

Conceitos básicos



30 de agosto de 2022

João Paulo Ataíde Martins

<http://joaopauloam.qui.ufmg.br/>

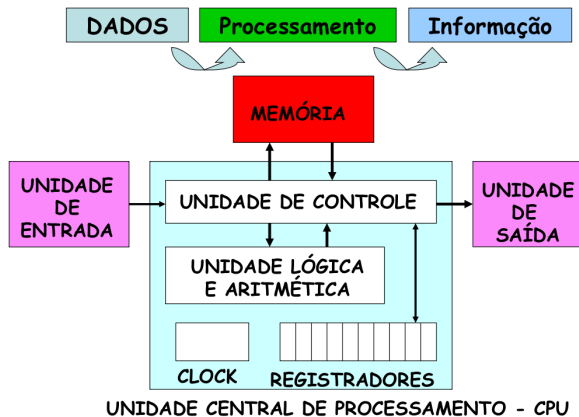
Departamento de Química / Instituto de Ciências Exatas – UFMG

INTRODUÇÃO

- Como duas pessoas podem realizar uma soma com 2 números sem calculadora?

- Como duas pessoas podem realizar uma soma com 2 números sem calculadora?
- Uma delas escolhe os números

- Como duas pessoas podem realizar uma soma com 2 números sem calculadora?
- Uma delas escolhe os números
- A outra realiza o cálculo



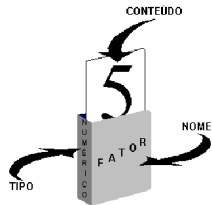
- Uma memória é uma sequência de células (ou posições)

- Uma memória é uma sequência de células (ou posições)
- Armário com várias gavetas, sendo que em cada gaveta é possível guardar apenas um único valor por vez, e essas gavetas deverão estar identificadas com uma etiqueta contendo um nome de identificação

VARIÁVEIS

- É um espaço da memória do computador reservado para guardar dados (valores) dos programas.

- É um espaço da memória do computador reservado para guardar dados (valores) dos programas.



- Caixa com um rótulo (nome) colado nela, que em um dado momento guarda um determinado objeto. O conteúdo desta caixa não é algo fixo, permanente.

- . Atributos de uma variável:

- Atributos de uma variável:
 - Nome ou identificador

- Atributos de uma variável:
 - Nome ou identificador
 - Endereço

- Atributos de uma variável:
 - Nome ou identificador
 - Endereço
 - Tipo de dados

- Atributos de uma variável:
 - Nome ou identificador
 - Endereço
 - Tipo de dados
 - Conteúdo

- Sequência de caracteres utilizada para identificar a variável

- Sequência de caracteres utilizada para identificar a variável
- Regras para a formação dos nomes na linguagem Python:

- Sequência de caracteres utilizada para identificar a variável
- Regras para a formação dos nomes na linguagem Python:
 - Primeiro caractere deve ser uma letra (a-z ou A-Z) ou o caractere sublinhado ()

- Sequência de caracteres utilizada para identificar a variável
- Regras para a formação dos nomes na linguagem Python:
 - Primeiro caractere deve ser uma letra (a-z ou A-Z) ou o caractere sublinhado (`_`)
 - Os demais caracteres podem ser letras (a-z ou A-Z), números (0-9) ou sublinhado (`_`)

- Sequência de caracteres utilizada para identificar a variável
- Regras para a formação dos nomes na linguagem Python:
 - Primeiro caractere deve ser uma letra (a-z ou A-Z) ou o caractere sublinhado (`_`)
 - Os demais caracteres podem ser letras (a-z ou A-Z), números (0-9) ou sublinhado (`_`)
 - Não são permitidos espaços em branco e caracteres especiais (`@`, `$`, `+`, `-`, `%`, `!`)

- Exemplos:

- Exemplos:
 - Nomes válidos:

- Exemplos:
 - Nomes válidos:
 - valor

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:
 - 2prova

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:
 - 2prova
 - preço

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:
 - 2prova
 - preço
 - salário

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:
 - 2prova
 - preço
 - salário
 - ano atual

- Exemplos:
 - Nomes válidos:
 - valor
 - media_aritmetica
 - nota1
 - FGTS
 - __pessoa
 - Nomes inválidos:
 - 2prova
 - preço
 - salário
 - ano atual
 - valor\$

- Regras Gerais

- Regras Gerais
 - Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo

- Regras Gerais
 - Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`

- Regras Gerais
 - Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`
 - `x`, `y`, `z`, `a`, `b`, `c`

- Regras Gerais
 - Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`
 - `x`, `y`, `z`, `a`, `b`, `c`
 - Algumas LPs fazem diferenciação entre maiúsculas e minúsculas no nome (C, Java, Python)

- Regras Gerais

- Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`
 - `x`, `y`, `z`, `a`, `b`, `c`
- Algumas LPs fazem diferenciação entre maiúsculas e minúsculas no nome (C, Java, Python)
 - `Peso` \neq `peso` \neq `PESO`

- Regras Gerais

- Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`
 - `x`, `y`, `z`, `a`, `b`, `c`
- Algumas LPs fazem diferenciação entre maiúsculas e minúsculas no nome (C, Java, Python)
 - `Peso` \neq `peso` \neq `PESO`
- Outras não fazem diferenciação como Pascal

- Regras Gerais

- Devem ser claros e precisos, dando uma ideia do “papel” da variável no algoritmo
 - `media_aritmetica`, `nota1`, `peso`, `altura`
 - `x`, `y`, `z`, `a`, `b`, `c`
- Algumas LPs fazem diferenciação entre maiúsculas e minúsculas no nome (C, Java, Python)
 - `Peso` \neq `peso` \neq `PESO`
- Outras não fazem diferenciação como Pascal
 - `Peso` = `peso` = `PESO`

- Assinale com um X os nomes de variáveis válidos em Python.

☐ abc ☐ 3abc ☐ a
☐ 123a ☐ -a ☐ acd1
☐ -_ad ☐ A&a ☐ guarda-chuva
☐ A123 ☐ Aa ☐ guarda_chuva
☐ ABC DE ☐ etc. ☐ b316
☐ int ☐ float ☐ str

- A memória é uma sequência de células endereçáveis

- A memória é uma sequência de células endereçáveis
- Cada variável irá ocupar uma ou mais células da memória e, portanto, terá o seu próprio endereço de memória.

TIPOS DE DADOS

- O que são tipos de dados?

- O que são tipos de dados?
 - Conjunto de Valores + Operações

- O que são tipos de dados?
 - Conjunto de Valores + Operações
- Exemplo:

- O que são tipos de dados?
 - Conjunto de Valores + Operações
- Exemplo:
 - Tipo de Dados Inteiro

- O que são tipos de dados?
 - Conjunto de Valores + Operações
- Exemplo:
 - Tipo de Dados Inteiro
 - Números Inteiros + Operações Aritméticas

- Os tipos de dados mais simples da linguagem Python são:

- Os tipos de dados mais simples da linguagem Python são:
 - Numéricos

- Os tipos de dados mais simples da linguagem Python são:
 - Numéricos
 - Inteiros \Rightarrow int

- Os tipos de dados mais simples da linguagem Python são:
 - Numéricos
 - Inteiros \Rightarrow int
 - Reais \Rightarrow float

- Os tipos de dados mais simples da linguagem Python são:
 - Numéricos
 - Inteiros \Rightarrow int
 - Reais \Rightarrow float
 - Cadeia de caracteres \Rightarrow str

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal
 - Exemplos: -23 98 0 1350 -357 237

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal
 - Exemplos: -23 98 0 1350 -357 237
 - Exemplos de proposições:

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal
 - Exemplos: -23 98 0 1350 -357 237
 - Exemplos de proposições:
 - Ele tem 15 irmãos.

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal
 - Exemplos: -23 98 0 1350 -357 237
 - Exemplos de proposições:
 - Ele tem 15 irmãos.
 - A escada possui 8 degraus.

- Números inteiros podem ser positivos ou negativos e não possuem parte decimal
 - Exemplos: -23 98 0 1350 -357 237
 - Exemplos de proposições:
 - Ele tem 15 irmãos.
 - A escada possui 8 degraus.
 - Meu vizinho comprou 2 carros novos.

- Números reais podem ser positivos ou negativos e possuem parte decimal

- Números reais podem ser positivos ou negativos e possuem parte decimal
 - Exemplos: 23.45 346.89 -34.88 0.0

- Números reais podem ser positivos ou negativos e possuem parte decimal
 - Exemplos: 23.45 346.89 -34.88 0.0
 - Exemplos de proposições:

- Números reais podem ser positivos ou negativos e possuem parte decimal
 - Exemplos: 23.45 346.89 -34.88 0.0
 - Exemplos de proposições:
 - Ela tem 1,73 metros

- Números reais podem ser positivos ou negativos e possuem parte decimal
 - Exemplos: 23.45 346.89 -34.88 0.0
 - Exemplos de proposições:
 - Ela tem 1,73 metros
 - Meu saldo bancário é de 215,20.

- Números reais podem ser positivos ou negativos e possuem parte decimal
 - Exemplos: 23.45 346.89 -34.88 0.0
 - Exemplos de proposições:
 - Ela tem 1,73 metros
 - Meu saldo bancário é de 215,20.
 - No momento estou pesando 72,5 Kg.

- Representa uma cadeia formada por letras ou símbolos de texto

- Representa uma cadeia formada por letras ou símbolos de texto
- Indicado entre aspas simples (' ') ou aspas duplas (“ ”)

- Representa uma cadeia formada por letras ou símbolos de texto
- Indicado entre aspas simples (' ') ou aspas duplas (" ")
 - Exemplos: 'a', '@', 'G', ' ', 'abc', "123"

- Valor armazenado na variável

- Valor armazenado na variável
- Conteúdo de uma variável pode ser alterado, consultado ou apagado várias vezes

- Valor armazenado na variável
- Conteúdo de uma variável pode ser alterado, consultado ou apagado várias vezes
 - Informação anterior é perdida

- Valor armazenado na variável
- Conteúdo de uma variável pode ser alterado, consultado ou apagado várias vezes
 - Informação anterior é perdida



- A declaração de uma nova variável provoca:

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.
- Em Python essa declaração ocorre de maneira implícita, no momento em que a variável aparece pela primeira vez no programa

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.
- Em Python essa declaração ocorre de maneira implícita, no momento em que a variável aparece pela primeira vez no programa
 - `idade = 18` # variável do tipo inteiro

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.
- Em Python essa declaração ocorre de maneira implícita, no momento em que a variável aparece pela primeira vez no programa
 - idade = 18 # variável do tipo inteiro
 - altura = 1.72 # variável do tipo float

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.
- Em Python essa declaração ocorre de maneira implícita, no momento em que a variável aparece pela primeira vez no programa
 - idade = 18 # variável do tipo inteiro
 - altura = 1.72 # variável do tipo float
 - nome = "Rubens" # variável do tipo str

- A declaração de uma nova variável provoca:
 - A reserva de um espaço de memória suficiente para armazenar o tipo de dados declarado;
 - A associação do nome dado na declaração ao espaço de memória alocado.
- Em Python essa declaração ocorre de maneira implícita, no momento em que a variável aparece pela primeira vez no programa
 - idade = 18 # variável do tipo inteiro
 - altura = 1.72 # variável do tipo float
 - nome = "Rubens" # variável do tipo str

- Classifique os conteúdos das variáveis abaixo de acordo com seu tipo, assinalando com I os dados numéricos do tipo inteiro, com R os dados numéricos do tipo real e com S os dados do tipo string.

() 'laranja' () 5.7 () 1012
() -49 () +342 () -5.69
() 'L' () "casa" () 0.00001

COMANDO DE ATRIBUIÇÃO

- Comando de atribuição é utilizado para atribuir um valor a uma variável.

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:

(1) `num = 2`

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:

(1) num = 2

2

- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:

(1) num = 2

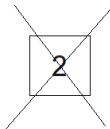
(2) num = num + 1

2

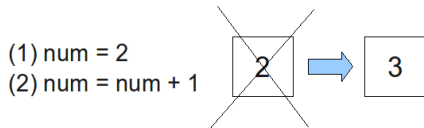
- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:

(1) `num = 2`

(2) `num = num + 1`



- Comando de atribuição é utilizado para atribuir um valor a uma variável.
- Operador de atribuição em Python $\Rightarrow =$
- Modo de Funcionamento:
 - Avalia a expressão situada no lado direito do comando de atribuição;
 - Armazena o valor resultante na variável situada do lado esquerdo do comando de atribuição.
 - Exemplo:



- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.

- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.
- Exemplo:

- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.
- Exemplo:
salario = 200.50

- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.
- Exemplo:
salario = 200.50 \Leftarrow salario nesse momento tem o tipo float;

- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.
- Exemplo:
salario = 200.50 \Leftarrow salario nesse momento tem o tipo float;
salario = 'a'

- É importante lembrar que quando se atribui um valor de tipo diferente daquele tipo definido para a variável inicialmente, a variável passa a ter o tipo definido na última atribuição.
- Exemplo:
salario = 200.50 \Leftarrow salario nesse momento tem o tipo float;
salario = 'a' \Leftarrow agora salario passa a ter o tipo str;

- Na atribuição simples em Python, sempre à esquerda do comando de atribuição deve haver um identificador de variável.

- Na atribuição simples em Python, sempre à esquerda do comando de atribuição deve haver um identificador de variável.
- Exemplos incorretos:

- Na atribuição simples em Python, sempre à esquerda do comando de atribuição deve haver um identificador de variável.
- Exemplos incorretos:
`2060 = numeroConta`

- Na atribuição simples em Python, sempre à esquerda do comando de atribuição deve haver um identificador de variável.
- Exemplos incorretos:
2060 = numeroConta
numeroAgencia + digitoControle = 2345

- No entanto, Python suporta atribuição múltipla com mais de um identificador à esquerda do operador de atribuição. Esses identificadores devem estar separados por vírgulas.

- No entanto, Python suporta atribuição múltipla com mais de um identificador à esquerda do operador de atribuição. Esses identificadores devem estar separados por vírgulas.
- Exemplos:

- No entanto, Python suporta atribuição múltipla com mais de um identificador à esquerda do operador de atribuição. Esses identificadores devem estar separados por vírgulas.
- Exemplos:
`a, b = 2`

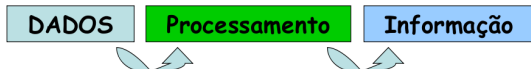
- No entanto, Python suporta atribuição múltipla com mais de um identificador à esquerda do operador de atribuição. Esses identificadores devem estar separados por vírgulas.
- Exemplos:
a, b = 2
x, y, z = 2, 3, 4

ENTRADA DE DADOS





- Nem todos os dados que um algoritmo manipula são gerados por ele.



- Nem todos os dados que um algoritmo manipula são gerados por ele.
 - Um algoritmo (programa) de caixa automático, por exemplo, tem que obter do usuário o número da conta, a senha, a opção de serviço desejada, etc.



- Nem todos os dados que um algoritmo manipula são gerados por ele.
 - Um algoritmo (programa) de caixa automático, por exemplo, tem que obter do usuário o número da conta, a senha, a opção de serviço desejada, etc.
- Portanto, deve haver um meio para que sejam digitados dados para o algoritmo.



- Nem todos os dados que um algoritmo manipula são gerados por ele.
 - Um algoritmo (programa) de caixa automático, por exemplo, tem que obter do usuário o número da conta, a senha, a opção de serviço desejada, etc.
- Portanto, deve haver um meio para que sejam digitados dados para o algoritmo.
- Todas as linguagens de programação têm comandos para permitir a entrada de dados.

- Comando input (Em Python):

- Comando input (Em Python):
 - É utilizado para comunicação entre o usuário e o programa;

- Comando input (Em Python):
 - É utilizado para comunicação entre o usuário e o programa;
 - Permite que o usuário forneça dados ao programa.

- Comando input (Em Python):
 - É utilizado para comunicação entre o usuário e o programa;
 - Permite que o usuário forneça dados ao programa.
 - Sintaxe:
`variavel = input("Mensagem opcional")`

- Comando input (Em Python):
 - É utilizado para comunicação entre o usuário e o programa;
 - Permite que o usuário forneça dados ao programa.
 - Sintaxe:
`variavel = input("Mensagem opcional")`
 - Armazena o dado digitado pelo usuário do programa na variável indicada.

- Exemplo em Python:
`nome = input("Qual o seu nome? ")`

- Exemplo em Python:
`nome = input("Qual o seu nome? ")`
- Armazena o dado digitado pelo usuário do programa na variável nome.

- Exemplo em Python:
`nome = input("Qual o seu nome? ")`
- Armazena o dado digitado pelo usuário do programa na variável nome.
- OBS: O resultado do comando input é sempre do tipo str. Para armazenar o resultado do comando input em uma variável de outro tipo deve-se fazer uma conversão de tipo.
- Exemplos
 - `idade = int(input("Qual a sua idade? "))`
 - `altura = float(input("Qual a sua altura? "))`

SAÍDA DE DADOS

- Programas precisam mostrar os resultados de seu processamento.

- Programas precisam mostrar os resultados de seu processamento.
- Portanto, em algum ponto do algoritmo geralmente deve ocorrer a exibição de valores.

- Programas precisam mostrar os resultados de seu processamento.
- Portanto, em algum ponto do algoritmo geralmente deve ocorrer a exibição de valores.
- Todas as linguagens de programação têm comandos para permitir a saída de dados.

- Comando print (Em Python):

- Comando print (Em Python):
 - É utilizado para comunicação entre o programa e o usuário;

- Comando print (Em Python):
 - É utilizado para comunicação entre o programa e o usuário;
 - Permite que o programa forneça informações ao usuário.

- Comando print (Em Python):
 - É utilizado para comunicação entre o programa e o usuário;
 - Permite que o programa forneça informações ao usuário.
 - Sintaxes:

```
print("Sequência de caracteres")  
print("Sequência de caracteres", variavel)  
print("Sequência de caracteres", variavel, "Sequência de caracteres")  
etc
```

- Exemplos em Python:

- Exemplos em Python:
`print(idade)`

- Exemplos em Python:
`print(idade)`
`print("Qual é a sua idade?")`

- Exemplos em Python:
`print(idade)`
`print("Qual é a sua idade?")`
`print("Minha idade é ", idade)`

- Exemplos em Python:

```
print(idade)
```

```
print("Qual é a sua idade?")
```

```
print("Minha idade é ", idade)
```

```
print("Minha idade é ", idade, "e tenho ", altura, " de altura")
```

- O método format() serve para criar uma string que contem campos entre chaves a serem substituídos pelos argumentos de format

- O método `format()` serve para criar uma string que contem campos entre chaves a serem substituídos pelos argumentos de `format`
 - `str = 'O filme {0} merece {1} estrelas'`
 - `str.format('Exterminador do Futuro', 4)`

EXPRESSÕES E OPERADORES ARITMÉTICOS

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração
 - * : Multiplicação

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração
 - * : Multiplicação
 - ** : Potenciação

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração
 - * : Multiplicação
 - ** : Potenciação
 - / : Divisão real

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração
 - * : Multiplicação
 - ** : Potenciação
 - / : Divisão real
 - // : Divisão inteira

- Expressão aritmética é aquela cujos operadores são aritméticos e cujos operandos são constantes ou variáveis do tipo numérico (inteiros ou reais)
- Operadores Aritméticos em Python:
 - + : Adição
 - - : Subtração
 - * : Multiplicação
 - ** : Potenciação
 - / : Divisão real
 - // : Divisão inteira
 - % : Resto da divisão de inteiros

- Exemplos em Python:

- Exemplos em Python:
 - Adição: $2 + 5$; $x + y$

- Exemplos em Python:
 - Adição: $2 + 5$; $x + y$
 - Subtração: $10 - 8$; $x - y$

- Exemplos em Python:
 - Adição: $2 + 5$; $x + y$
 - Subtração: $10 - 8$; $x - y$
 - Multiplicação: $7 * 8$; $x * y$

- Exemplos em Python:
 - Adição: $2 + 5$; $x + y$
 - Subtração: $10 - 8$; $x - y$
 - Multiplicação: $7 * 8$; $x * y$
 - Divisão real: $3.0 / 2.0$; x / y

- Exemplos em Python:
 - Adição: $2 + 5$; $x + y$
 - Subtração: $10 - 8$; $x - y$
 - Multiplicação: $7 * 8$; $x * y$
 - Divisão real: $3.0 / 2.0$; x / y
 - Resto da divisão de inteiros: $3 \% 2$; $x \% y$

- Exemplos em Python:

- Adição: $2 + 5$; $x + y$
- Subtração: $10 - 8$; $x - y$
- Multiplicação: $7 * 8$; $x * y$
- Divisão real: $3.0 / 2.0$; x / y
- Resto da divisão de inteiros: $3 \% 2$; $x \% y$
- Potenciação: $2^{**}3$; $x^{**}y$ ou $\text{pow}(2,3)$; $\text{pow}(x,y)$

- Exemplos em Python:

- Adição: $2 + 5$; $x + y$
- Subtração: $10 - 8$; $x - y$
- Multiplicação: $7 * 8$; $x * y$
- Divisão real: $3.0 / 2.0$; x / y
- Resto da divisão de inteiros: $3 \% 2$; $x \% y$
- Potenciação: $2^{**}3$; $x^{**}y$ ou $\text{pow}(2,3)$; $\text{pow}(x,y)$

- Na biblioteca math:

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x
 - `math.log(x)` : Obtém o valor do logaritmo de x na base e ($\ln x$)

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x
 - `math.log(x)` : Obtém o valor do logaritmo de x na base e ($\ln x$)
 - `math.log10(x)` : Obtém o valor do logaritmo de x na base 10

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x
 - `math.log(x)` : Obtém o valor do logaritmo de x na base e ($\ln x$)
 - `math.log10(x)` : Obtém o valor do logaritmo de x na base 10
 - `math.sin(x)` : Calcula o seno de x

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x
 - `math.log(x)` : Obtém o valor do logaritmo de x na base e ($\ln x$)
 - `math.log10(x)` : Obtém o valor do logaritmo de x na base 10
 - `math.sin(x)` : Calcula o seno de x
 - `math.cos(x)` : Calcula o cosseno de x

- Na biblioteca math:
 - `math.sqrt(x)` : Raiz quadrada de x
 - `math.exp(x)` : Obtém o valor de e^x
 - `math.log(x)` : Obtém o valor do logaritmo de x na base e ($\ln x$)
 - `math.log10(x)` : Obtém o valor do logaritmo de x na base 10
 - `math.sin(x)` : Calcula o seno de x
 - `math.cos(x)` : Calcula o cosseno de x
 - ...

	Prioridade
Parênteses mais internos	1 (Maior)
Funções matemáticas	2
$*, /, \%$	3
$+, -$	4 (Menor)

	Prioridade
Parênteses mais internos	1 (Maior)
Funções matemáticas	2
$*, /, \%$	3
$+, -$	4 (Menor)

- Os critérios de prioridade são seguidos no cálculo de uma expressão aritmética;

	Prioridade
Parênteses mais internos	1 (Maior)
Funções matemáticas	2
$*, /, \%$	3
$+, -$	4 (Menor)

- Os critérios de prioridade são seguidos no cálculo de uma expressão aritmética;
- Operadores de mesma prioridade a avaliação é feita da esquerda para a direita.

- Analise o programa em Python abaixo e descreva o que ele faz

programa1

February 11, 2019

```
In [ ]: matricula = int(input("Digite a matrícula: "))
        nota1 = float(input("Digite a 1ª nota: "))
        nota2 = float(input("Digite a 2ª nota: "))
        nota3 = float(input("Digite a 3ª nota: "))
        media = (nota1 + nota2 + nota3)/3
        print("{} aluno de matrícula {} obteve media = {:.2f}".format(matricula,media))
```