

Profa. Dra. Raquel C. de Melo-Minardi  
Departamento de Ciência da Computação  
Instituto de Ciências Exatas  
Universidade Federal de Minas Gerais

---



# MÓDULO 2 – PROGRAMAÇÃO

## Variáveis

**Variáveis** representam um espaço de memória do computador para armazenar determinado dado

Normalmente, o compilador / interpretador faz para o programador a **associação nome-valor** de forma que os nomes das variáveis localizam a sua posição na memória em que um determinado (ou diversos) tipo(s) de dado(s) podem ser armazenados

**Waldemar Celes et al.**

## DECLARAÇÃO

- ▶ Em **linguagens tipadas**, como C, todas as variáveis precisam ser explicitamente declaradas antes de seu uso
- ▶ A declaração consiste em **especificar o nome e tipo** de uma variável
  - ▶ O **nome** serve para associar o dado armazenado ao espaço de memória da variável
  - ▶ O **tipo** determina a natureza do dado que será armazenado
- ▶ Uma variável só pode receber valores do tipo de dados previamente acordado na declaração (domínio)
  - ▶ Uma variável do tipo inteiro pode unicamente armazenar números inteiro

# DECLARAÇÃO

- ▶ Em C, existem alguns **tipos básicos**
  - ▶ *char* (caracter), *short* (inteiro pequeno), *int* (inteiro), *long* (inteiro grande)
- ▶ Esses tipos diferem
  - ▶ pelo espaço de memória que ocupa
  - ▶ pelo intervalo de valores que conseguem representar
    - ▶ Um número *char* é representado em *1 Byte* (*8 bits*) e pode representar  $2^8 = 256$  valores distintos
    - ▶ Um *short* é representado por *2 Bytes* e *long* por *4 Bytes*

## DECLARAÇÃO

- ▶ Em Python (e Perl), as coisas funcionam de forma bem diferente
- ▶ **Não é preciso declarar variáveis** antes do uso e nem definir o tipo de dados que será associado às mesmas
- ▶ Nomes de variáveis podem surgir ao longo do código e podem assumir dinamicamente diversos tipos
- ▶ Há portanto implicações (as boas e as nem tão boas)
  - ▶ Facilidade de programação
  - ▶ Impossibilidade de validação dos dados pode gerar problemas em tempo de execução
- ▶ O excesso de **flexibilidade** e indulgência da linguagem traz uma grande **responsabilidade** ao programador que pode cometer erros inadvertidamente



# TIPOS DE VARIÁVEIS

- ▶ Pode-se categorizar variáveis também em termos da organização da estrutura de dados envolvida. Vamos definir então esses tipos
- ▶ **Variáveis simples ou atômicas:** associadas a dados simples compostos por apenas um elemento como um número ou um caracter. Exemplo: um aminoácido
- ▶ **Variáveis compostas:** definidas por um nome associado a diversos elementos  
Exemplo: a sequência de aminoácidos de uma proteína
- ▶ **Variáveis homogêneas:** são variáveis compostas por elementos de um mesmo tipo de dados. Exemplo: a mesma sequência de aminoácidos é uma variável composta de inúmeros caracteres, ou seja, elementos do mesmo tipo
- ▶ **Variáveis heterogêneas:** armazenam conjuntos de dados formados por tipos diferentes (campos do registro) em uma mesma estrutura. Exemplo: uma variável proteína que armazene os campos id (*string*), tamanho (*int*), massa (*float*), sequência (*string*)

## TIPOS DE VARIÁVEIS EM PYTHON

- ▶ Em Python, as variáveis não seguem essas definições de forma rígida
  - ▶ Há variáveis **simples** e **compostas**
  - ▶ **Não há como garantir que uma variável composta seja homogênea**, ou seja, a princípio, todas as variáveis compostas são heterogêneas
- ▶ A diante, vamos descrever em detalhes aos tipos de variáveis em Python

# TIPOS DE DADOS EM PYTHON

- ▶ Tudo em Python são **objetos**
- ▶ Como dissemos anteriormente, uma variável é uma **associação nome-valor**
  - ▶ Em Python, os **tipos são dinâmicos**, logo podem mudar ao longo da execução
- ▶ Python tem três tipos de dados básicos
  - ▶ inteiro
  - ▶ ponto flutuante
  - ▶ booleano
  - ▶ complexo



## NOMES DE VARIÁVEIS

- ▶ Não se pode usar **palavras reservadas** como identificadores para dar nomes a variáveis:
- ▶ *false, await, else, import, pass, none, break, except, in, raise, true, class, finally, is, return, and, continue, for, lambda, try, as, def, from, nonlocal, while, assert, del, global, not, with, async, elif, if, or, yield*

# TIPOS DE VARIÁVEIS OU ESTRUTURAS DE DADOS EM PYTHON

- ▶ Com relação às **variáveis compostas**, Python apresenta:
  - ▶ **sequências**: são objetos ordenados e finitos
  - ▶ **dicionários**: é um conjunto de elementos indexados por chaves (que podem ser sequências de caracteres)

# TIPOS DE VARIÁVEIS OU ESTRUTURAS DE DADOS EM PYTHON

- ▶ As **sequências** podem ser de dois tipos:
  - ▶ **sequências imutáveis**: são objetos ordenados e finitos
    - ▶ **strings**: cadeias de caracteres
    - ▶ **tuples**: dois ou mais elementos de qualquer tipo dentro de parênteses e separados por vírgula
  - ▶ **sequências mutáveis**
    - ▶ **lists**: conhecidas em outras linguagens como vetores ou arranjos
    - ▶ **sets**: coleções não ordenadas e não indexadas escritas entre chaves
    - ▶ **dictionaries**: conjuntos de elementos indexados por chaves (que podem ser sequências de caracteres)