

## Aula 2

### Professores:

Dante Corbucci Filho

Leandro A. F. Fernandes

### Conteúdo:

- Variáveis
- Tipos
- Comandos Básicos

## Variáveis

- Área de memória que mantém um valor, que pode ser mudado.
- Identificador da variável: é o nome da variável, representada por uma sequência de caracteres, iniciada por uma letra minúscula.
  - Exemplos: nota, saldo, deposito, saque, casa13Buzios.

## Variáveis

- Área de memória que mantém um valor, que pode ser mudado.
- Identificador da variável: é o nome da variável, representada por uma sequência de caracteres, iniciada por uma letra minúscula.
  - Exemplos: nota, saldo, deposito, saque, casa13Buzios.
- Nossa padronização: não deve ter acento e deve possuir apenas caracteres alfa-numéricos. Usar notação de camelo
  - Exemplos: minhaNota, notaTurma, cartoesAmarelos, cartoesVermelhos.
  - Isto é, sempre que for uma composição de várias palavras, iniciar a próxima com letra maiúscula e as demais letras sempre deverão ser minúsculas.

## Tipos Básicos (Embutidos)

- Tipos básicos são imutáveis

## Tipos Básicos (Embutidos)

- Tipos básicos são imutáveis
- Tipos Integrais
  - Inteiro: (**int**)
    - Pode ter centenas de dígitos, limitado apenas pela memória do computador;
    - O padrão é decimal, mas pode-se usar outras bases como binária (iniciada com **0b**), octal (iniciada com **0o**) ou hexadecimal (iniciada com **0x**).
  - Lógico (ou Booleano): (**bool**)
    - 0 é **False** e 1 é **True**.

## Tipos Básicos (Embutidos)

- Tipos básicos são imutáveis
- Tipos Integrais
  - Inteiro: (**int**)
    - Pode ter centenas de dígitos, limitado apenas pela memória do computador;
    - O padrão é decimal, mas pode-se usar outras bases como binária (iniciada com **0b**), octal (iniciada com **0o**) ou hexadecimal (iniciada com **0x**).
  - Lógico (ou Booleano): (**bool**)
    - 0 é **False** e 1 é **True**.
- Tipos de Ponto-Flutuante
  - Número de Ponto-Flutuante: (**float**)
  - Número Complexo: (**complex**)
    - Representado por um par de números de ponto-flutuante.

## Tipos Básicos (Embutidos)

- Tipos básicos são imutáveis
- Tipos Integrais
  - Inteiro: (**int**)
    - Pode ter centenas de dígitos, limitado apenas pela memória do computador;
    - O padrão é decimal, mas pode-se usar outras bases como binária (iniciada com **0b**), octal (iniciada com **0o**) ou hexadecimal (iniciada com **0x**).
  - Lógico (ou Booleano): (**bool**)
    - 0 é **False** e 1 é **True**.
- Tipos de Ponto-Flutuante
  - Número de Ponto-Flutuante: (**float**)
  - Número Complexo: (**complex**)
    - Representado por um par de números de ponto-flutuante.
- String (**str**)
  - Representada por uma sequência de caracteres Unicode, iniciada e terminada por aspas simples ou duplas.

## Conversão de Tipos

- String, booleano ou ponto-flutuante para inteiro: **int**
- String, booleano ou inteiro para ponto-flutuante: **float**
- String, inteiro ou ponto-flutuante para booleano: **bool**
- Booleano, inteiro ou ponto-flutuante para String: **str**

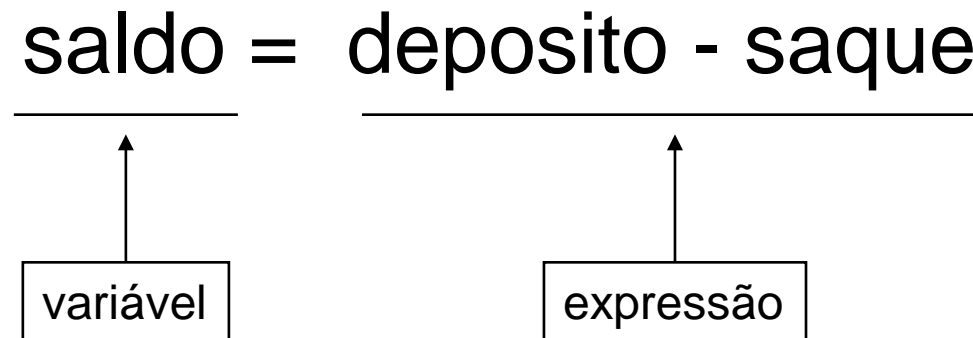


## Comando de Atribuição

- Comando mais importante de uma linguagem imperativa.

## Comando de Atribuição

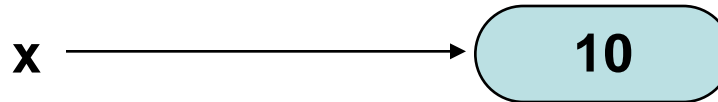
- Comando mais importante de uma linguagem imperativa.
- Em um comando de atribuição, uma variável recebe o resultado da avaliação de uma expressão.



## Linguagem de Tipagem Dinâmica

Referência para

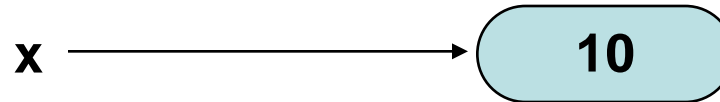
- $x = 10$



## Linguagem de Tipagem Dinâmica

Referência para

- $x = 10$



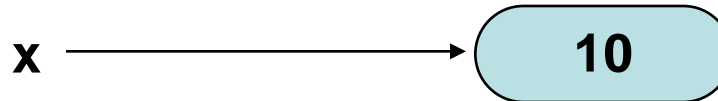
- $x = \text{"Maria"}$



## Linguagem de Tipagem Dinâmica

Referência para

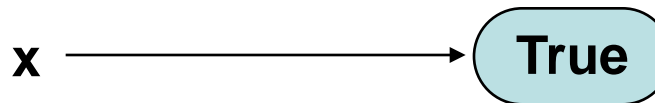
- `x = 10`



- `x = "Maria"`



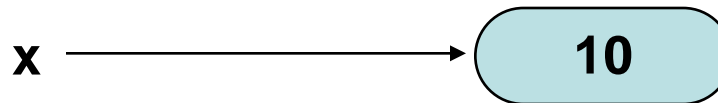
- `x = True`



## Linguagem de Tipagem Dinâmica

Referência para

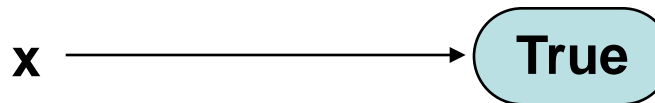
- $x = 10$



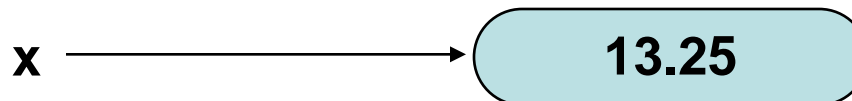
- $x = \text{"Maria"}$



- $x = \text{True}$



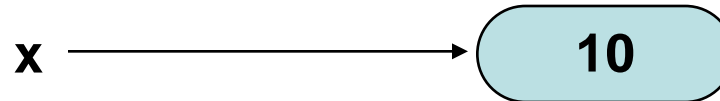
- $x = 13.25$



## Linguagem de Tipagem Dinâmica

Referência para

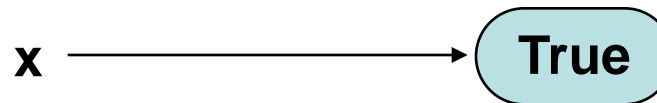
- $x = 10$



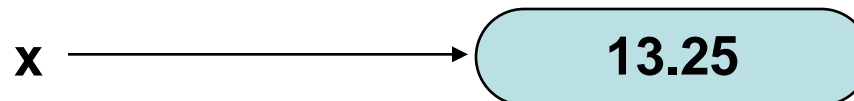
- $x = \text{"Maria"}$



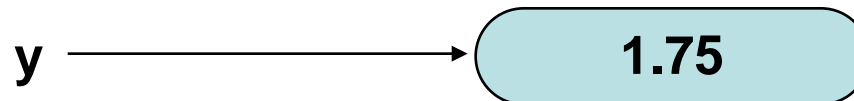
- $x = \text{True}$



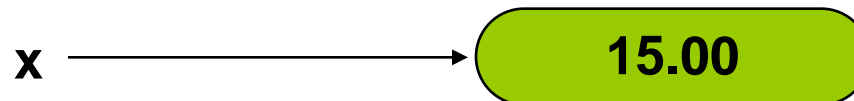
- $x = 13.25$



- $y = 1.75$

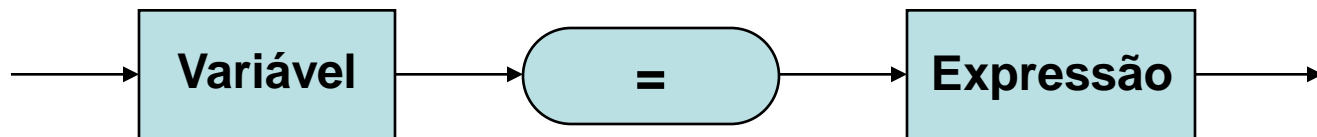


- $x = x + y$



## Diagramas Sintáticos

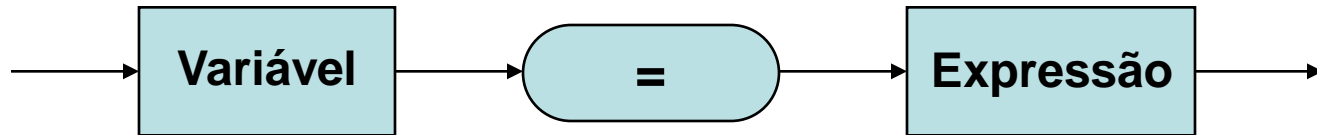
- Atribuição Simples:



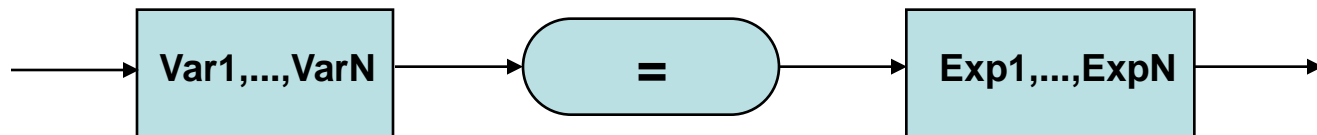


## Diagramas Sintáticos

- Atribuição Simples:

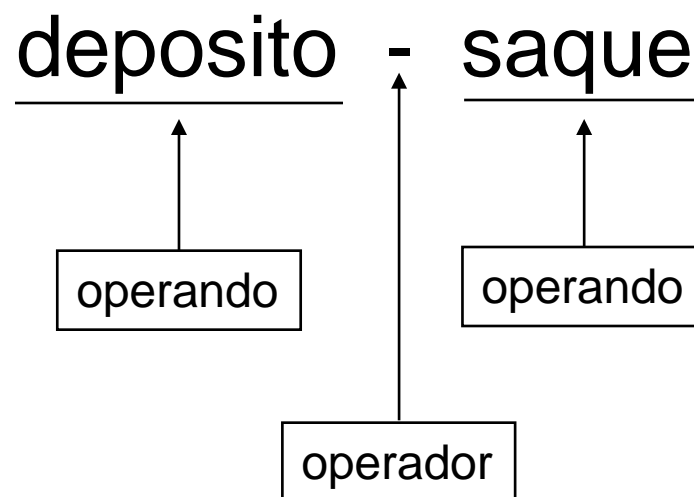


- Atribuição Múltipla:



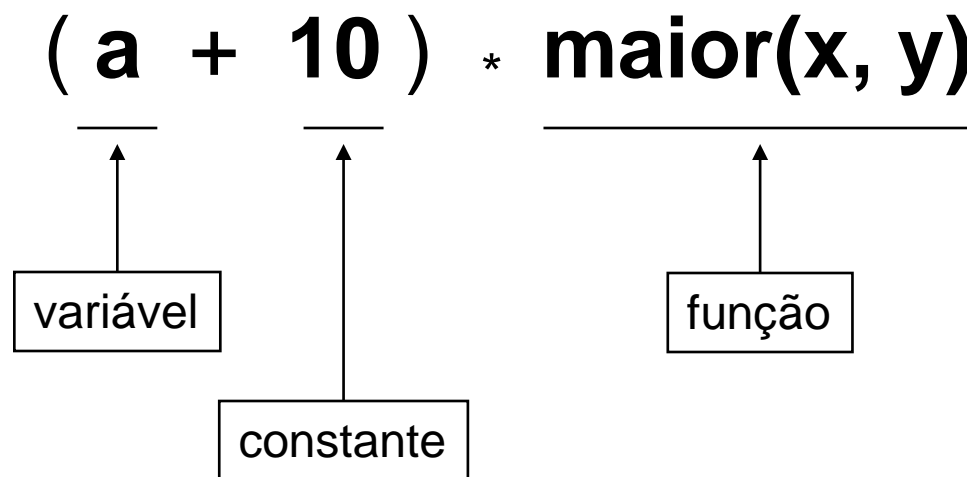
## Expressão

- Uma expressão especifica o cálculo de um valor.
- É definida por operando(s) e operador(es).



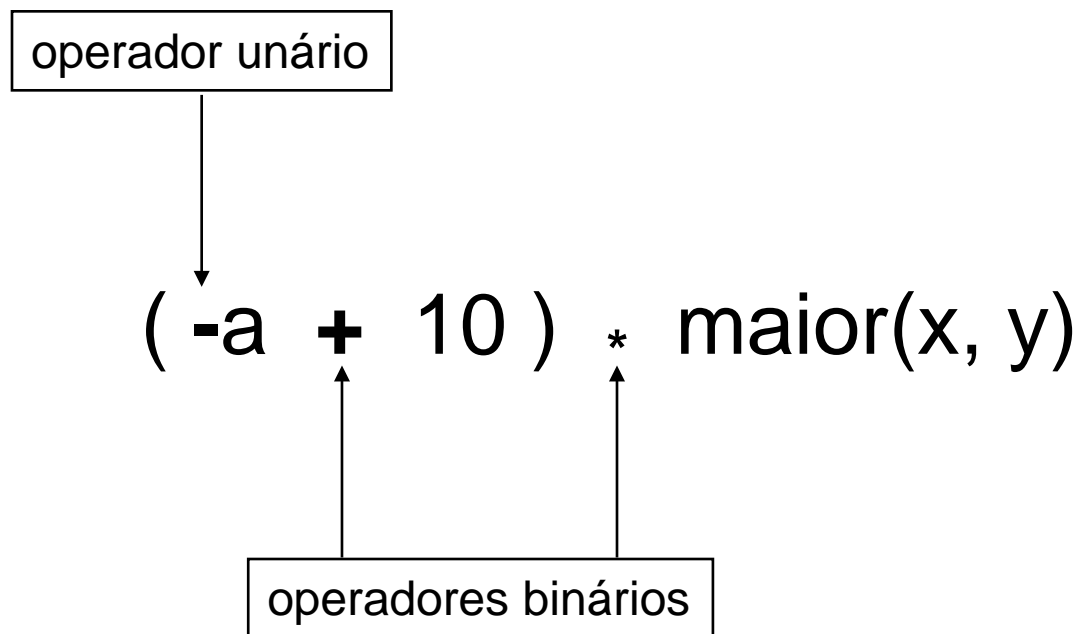
## Operando

- Operando pode ser uma constante, uma variável ou um resultado de função.



## Operador

- Operador pode ser unário ou binário, dependendo se admite um operando ou dois operandos, respectivamente.



## Operadores Unários

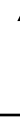
- Operador numérico positivo (+)
  - Operando deve ser numérico
- Operador numérico negativo (-)
  - Operando deve ser numérico

**+5.75**



constante real

**-a**



variável numérica

## Operadores Unários

- Operador numérico positivo (+)
  - Operando deve ser numérico
- Operador numérico negativo (-)
  - Operando deve ser numérico
- Operador lógico negação (**not**)
  - Operando deve ser booleano

**+5.75**



constante real

**-a**



variável numérica

**not fim**



variável booleana

22

## Operadores Binários Aritméticos

- Soma (+)
  - Subtração (-)
- } operadores  
aditivos

## Operadores Binários Aritméticos

- Soma (+)
  - Subtração (-)
- } operadores aditivos
- 
- Produto (\*)
  - Divisão de ponto flutuante (/)
  - Divisão inteira (//)
  - Resto da divisão inteira (%)
- } operadores multiplicativos



## Operadores Binários Aritméticos

- Soma (+)
  - Subtração (-)
- } operadores aditivos
- 
- Produto (\*)
  - Divisão de ponto flutuante (/)
  - Divisão inteira (//)
  - Resto da divisão inteira (%)
- } operadores multiplicativos
- 
- Potenciação (\*\*)

## Operadores Binários Aritméticos

- Soma (+)
  - Subtração (-)
- } operadores aditivos
- 
- Produto (\*)
  - Divisão de ponto flutuante (/)
  - Divisão inteira (//)
  - Resto da divisão inteira (%)
- } operadores multiplicativos
- 
- Potenciação (\*\*)

$$( 5.75 + ( a \% b ) - 7 ) / 8.1$$

## Operadores Binários Lógicos

- Disjunção lógica ou soma lógica (**or**)

## Operadores Binários Lógicos

- Disjunção lógica ou soma lógica (**or**)
- Conjunção lógica ou produto lógico (**and**)

## Operadores Binários Lógicos

- Disjunção lógica ou soma lógica (**or**)
- Conjunção lógica ou produto lógico (**and**)

**( p or q ) and r**

## Operadores Binários Relacionais

- Igual a (==)
- Diferente de (!=)

O resultado de uma operação relacional é um valor booleano.

## Operadores Binários Relacionais

- Igual a (==)
- Diferente de (!=)

O resultado de uma operação relacional é um valor booleano.

$(2 + 2) == 5$  é falso!

$a == 5$

$x != y$

## Operadores Binários Relacionais

- Igual a (==)
- Diferente de (!=)
- Maior que (>)
- Menor que (<)
- Maior ou igual a (>=)
- Menor ou igual a (<=)

O resultado de uma operação relacional é um valor booleano.

$(2 + 2) == 5$	é falso!
$(2 + 2) <= 5$	é verdadeiro!

$a == 5$

$x != y$

$a <= 6 + c$



## Precedência dos Operadores

1. Expressões entre Parênteses ( ) – Maior Prioridade;
2. Potenciação (\*\*);
3. Unários (+, -);
4. Binários Multiplicativos (\*, /, %, //);
5. Binários Aditivos (+, -);
6. Relacionais (==, !=, <, >, <=, >=);
7. Lógico **not**;
8. Lógico **and**;
9. Lógico **or** – Menor Prioridade.

## Precedência dos Operadores

1. Expressões entre Parênteses ( ) – Maior Prioridade;
2. Potenciação (\*\*);
3. Unários (+, -);
4. Binários Multiplicativos (\*, /, %, //);
5. Binários Aditivos (+, -);
6. Relacionais (==, !=, <, >, <=, >=);
7. Lógico **not**;
8. Lógico **and**;
9. Lógico **or** – Menor Prioridade.

$5.75 + a\%b - 7/8.1$  equivale a  $5.75 + (a\%b) - (7/8.1)$

34

## Precedência dos Operadores

$5 * a \% b / 8.1$

equivale a

$((5 * a) \% b) / 8.1$

## Comandos de Saída Padrão

- **print()**
  - Pula para a próxima linha na saída padrão (vídeo).

## Comandos de Saída Padrão

- **print()**
  - Pula para a próxima linha na saída padrão (vídeo).
- **print(*expressão*)**
  - Escreve na saída padrão (vídeo) o resultado da avaliação da *expressão*;
  - Ao final, pula para a próxima linha.

## Comandos de Saída Padrão

## Comandos de Saída Padrão

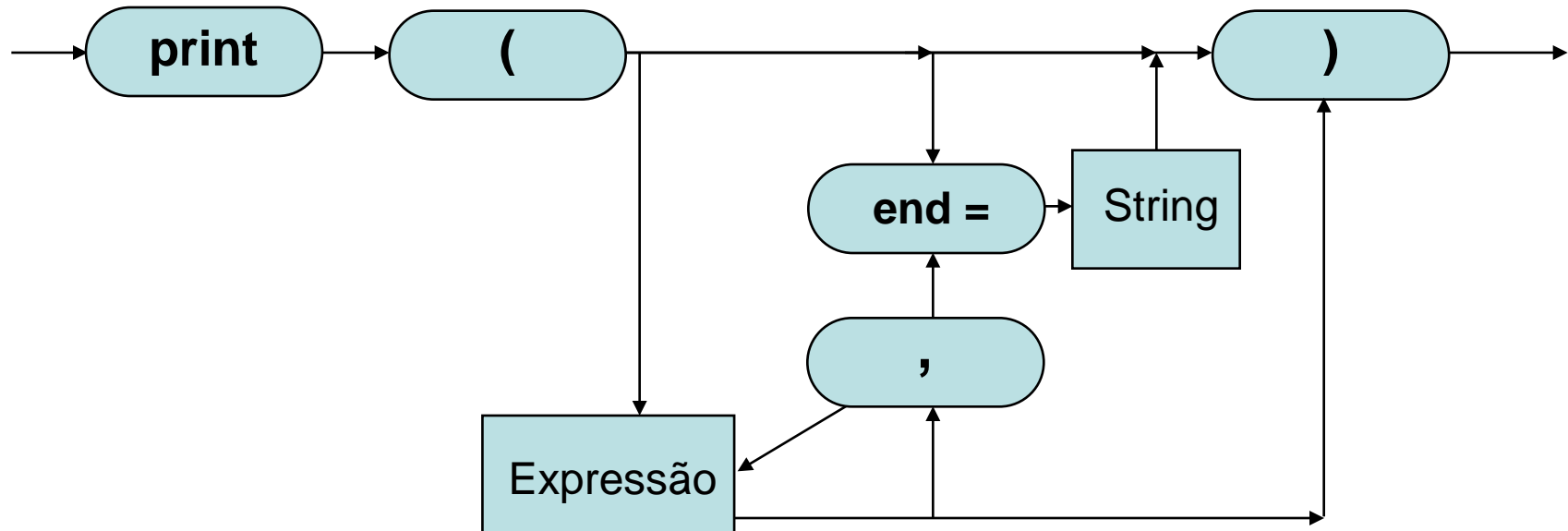
- **print(*expressão*, end = *término*)**
  - Escreve na saída padrão (vídeo) o resultado da avaliação da *expressão*;
  - Ao final, escreve a String de *término*.

## Comandos de Saída Padrão

- **print(*expressão*, end = *término*)**
  - Escreve na saída padrão (vídeo) o resultado da avaliação da *expressão*;
  - Ao final, escreve a String de *término*.
- **print(*exp1*, *exp2*, ..., *expN*, end = *término*)**
  - Escreve na saída padrão (vídeo) o resultado da avaliação de cada expressão *expNum*;
  - Um espaço em branco é escrito entre cada par de *expNum*;
  - Ao final, escreve a String de *término*.



## Diagrama Sintático do Comando *print*



## Expressões Formatadas (operador %(...))

- Sintaxe da expressão formatada:

*texto* = "...%formato1 ... %formatoN..." %(exp1, ..., expN)

- Formatos mais comuns:

**d** (inteiro), **f** (número com ponto flutuante) e **s** (String)

## Expressões Formatadas (operador %(...))

- Sintaxe da expressão formatada:

*texto* = "...%formato1 ... %formatoN..." %(exp1, ..., expN)

- Formatos mais comuns:

**d** (inteiro), **f** (número com ponto flutuante) e **s** (String)

- Exemplos:

–msg = "A média dos números é %4.2f" % (82432.923421)  
**print(msg)**

- Escreve "A média dos números é 82432.92" e pula de linha

## Expressões Formatadas (operador %(...))

- Sintaxe da expressão formatada:

*texto* = "...%formato1 ... %formatoN..." %(exp1, ..., expN)

- Formatos mais comuns:

**d** (inteiro), **f** (número com ponto flutuante) e **s** (String)

- Exemplos:

–msg = "A média dos números é %4.2f" % (82432.923421)  
**print(msg)**

- Escreve "A média dos números é 82432.92" e pula de linha

–**print("%f + %f = %4.1f" % (15,7.8313, 15+7.8313))**

- Escreve "15.000000 + 7.831300 = 22.8" e pula de linha

## Expressões Formatadas (operador %(...))

- Sintaxe da expressão formatada:

*texto* = "...%formato1 ... %formatoN..." %(exp1, ..., expN)

- Formatos mais comuns:

**d** (inteiro), **f** (número com ponto flutuante) e **s** (String)

- Exemplos:

–msg = "A média dos números é %4.2f" % (82432.923421)  
**print(msg)**

- Escreve "A média dos números é 82432.92" e pula de linha

–**print("%f + %f = %4.1f" % (15,7.8313, 15+7.8313))**

- Escreve "15.000000 + 7.831300 = 22.8" e pula de linha

–**print("%d + %d = %d" % (5.89, 7.83, 5.89+7.83), end = "!!!")**

- Escreve "5 + 7 = 13!!!" (apenas a parte inteira) e não pula de linha

## Comandos de Entrada Padrão

- **input()**
  - Comando que aguarda o usuário fornecer, pela entrada padrão (teclado), um valor expresso por uma sequência de caracteres, e o retorna.
  - Este comando tem o efeito de suspender a execução do programa até que o usuário escreva sua entrada e pressione a tecla <enter>.
  - Exemplo:

aluno = **input()**

## Comandos de Entrada Padrão

- **input()**

- Comando que aguarda o usuário fornecer, pela entrada padrão (teclado), um valor expresso por uma sequência de caracteres, e o retorna.
- Este comando tem o efeito de suspender a execução do programa até que o usuário escreva sua entrada e pressione a tecla <enter>.
- Exemplo:

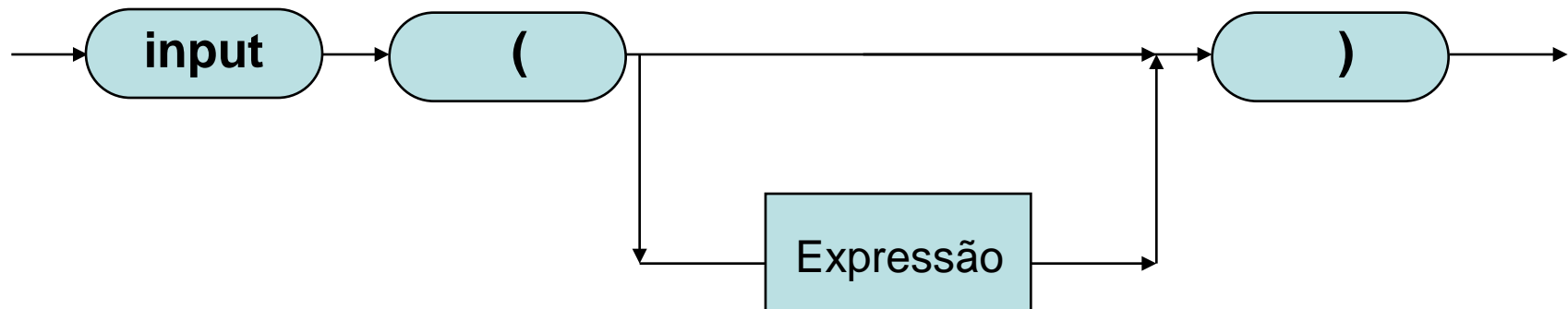
aluno = **input()**

- **input(*mensagem*)**

- Comando que escreve a expressão *mensagem* na saída padrão (vídeo) e aguarda, via interrupção, que o usuário escreva sua resposta, composta de uma sequência de caracteres digitados, e pressione a tecla <enter>.
- Exemplo:

aluno = **input**("Digite o nome do aluno: ")

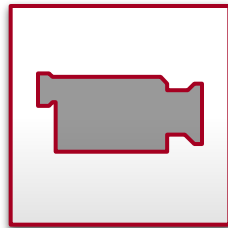
## Diagrama Sintático do Comando *input*





## Exemplos de Aplicação dos Conteúdos Vistos

Clique no botão para assistir ao tutorial:



## **Faça os Exercícios Relacionados a essa Aula**

Clique no botão para visualizar os enunciados:



## Aula 2

### Professores:

Dante Corbucci Filho  
Leandro A. F. Fernandes

### Conteúdo Apresentado:

- Variáveis
- Tipos
- Comandos Básicos