



Aula 01: **Apresentação**

Programação de Computadores I

Túlio Toffolo

www.toffolo.com.br

Departamento de Computação
Universidade Federal de Ouro Preto

Agenda

1. Objetivo, ementa e avaliação
2. Moodle e site da disciplina
3. Metodologia
4. Dicas de ferramentas e ambiente

Objetivo, ementa e avaliação

Objetivo

O objetivo principal dessa disciplina é:

- Desenvolver a capacidade de construir **programas de computadores**, de pequeno e médio porte, com a utilização de conceitos de **programação estruturada**;

Ementa

- Introdução a ambientes de programação.
- Conceitos de algoritmo.
- Conceitos básicos de programação:
 - valores e expressões de tipos primitivos,
 - variáveis,
 - comando de atribuição,
 - comandos de controle de fluxo,
 - entrada e saída padrão,
 - procedimentos e funções,
 - tipos de dados compostos.

Avaliação

Provas Teóricas: 70% da nota

- PT_1 (Prova Teórica 1 - sábado, dia 13/03 às 10h): 30% da nota
- PT_2 (Prova Teórica 2 - sábado, dia 17/04 às 10h): 40% da nota

Exercícios Propostos: 30% da nota

- EP_1 (Exercícios Propostos de 18/01 a 12/03): 15% da nota
 - EP_2 (Exercícios Propostos de 15/03 a 23/04): 15% da nota
- Note que os exercícios serão utilizados para aferir a frequência!

Avaliação em detalhes

1ª Parte da disciplina: $PT_1 + EP_1$

PESO	ATV.	DATA	TEMA
1,00	P-02	28/jan.	Prática: conceitos básicos
	P-03	04/fev.	Prática: fluxogramas e comandos de decisão
	P-04	11/fev.	Prática: comandos de decisão
	P-06	25/fev.	Prática: comandos de repetição
	P-07	04/mar.	Prática: laços aninhados
0,50	P-08	11/mar.	Prática especial: preparação para a prova
3,00	PT-01	13/mar.	Prova Teórica 01

Avaliação em detalhes

2ª Parte da disciplina: $PT_2 + EP_2$

PESO	ATV.	DATA	TEMA
1,00	P-09	18/mar.	Prática: funções
	P-10	25/mar.	Prática: vetores
	P-12	08/abr.	Prática: matrizes
	P-14	22/abr.	Prática: matrizes
0,50	P-13	15/abr.	Prática especial: preparação para a prova
4,00	PT-02	17/abr.	Prova Teórica 02

Frequência

Muito importante:

- A frequência será computada por meio das entregas feitas na plataforma *moodle*.
 - Fiquem atentos aos **prazos** das atividades!!
 - Sejam **rigorosos** com os horários de entrega!
-
- Quem perder mais de **25% das atividades** será **reprovado**.
 - E não terá direito ao **exame especial**.

Moodle e site da disciplina

Moodle

- Utilizaremos a plataforma *moodle* para a condução da disciplina.
 - Um **guia de utilização** do *moodle* foi disponibilizado pelo CEAD/UFOP para os alunos.
 - www.moodlepresencial.ufop.br



BCC701 - Programação de Computadores I

Túlio Toffolo & Reinaldo Fortes
www.toffolo.com.br



Retomada do Período Letivo 2020/1

Site da disciplina

Todo o material da disciplina pode ser acessado via *moodle*.

No entanto, pode ser mais rápido acessar vídeos e o livro-texto no site da disciplina.

- www.toffolo.com.br
- Ensino → BCC701: Programação de Computadores I.
- Acesso irrestrito (não requer login/senha).

Metodologia

Plano de ensino

- O [plano de ensino](#) detalhado da disciplina pode ser baixado diretamente do *moodle* ou do site da disciplina.
- O [calendário de entrega das atividades](#) também está disponível.

Metodologia de Ensino

Toda semana teremos aulas **teóricas** e **práticas**.

Aulas teóricas

- Serão assíncronas, oferecidas por meio de vídeos e textos.
- Serão disponibilizadas (via *moodle*) no horário da aula.
- Eventuais dúvidas podem ser sanadas nas monitorias, aulas práticas ou por meio do **fórum** disponível no *moodle*.

Metodologia de Ensino

Toda semana teremos aulas **teóricas** e **práticas**.

Aulas práticas

- Exercícios serão postados no moodle no dia anterior (quarta).
- As aulas práticas serão síncronas e para sanar dúvidas, sempre realizadas via **Google Meet**.
- Vocês terão dois canais para tirar dúvidas (e dois professores): eu e o *Prof. Reinaldo Fortes*.
- Links para acesso estão disponíveis no *moodle*.

Dicas de ferramentas e ambiente

Linguagens de programação

Existem diversas linguagens nas quais podemos desenvolver nosso código fonte. Entre elas estão:

- C / C++
- Java
- Javascript
- Julia
- Matlab/Octave
- **Python 3**
- ...

Linguagem Python

- Uma das linguagens mais populares e utilizadas.
- Simples de entender e mais ainda de usar.
- É como escrever uma lista de instruções!!

Exemplo:

```
1 x = 10
2 y = x * x * x
3
4 print(f"O resultado do processamento é {y}")
```

Testando seus códigos Python

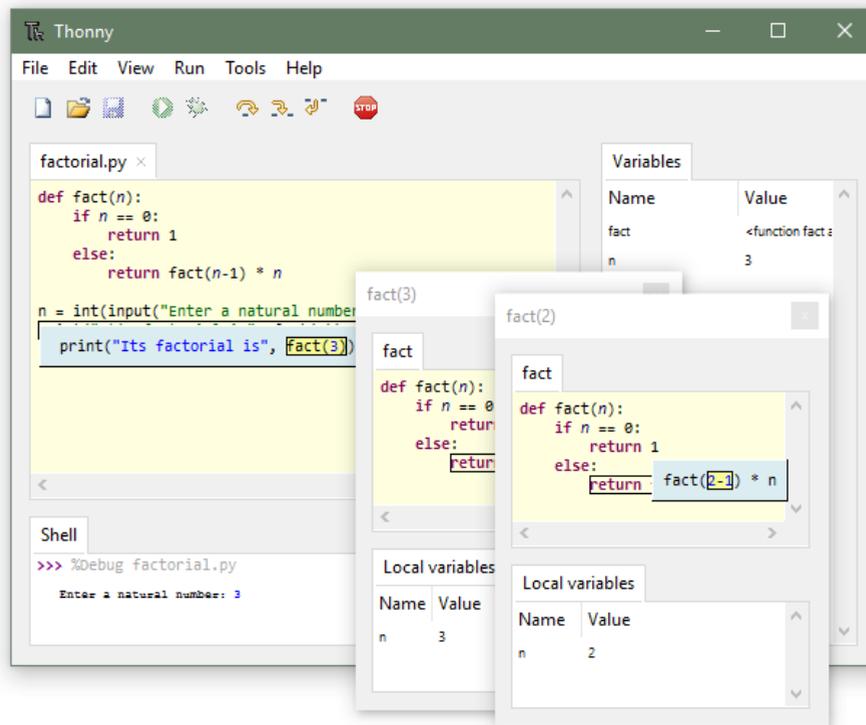
Há **inúmeras** ferramentas excelentes para auxiliar no desenvolvimento de códigos Python. *Algumas sugestões:*

- Atom
- IDLE
- **Pycharm**
- Pydev
- Spyder
- **Visual Studio Code**

No entanto, para os iniciantes recomendamos outras ferramentas...

Testando seus códigos Python

- Recomendação 1: o simples **Thonny**



Testando seus códigos Python

- Recomendação 2: a ferramenta web [Online Python](#)



The screenshot shows a web-based Python IDE interface. At the top, there are navigation icons (file, lock, refresh, copy) and system icons (moon, power, settings). The main editor area shows a file named 'main.py' with the following Python code:

```
1 x = 10
2 y = x * x * x
3
4 print(f"Resultado = {y}")
```

Below the code editor, the status bar indicates 'Ln: 4, Col: 26'. There are 'Run' and 'Share' buttons, and a field for 'Command Line Arguments'. The output terminal at the bottom shows the execution result:

```
Resultado = 1000
** Process exited - Return Code: 0 **
Press Enter to exit terminal
```

Perguntas?
