

{



range()

input() randint()

len()

append() int()

print()

}



**Formador:** Ricardo Mourão

# Rotinas e Funções



print()

```
import sys

def custom_print(*args, sep=' ', end='\n', file=None):
    """
    Uma função customizada que imita o comportamento da função print do Python.
    :param args: Argumentos variáveis. Todos os objetos passados serão convertidos em strings.
    :param sep: Separador entre os objetos. Padrão é um espaço.
    :param end: String adicionada ao final da saída. Padrão é uma nova linha.
    :param file: Objeto de arquivo ou stream onde a saída será escrita. Padrão é sys.stdout.
    """
    # Converte todos os argumentos para strings
    str_args = [str(arg) for arg in args]
    # Concatena os argumentos com o separador
    output = sep.join(str_args)
    # Adiciona o final de linha ou caractere especificado
    output += end

    # Escreve a saída no stream especificado, ou sys.stdout por padrão
    if file is not None:
        file.write(output)
    else:
        sys.stdout.write(output)

# Exemplo de uso da função custom_print
custom_print("Olá,", "mundo!", sep="-", end="*\n")
custom_print("Teste", "de", "função", "customizada.")
```



**Formador:** Ricardo Mourão



# Rotinas e Funções



```
-----  
REGISTO DE FUNCIONÁRIOS  
-----  
  
-----  
Calculadora dinâmica  
-----  
  
-----  
Jogo da Adivinha  
-----
```



**Formador:** Ricardo Mourão

# Rotinas e Funções



{

```
print( '-----' )
print( '          REGISTO DE FUNCIONÁRIOS          ' )
print( '-----' )

print( '-----' )
print( '          Calculadora dinâmica          ' )
print( '-----' )

print( '-----' )
print( '          Jogo da Adivinha          ' )
print( '-----' )
```

}

# Rotinas e Funções



{

```
print('-----')
```

```
print('-----')
```

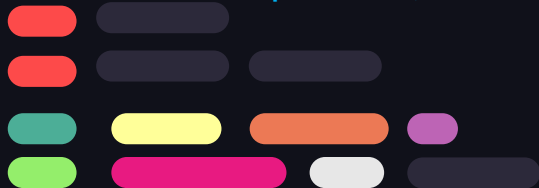
```
print('-----')
```

```
print('-----')
```

```
print('-----')
```

```
print('-----')
```

}



**Formador:** Ricardo Mourão

Rotinas



# Rotinas e Funções

{



`insere_linha()`

`print('-----')`

}



**Formador:** Ricardo Mourão

# Rotinas e Funções



{

```
print( '-----' )
print( '          REGISTO DE FUNCIONÁRIOS          ' )
print( '-----' )

print( '-----' )
print( '          Calculadora dinâmica          ' )
print( '-----' )

print( '-----' )
print( '          Jogo da Adivinha          ' )
print( '-----' )
```

}





# Rotinas e Funções

```
{  
insere_linha()  
print( '          REGISTO DE FUNCIONÁRIOS          ', )  
insere_linha()  
insere_linha()  
print( '          Calculadora dinâmica          ', )  
insere_linha()  
insere_linha()  
print( '          Jogo da Adivinha          ', )  
insere_linha()  
}
```

# Rotinas e Funções



As funções são uma das principais construções em qualquer código. Elas permitem que se encapsule um bloco de código que realiza uma tarefa específica num único lugar. Uma vez definida, uma função pode ser chamada repetidamente de diferentes partes do código. Isto permite a reutilização de código e torna o seu programa mais **modular**, fácil de compreender e manter.

```
def nome_da_função (lista_de_parâmetros):
```

```
// corpo da função
```



**Formador:** Ricardo Mourão



# Rotinas e Funções

{



```
def nome_da_função (lista_de_parâmetros):
```

```
    // corpo da função
```

}



**Formador:** Ricardo Mourão



# Rotinas e Funções



Vamos ver na  
prática



**Formador:** Ricardo Mourão

# Rotinas e Funções



```
print( '-----' )
print( '          REGISTO DE FUNCIONÁRIOS          ' )
print( '-----' )
print( '-----' )
print( '          Calculadora dinâmica          ' )
print( '-----' )
print( '-----' )
print( '          Jogo da Adivinha          ' )
print( '-----' )
```



# Rotinas e Funções

```
{ def mensagem( msg ):  
    print('-----')  
    print(msg)  
    print('-----')  
  
mensagem("REGISTO DE FUNCIONÁRIOS")  
}
```



# Rotinas e Funções



Vamos ver na  
prática



**Formador:** Ricardo Mourão

# Rotinas e Funções

```
{ def contador(*num):  
    for valor in num:  
        print(f'{valor} ', end='')  
    print('FIM')  
  
    contador(1, 2, 3, 4, 5)  
  
    contador(2, 4, 6)  
  
    contador(1, 3, 5, 7, 9, 11, 13)
```

```
}
```





# Rotinas e Funções

```
{ def contador(num):  
    for valor in num:  
        print(f'{valor} ', end='')  
    print('FIM')
```

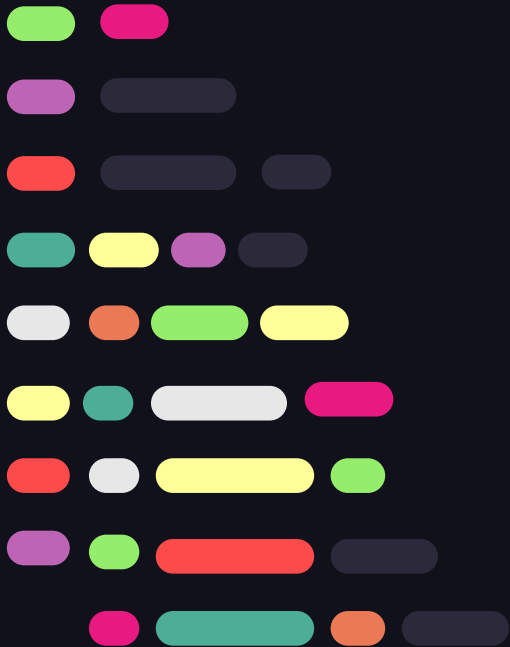
```
numeros = [1, 2, 3, 4, 5]
```

```
contador(numeros)
```

```
}
```



# PRÁTICA! Exercício 32

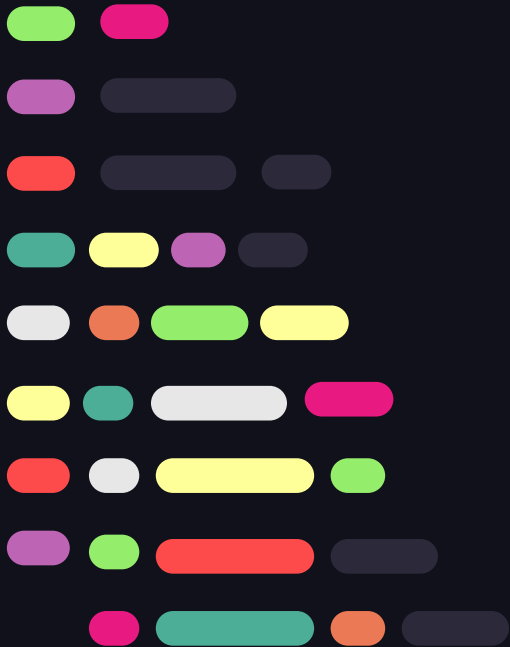


Crie um programa que tenha uma função chamada `area()` que receba as dimensões de um terreno e mostre a área do terreno.





# PRÁTICA! Exercício 33



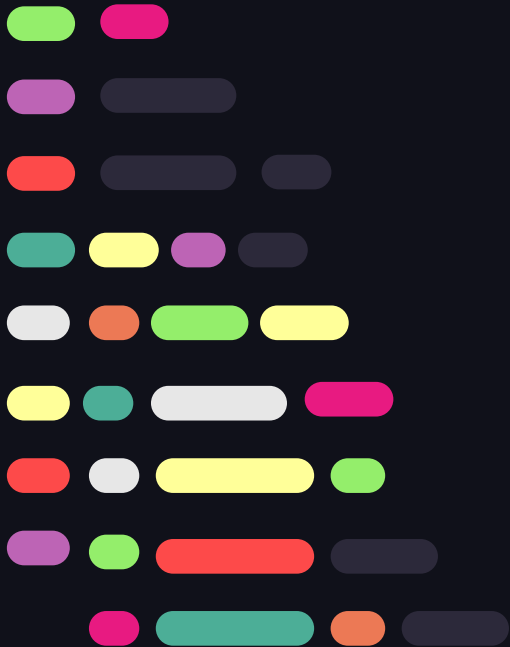
Crie um programa que tenha uma função que receba 3 parâmetros: início, fim e passo. O programa deve realizar 3 contagens através da função.

- a) De 1 até 20, de 2 em 2
- b) De 10 até 0, de 1 em 1
- c) Contagem personalizada





# PRÁTICA! Exercício 34



Crie um programa que tenha uma função que converta a temperatura de Celsius para Fahrenheit.

