

# LÓGICA DE PROGRAMAÇÃO

## Aula 2



# AGENDA

## Variáveis, Tipos de Dados e Operadores

- Variáveis, constantes e tipos de dados
- Regras de nomenclatura para variáveis
- Declaração, atribuição e inicialização de variáveis
- Classificação das linguagens de programação

# Variáveis, Constantes e Tipos de Dados

Variáveis e constantes são os elementos básicos que um algoritmo manipula. Uma variável/constante é um espaço reservado na memória do computador para armazenar um valor/dado. Elas devem receber nomes para poderem ser referenciadas e modificadas quando necessário.

## VARIÁVEL

Dado cujo valor se modifica ao longo da execução do algoritmo.

## CONSTANTE

Dado cujo valor permanece o mesmo do início ao fim do algoritmo.

## TIPOS DE DADOS

- **Numéricos:** armazenam números Inteiros (sem casas decimais) ou Reais (com casas decimais).
- **Alfanuméricos:** armazenam sequências de caracteres, ou seja, textos.
- **Lógicas/Booleanas:** podem assumir apenas valores lógicos, isto é, Verdadeiro ou Falso.



# Regras de nomenclatura para variáveis

## PROIBIDO

- Acentuação
- Espaço
- Caracteres especiais
- Números na primeira posição

## PERMITIDO

- Letras maiúsculas e minúsculas
- Números após a primeira posição

## CONVENÇÕES DE NOMENCLATURA

- **Camel Case:** primeira letra minúscula e a inicial de cada nova palavra maiúscula ( *valorFinal* )
- **Pascal Case:** todas as palavras com a inicial maiúscula ( *ValorFinal* )
- **Snake Case:** *underline* no lugar do espaço para separar as palavras ( *valor\_final* )
- **Upper Case:** todas as letras maiúsculas ( *VALORFINAL* )
- **Lower Case:** todas as letras minúsculas ( *valorfinal* )

Verifique as convenções utilizadas pela linguagem de programação escolhida!

# Declaração, atribuição e inicialização

**DECLARAÇÃO:** reservar uma variável (posição na memória) para ser ocupada posteriormente.

*var nome*

*let x*

*const pi*

**ATRIBUIÇÃO:** guardar um valor em uma variável já existente.

*nome = "Ana"*

*x = 10*

*pi = 3.14*

**INICIALIZAÇÃO:** realizar na mesma instrução a declaração e a atribuição da variável.

*var nome = "Pedro"*

*let x = 20*

*const pi = 3.14*

A linguagem Python aceita somente a inicialização de variáveis, não sendo possível declarar uma variável sem atribuir a ela um valor.

# Classificação das Linguagens de Programação

## Baixo Nível x Alto Nível

Quanto mais próximo da linguagem de máquina (bits e bytes), mais baixo é o nível da linguagem. O contrário também é verdadeiro: quanto mais próximo da linguagem humana, mais alto é o nível da linguagem. Linguagens de baixo nível são mais rápidas na comunicação direta com o hardware, enquanto linguagens de alto nível são mais adequadas para a web.

## Tipagem Forte x Tipagem Fraca

Em linguagens fortemente tipadas é necessário determinar o tipo do dado a ser armazenado na variável. Já nas linguagens fracamente tipadas, basta declarar a variável usando uma palavra reservada, que o tipo será inferido pela própria linguagem. Isso é feito no momento da declaração da variável. A tipagem forte é mais adequada para aplicações backend para evitar erros de conversão de tipo. Por sua vez, a tipagem fraca pode ser usada com maior segurança no frontend.

## Tipagem Estática x Tipagem Dinâmica

Linguagens com tipagem estática não permitem que o tipo da variável seja alterado depois de ter sido determinado. Já a tipagem dinâmica é o comportamento oposto: a cada vez que um novo valor for atribuído à variável, seu tipo será modificado de acordo com o valor inserido.

# Classificação das Linguagens de Programação

## Compilada x Interpretada

Quando o código é traduzido em linguagem de máquina linha após linha, estamos falando de uma linguagem interpretada. Já quando ele é transformado em bits por um processo chamado build e lido todo de uma vez, trata-se de uma linguagem compilada. As linguagens interpretadas são mais utilizadas na web, pois os navegadores são interpretadores de código. Já as linguagens compiladas são mais utilizadas em sistemas operacionais e firmwares, no que chamamos de programação embarcada.

## Estruturada x Orientada a Objetos

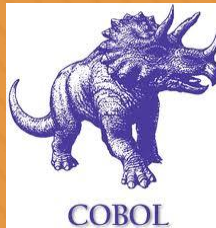
As linguagens que trazem suporte à Programação Estruturada são chamadas de linguagens estruturadas. Já as linguagens orientadas a objetos, como o próprio nome sugere, trazem suporte ao paradigma da POO. Linguagens estruturadas eram mais comuns nos primórdios da programação, especialmente nas décadas de 1950 até 1980, pois nesta época os computadores ainda não funcionavam em rede e não realizavam processamento paralelo. Então a programação estruturada, onde cada comando é executado após o anterior, funcionava muito bem. Com o surgimento da Internet e da Web, no entanto, o paradigma da Orientação a Objetos se tornou mais adequado para representar as interações entre componentes do mundo real. Por isso as linguagens modernas já oferecem suporte à POO.



# LINHA DO TEMPO

ANOS 1950

DIAS ATUAIS



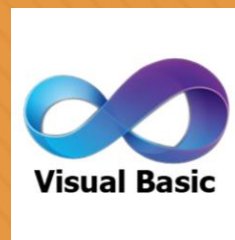


INTERPRETADA



BAIXO NÍVEL

ALTO NÍVEL



COMPILADA

DINÂMICA



FORTE

FRACA



ESTÁTICA

# ORIENTAÇÃO A OBJETOS

NÃO



SIM





danielle.carvalho@go.senac.br

Serviço Nacional de  
Aprendizagem  
Comercial - Goiás

go.senac.br  
senacgoias  
senacgoias

