

# PythonLab

Prof. Dr. Álvaro Campos  
Ferreira

# Comentários

# Variáveis

# Tipos das variáveis

# Tipos de variáveis

Para cada tipo de variável, as funções tem comportamento diferente.

- Texto (str)
- Número inteiro (int)
- Número real (float)
- Booleano (bool), True ou False

# Tipos de variáveis

Ao declarar uma variável, o Python determina seu tipo dinamicamente, uma variável pode mudar de tipo apenas declarando-a novamente:

```
idade = "Trinta e dois" # tipo texto
```

```
idade = 32 # tipo inteiro
```

# Listas

# Dicionários



# Controle de fluxo

# Laços for ou “para cada”

Com **for**, o bloco irá executar para cada elemento do objeto.

```
for cliente in clientes:  
    print(cliente)
```

# Laços while ou “enquanto”

O **while** executará o bloco “enquanto” a condição a direita dele resultar em verdadeiro, ou booleano True.

```
while nome == “Álvaro”:  
    print(“Olá, professor!”)
```

# Blocos if... else ou “se... senão”

O **if** ou “se”, executa o bloco se a condição for satisfeita.

```
if nome == “Álvaro”:
```

```
    print(“Bom dia, professor”)
```

```
else:
```

```
    print(“Bom dia, querido colega”)
```

# Funções

# Funções

Define-se funções com a palavra-chave **def** da palavra definir:

```
def somar_numeros(a,b):  
    return a + b
```

# Documentando funções

A forma padrão de documentar uma função chama-se Docstring e nada mais é que um bloco de comentário.

```
def somar_numeros(a,b):  
    """ Soma dois números.  
    """  
    return a + b
```

# Retornando valores

Uma função pode retornar um ou mais valores após a execução.

```
def somar_numeros(a,b):  
    """ Soma dois números.  
    """  
  
    return a + b
```



# Módulos

# Módulos de funções

É comum e boa prática gerar arquivos com funções relacionadas para que possam ser importados em outros scripts.

O funcionamento é semelhante a importar bibliotecas.

# Módulos de funções

Nesses casos, é útil saber quando um módulo está sendo importado e quando está sendo executado como script e definir comportamentos diferentes dependendo do caso.

# Módulos de funções

```
def main():  
    print("Meu módulo é o máximo.")  
if __name__ == '__main__':  
    main()
```

# Bibliotecas

# Especificar é programar

# Pensamento computacional

- Decomposição
- Reconhecimento de padrões
- Abstração
- Algoritmo

# Especificação

Ao especificar, é necessário decompor e abstrair a sua ideia, reconhecer os padrões para por fim poder determinar o algoritmo e de fato desenvolver a aplicação.



# Especificação

Isso pode ser realizado formalmente através de um documento de especificação de requisitos.

Para problemas pequenos, a especificação é simples e não precisa ser formalizada, mas sempre existe.

# Especificação e planejamento

# Especificação e planejamento

A especificação vai permitir planejar o curso do projeto pois será possível estimar o tempo para a implementação de cada funcionalidade especificada.

# Metodologias de desenvolvimento



INSTITUTO BRASILEIRO DE ENSINO,  
DESENVOLVIMENTO E PESQUISA