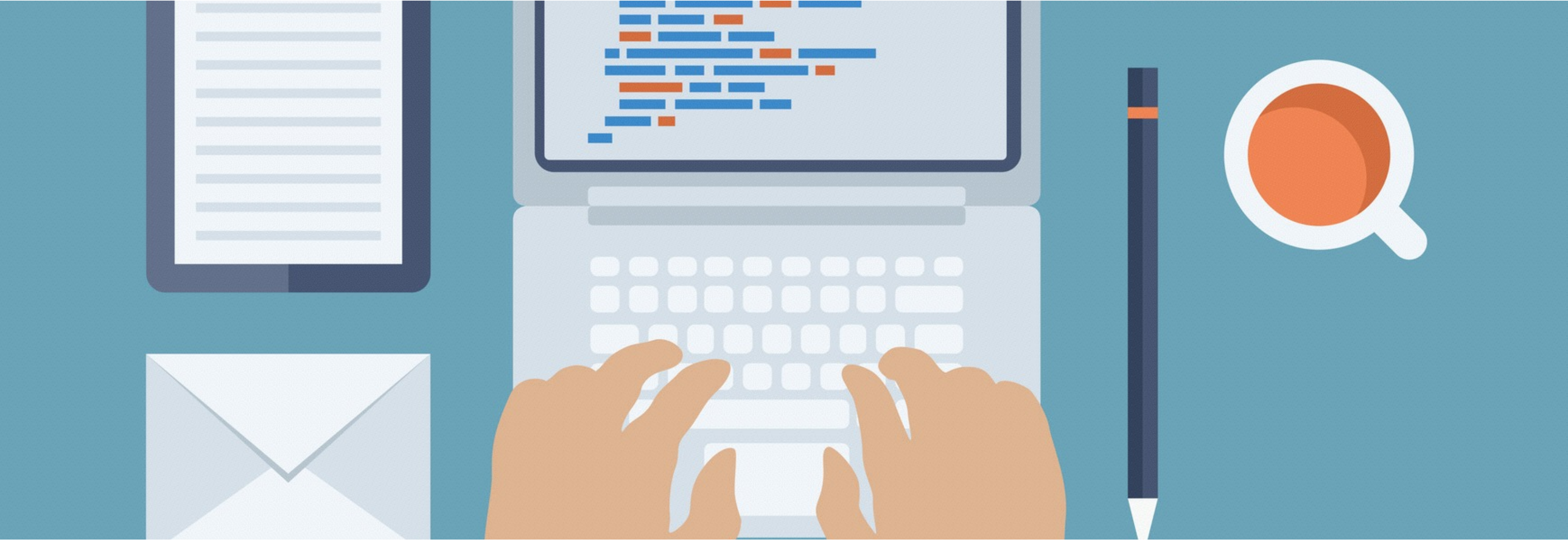


UNIVERSIDADE FEDERAL DO CEARÁ

DEPARTAMENTO DE COMPUTAÇÃO

CIÊNCIA DA COMPUTAÇÃO

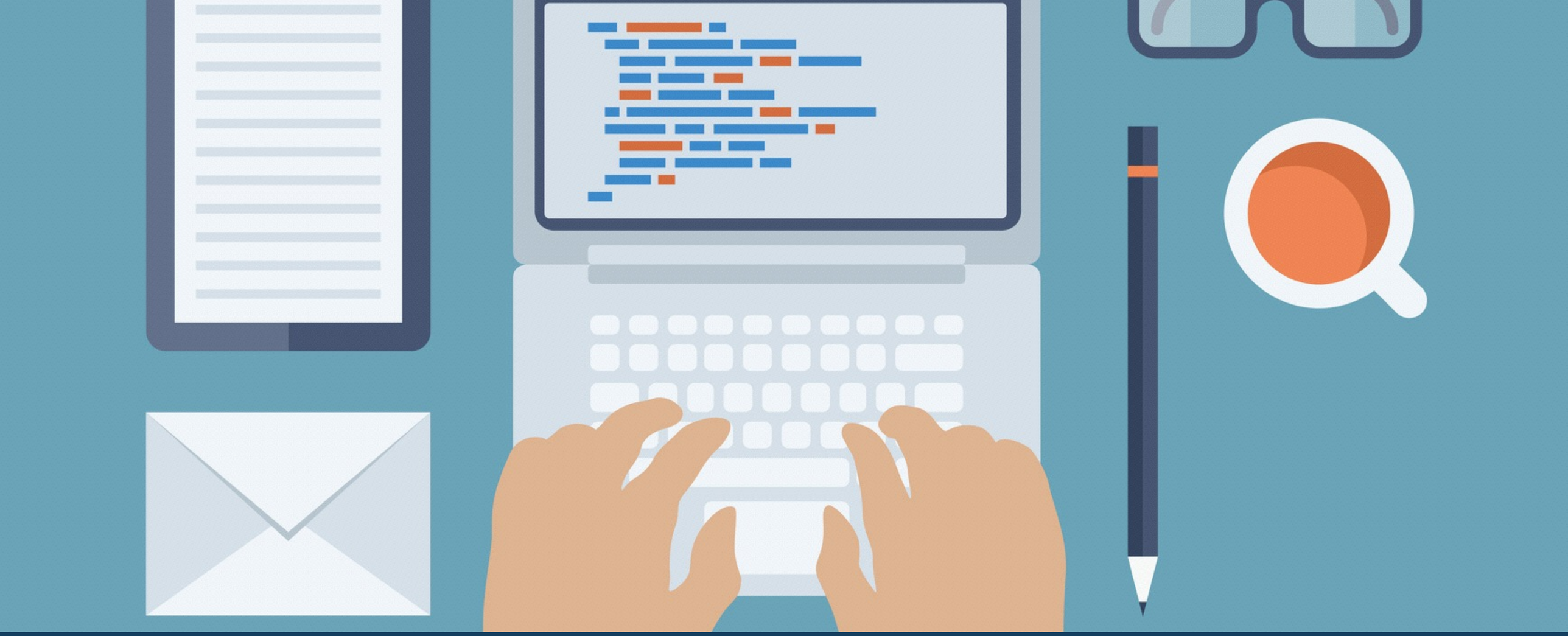
CK0211 - Fundamentos de Programação: Paradigmas de Programação e Estrutura Sequencial

Emanuele Santos

Bibliografia: Ascencio, Cap. 2 e 3

Objetivos

- Apresentar os paradigmas de programação Estruturado e Orientado a Objetos
- Apresentar a Estrutura Sequencial em Algoritmos
- Apresentar a Estrutura Sequencial em Python



PARADIGMAS DE PROGRAMAÇÃO

Introdução

- Um paradigma de programação está relacionado com a forma de pensar do programador para resolver problemas
- Cada paradigma tem as suas próprias técnicas de programação
- Existem vários tipos de paradigmas, mas os principais são os paradigmas estruturado e o orientado a objetos

Paradigma Estruturado

- Também conhecido como imperativo ou procedural
- Modularização:
  - Qualquer problema pode ser quebrado em problemas menores, de mais fácil solução
    - Sub-rotinas ou **funções** ou procedures
  - Cada **função** pode receber valores de entrada, realizar um processamento e gerar um valor de saída para quem chamou a função

Exemplo do uso de funções

- Na aula passada, usamos a função **input** em Python:

```
nome = input('Qual o seu nome? ')
```

- A função input captura os dados de entrada do usuário. Para isso ela recebe como entrada um texto que será mostrado ao usuário e retorna como saída, o que o usuário digitou também como texto

Paradigma Estruturado

- Todo processamento no paradigma estruturado pode ser feito usando três tipos de estrutura:
  - Sequencial
  - Condicional
  - De repetição (iterativa)

Paradigma Orientado a Objetos

- Lida com o problema a ser resolvido como uma coleção de objetos interagindo por meio de trocas de mensagem
- **Objetos**: Estruturas de dados contendo **estado** (dados) e **comportamento** (lógica)
- **Classe**: Conjunto de objetos com informações comuns e com o mesmo comportamento

Exemplo do uso de objetos

- Exemplo hipotético do uso de objetos em Python

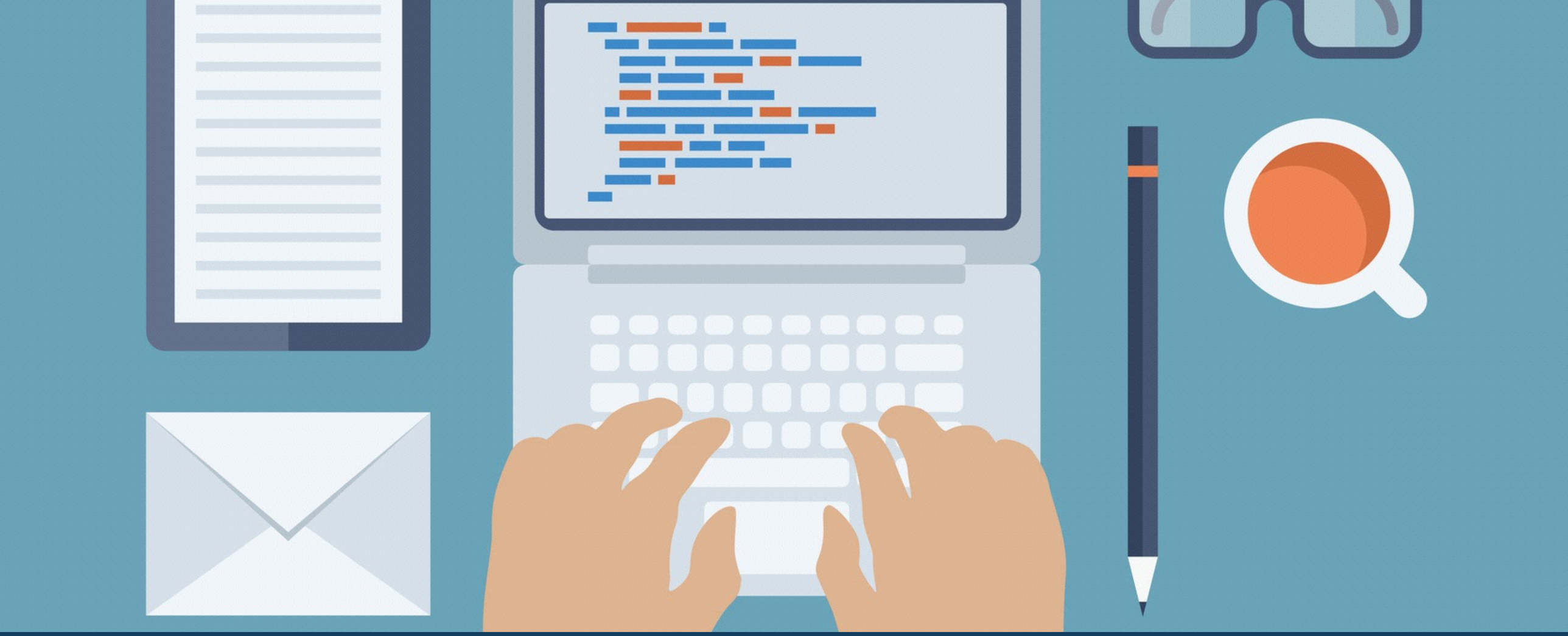
```
livro1 = Livro(titulo="O Hobbit", autor="J R R Tolkien")
livro2 = Livro(titulo="O Silmarillion", autor="J R R Tolkien")
biblioteca.insere(livro1)
biblioteca.insere(livro2)
```

Linguagens de Programação e Paradigmas (1/2)

- Um paradigma de programação está ligado à forma de pensar do programador
- Cada linguagem de programação atende a pelo menos um paradigma
- Algumas linguagens de programação possuem suporte nativo à orientação a objetos:
  - Java, Python, C++ entre outras
- Isso não implica que todos os programas escritos nessas linguagens serão orientados a objetos (vai depender de como o programador pensou)

Linguagens de Programação e Paradigmas (2/2)

- Os paradigmas não são mutuamente exclusivos: o paradigma orientado a objetos não exclui o paradigma estruturado
  - Toda a lógica embutida nos objetos segue o pensamento estruturado
- Ao longo desse semestre, estudaremos o paradigma estruturado
- O paradigma orientado a objetos é estudado na disciplina Técnicas de Programação I



ESTRUTURA SEQUENCIAL

Estrutura Sequencial em Algoritmos

```
ALGORITMO
  DECLARE nome_da_variável tipo_da_variável
  bloco_de_comandos
FIM_ALGORITMO.
```

Declaração de variáveis em algoritmos

- As variáveis são declaradas após a palavra **DECLARE**
- Os tipos mais utilizados são:
  - **NUMÉRICO**: para variáveis que receberão números
  - **LITERAL**: para variáveis que receberão caracteres
  - **LÓGICO**: para variáveis que receberão apenas dois valores: verdadeiro ou falso

Declaração de variáveis em algoritmos

- Exemplo:

```
ALGORITMO
  DECLARE x NUMÉRICO
          y,Z LITERAL
          teste LÓGICO
  bloco_de_comandos
FIM_ALGORITMO.
```

Comando de atribuição em algoritmos

- Atribui valores ou o resultados de operações (expressões) a variáveis
- Representado pelo símbolo: ←

```
ALGORITMO
  DECLARE x NUMÉRICO
          y,Z LITERAL
          teste LÓGICO
  x ← 4
  x ← x + 2
  y ← “aula”
  teste ← falso
FIM_ALGORITMO.
```

Comando de entrada em algoritmos

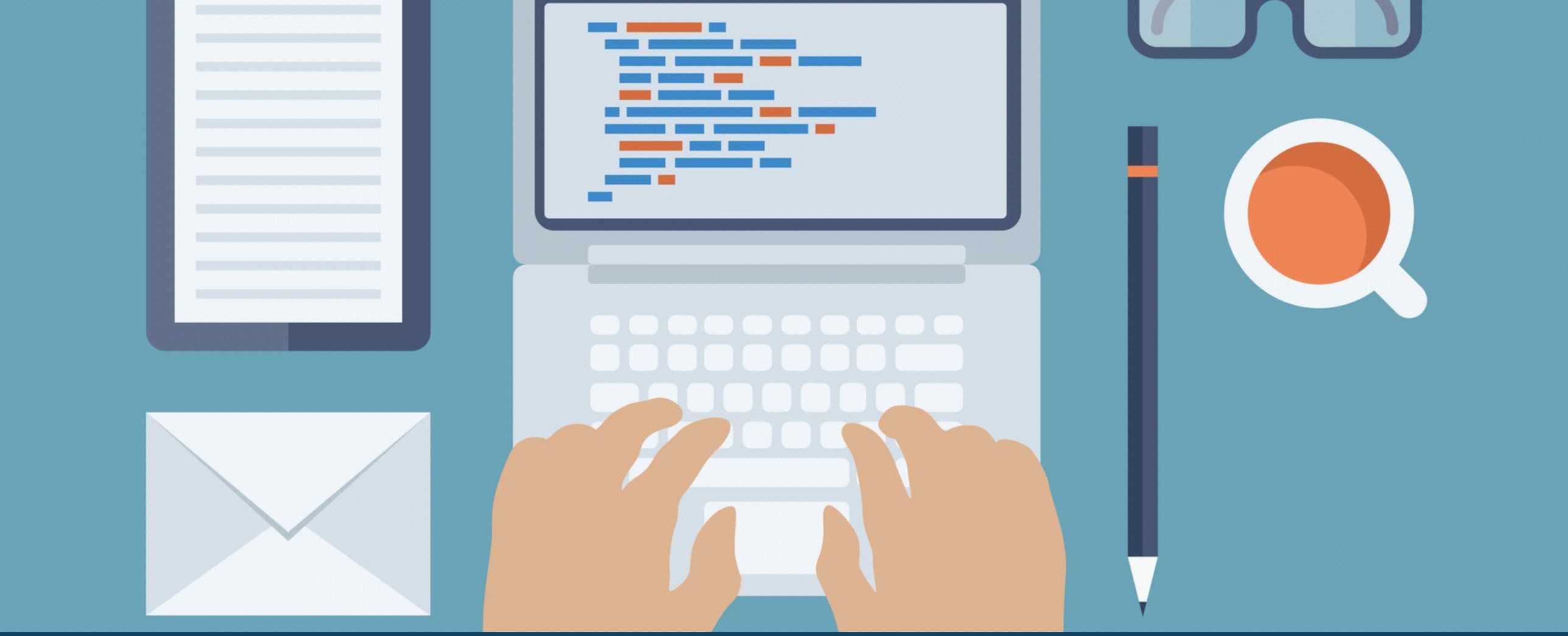
- Utilizado para receber dados digitados pelo usuário, os quais serão armazenados em variáveis
- Comando representado pela palavra **LEIA**

```
ALGORITMO
  DECLARE x NUMÉRICO
          y,Z LITERAL
          teste LÓGICO
  LEIA x
  LEIA y
FIM_ALGORITMO.
```

Comando de saída em algoritmos

- Utilizado para mostrar dados na tela ou na impressora
- Comando representado pela palavra **ESCREVA**
- Os dados podem ser variáveis ou expressões

```
ALGORITMO
  DECLARE x NUMÉRICO
          y,Z LITERAL
          teste LÓGICO
  LEIA x
  ESCREVA x
  ESCREVA “O dobro de x é igual a ”, 2*x
FIM_ALGORITMO.
```



ESTRUTURA SEQUENCIAL EM PYTHON

Declaração de variáveis e o comando de atribuição em Python

- Em Python é necessário apenas o nome da variável, seguido do símbolo **=** e o valor que ela irá armazenar

- O tipo da variável será o mesmo tipo de dado que ela armazena

```
idade = 23
preco = 56.78
teste = False
nome = “Emanuele”
```

Declaração de variáveis em outras linguagens

- Em C/C++ e Java:
- int idade;
- Em Pascal:
- VAR idade: INTEGER;

Descobrindo tipos das variáveis

- Usaremos a função **type**
- **type** retorna o tipo do valor passado entre parênteses

```
idade = 23
print(type(idade)) # <class 'int'>
preco = 56.78
print(type(preco)) # <class 'float'>
teste = False
print(type(teste)) # <class 'bool'>
nome = “Emanuele”
print(type(nome)) # <class 'str'>
```

Comentários em Python

- Os comentários são textos que serão ignorados pelo interpretador (ou pelo compilador)
- São utilizados para explicar os comandos utilizados
- Em Python existem duas maneiras de criar comentários
  - Usando o símbolo # para um comentário de uma única linha
  - Ou delimitados por “““ … ”””

Exemplos de Comentários

```
# o comando abaixo não será mostrado
# print(“oi!”)
print(“esse será mostrado”)
""" Este comentário ocupa
múltiplas linhas """
```

Entrada de dados em Python

- Usaremos a função input
- input mostra um literal passado na função e retorna um literal contendo o que foi digitado pelo usuário (sempre do tipo texto)
- Guardamos o valor retornado pela função input em uma variável

```
nome = input('Qual o seu nome?')
idade = input('Quantos anos você tem?’)
print(type(idade)) # o que será mostrado?
```

Entrada de dados em Python

- Usar as funções **int** ou **float** para converter a entrada de dados para um valor numérico

```
nome = input('Qual o seu nome?')
idade = int(input(‘Quantos anos você tem?’))
print(type(idade)) # o que será mostrado?
```

Saída de dados em Python

- Usaremos a função print
- **print** aceita uma sequência de valores a serem mostrados na tela, que podem ser de qualquer tipo: numérico, literal, incluindo variáveis

```
nome = input('Qual o seu nome?')
print('Boa tarde,', nome) # o que será mostrado?
```

Operadores predefinidos em Python

Operador	Exemplo	Comentário
=	x = y	O conteúdo da variável y é atribuído à variável x
+	x + y	Soma o conteúdo de x e de y
-	x - y	Subtrai o conteúdo de y do conteúdo de x
*	x * y	Multiplica o conteúdo de x por pelo conteúdo de y
/	x / y	Divide o conteúdo de x pelo conteúdo de y (Python v3)
%	x % y	Obtém o resto da divisão de x por y (inteiros)
//	x // y	Obtém o quociente inteiro da divisão de x por y (inteiros)
**	x ** y	Eleva o conteúdo de x à potência do conteúdo de y

Operadores predefinidos em Python

Operador	Exemplo	Comentário
+=	x += y	Equivale a x = x + y
-=	x -= y	Equivale a x = x - y
*=	x *= y	Equivale a x = x * y
/=	x /= y	Equivale a x = x / y
%=	x %= y	Equivale a x = x % y
//=	x //= y	Equivale a x = x // y

Operadores predefinidos em Python

Operador	Exemplo	Comentário
==	x == y	O conteúdo de x é igual ao conteúdo de y
!=	x != y	O conteúdo de x é diferente do conteúdo de y
<=	x <= y	O conteúdo de x é menor ou igual ao conteúdo de y
>=	x >= y	O conteúdo de x é maior ou igual ao conteúdo de y
<	x < y	O conteúdo de x é menor que o conteúdo de y
>	x > y	O conteúdo de x é maior que o conteúdo de y

Funções predefinidas em Python

Função	Exemplo	Comentário
`abs`	`abs(x)`	Obtém o valor absoluto de x
`int`	`int(x)`	Converte x para inteiro
`float`	`float(x)`	Converte x para real
`divmod`	`divmod(x, y)`	O par (x // y, x % y)
`pow`	`pow(x, y)`	O mesmo que x ** y

Funções matemáticas em Python

```
import math
```

Função	Exemplo	Comentário
`math.sqrt`	`math.sqrt(x)`	Calcula a raiz quadrada de x
`math.floor`	`math.floor(x)`	Arredonda um número real para baixo
`math.ceil`	`math.ceil(x)`	Arredonda um número real para cima
`round(x,n)`	`round(x,2)`	Arredonda x para conter n dígitos. Se n for omitido, o default é 0

Usando o interpretador para ver todas as funções em math

```
>>> import math
>>> dir(math)
['__doc__', '__file__', '__loader__', '__name__', '__package__',
'__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2',
'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e',
'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor',
'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf',
'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma',
'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow',
'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

Usando o interpretador para saber o que uma função faz

```
>>> import math
>>> help(math.log)
```

```
Help on built-in function log in module math:

log(...)
    log(x[, base])

    Return the logarithm of x to the given base.
    If the base not specified, returns the natural logarithm
(base e) of x.
(END)
```

Aperte a tecla q para voltar

Exercícios

- Escreva o algoritmo em pseudocódigo e o programa em Python para cada um dos exercícios a seguir

Programa 1

- Suponha que você deseja preencher a seguinte ficha de inscrição de um estudante:

nome:

matrícula:

curso:

idade:

e-mail:

- Escreva um programa que pede os dados do usuário e mostra a ficha preenchida

Programas 2 a 4

- Faça um programa que receba três notas e calcule e mostre a média aritmética

- Faça um programa que receba duas notas e seus respectivos pesos, calcule e mostre a média ponderada

- Faça um programa que calcule e mostre a área de um círculo. Sabe-se que área = $\pi * R^2$

Programa 5

- Faça um programa que receba um número positivo e maior que zero, calcule e mostre:
  - a) o número digitado ao quadrado
  - b) o número digitado ao cubo
  - c) a raiz quadrada do número digitado
  - d) a raiz cúbica do número digitado

Programa 6

- Escreva um programa que pede os seguintes dados:
  - Valor do salário de um funcionário
  - Aumento em porcentagem
- Depois mostre o valor do aumento e o salário com aumento arredondados para duas casas decimais

```
Digite o salário: 1000.45
Digite o aumento em porcentagem: 15
O aumento será: R$150.07
E o salário com aumento será: R$1150.52
```