

AULA 1

1. Apresentação do curso

Olá! Bem vindo ao curso de introdução à programação da UFRJ. :-) Este é o material de nossa primeira aula. Além de conhecer os objetivos e o formato do curso, essa primeira semana também será uma oportunidade para conhecer seu professor e alguns dos monitores das atividades práticas.

Este curso de introdução à programação é oferecido para estudantes de várias áreas dentro das ciências exatas e da terra da UFRJ. A programação é uma competência cada dia mais valiosa a todas as pessoas, e é muito importante para os universitários destas áreas.

Assista ao vídeo “porque aprender a programar”:

- <https://youtu.be/mHW1Hsqlp6A>

Agora, convidamos você a assistir o vídeo de apresentação deste curso:

- [Introdução - Apresentação do Curso](#)

Objetivos do curso

O objetivo deste curso é o **desenvolvimento das competências fundamentais da programação de computadores**. Para tanto, vamos aprender muitos **conceitos**, porém principalmente, temos que **praticar bastante para nos tornarmos competentes** em:

- Identificação das informações relevantes de um problema e sua respectiva representação e manipulação na programação.
- Compreensão dos aspectos sintáticos e semânticos da linguagem de programação.
- Articulação dos comandos, estruturas de dados e estruturas básicas da programação para a construção de soluções para problemas simples.
- Modelagem e implementação de código modularizado para problemas não elementares.
- Construção de código organizado, reutilizável e legível, seguindo os princípios das boas práticas de programação

Fique atento: A programação é uma atividade complexa, que envolve conhecimento, prática e proficiência. É imprescindível praticar muito! Além disso, receber bons fundamentos no estágio inicial é essencial para a capacitação de um programador. Procure sempre fontes confiáveis de informação, dando preferência a material produzido por instituições consolidadas nessa área, como por exemplo universidades e centros ou organizações de formação e ensino associados ao ministério da educação ou à CAPES.

2. Conceitos iniciais

Você sabe o que é um programa? Sabe a diferença entre programa e aplicativo (ou app)? Explore os links abaixo para ter um primeiro contato com definições destes conceitos, que são elementares para nosso curso. Você vai descobrir que, além dos conceitos de programa e aplicativo, outros conceitos como algoritmo e linguagem de programação vão aparecer de forma interligada. Ao fazer sua leitura do conteúdo abaixo, anote suas impressões e dúvidas, e aproveite a aula síncrona para conversar com seu professor e aprofundar seu entendimento.

Programa, Aplicativo e Algoritmo

- [programa de computador](#)
- [Software aplicativo](#)
- [Programa de computador](#)

Veja agora alguns exemplos de problemas a serem resolvidos e algoritmos para resolver esses problemas.

- [Primeiros algoritmos](#) - nesse vídeo falamos de dois problemas e chegamos a um algoritmo para a solução de cada um dos problemas. Ao final do segundo problema pedimos que proponha um problema similar, que se resolva da mesma maneira.
- [Primeiros algoritmos - mais um exemplo](#) - nesse outro vídeo estudamos um problema menos trivial, que exige um maior esforço de busca da solução.

Funcionamento do computador, Linguagem de programação e Execução de programas

Para ter uma ideia melhor de como os programas são executados pelo computador, temos que ter um entendimento inicial de como o computador funciona.

Assista ao vídeo

- [Como funciona um computador \(animação\)](#)

Agora, leia um pouco mais sobre o assunto e sobre como um programa de computador pode ser executado em:

- [Como funciona um programa de computador](#)

Entenda melhor o conceito de linguagem de programação:

- [O que é linguagem de programação?](#)
- [Linguagens de programação. O que são e para que servem?](#)

Agora você já sabe que programas são escritos em uma linguagem de programação, e que o computador, para executá-los, tem que saber como as instruções escritas na linguagem de programação são mapeadas para a sua linguagem própria, a linguagem de máquina. Duas maneiras de fazer isso são a compilação e a interpretação. Veja mais sobre elas em:

- [Interpretadores, compiladores e tradutores](#)
- [Qual a diferença entre código compilado e código interpretado?](#)

Neste curso, adotaremos a linguagem de programação Python. Ela é considerada uma excelente linguagem para iniciantes em programação, e também é uma linguagem amplamente utilizada com propósitos comerciais, acadêmicos e de pesquisa.

Veja este artigo sobre as razões para se aprender a programar em Python:

- [Por que aprender Python pode te levar mais longe na carreira!](#)

Conheça site oficial da linguagem de programação Python

- <https://www.python.org/>

Atividade: Vamos convidá-lo agora a exercitar os conhecimentos que acabamos de ver. Você vai agora fazer uma atividade respondendo algumas perguntas sobre os conceitos que acabamos de estudar. Fique atento: o importante é escrever respostas suas, próprias, que revelem o seu entendimento. É inútil copiar respostas de qualquer material. Nossa equipe já sabe esses conceitos, estamos interessados em ver como você os entendeu, para que possamos orientar nossas próximas ações educativas. Se você coloca respostas copiadas, ninguém se beneficia, nem nós, nem você mesmo. Ao contrário, você tirará grande proveito do exercício ao tentar elaborar suas próprias respostas, pois é nesse processo que você tem a oportunidade de refletir sobre o que foi estudado, articular cada parte de informação que está na sua memória e consolidar seus conhecimentos. Então respire fundo, e responda o formulário que está no google classroom.

3. Primeiros passos na programação

Ter uma instalação local do Python é importante para as práticas deste curso! Siga as instruções a seguir para fazer o download e a instalação do Python no seu computador.

Este link leva você para a página de download da versão mais recente do Python compatível com o sistema operacional da sua máquina:

<https://www.python.org/downloads/>

Nos links abaixo estão instruções que podem te ajudar no processo de instalação. Escolha o link de acordo com o sistema operacional que você usa. As informações sobre “pip” não são importantes no momento, caso não você não as entenda, não se preocupe com isso agora.

- Passo a passo de instalação de Python para Windows: <https://python.org.br/instalacao-windows/>
- Passo a passo de instalação de Python para Linux: <https://python.org.br/instalacao-linux/>
- Passo a passo de instalação de Python para Mac: <https://python.org.br/instalacao-mac/>

Caso você não tenha conseguido instalar, peça ajuda à equipe de monitores ou ao seu professor. Consulte do vídeo tutorial de instalação do Python:

[Instalando o Python](#)

Uma vez instalado o Python, podemos abrir o programa IDLE, que vem com o Python, no seu computador. Basta usar a busca do sistema operacional, que geralmente está na barra de

tarefas, na parte de baixo da tela do seu computador. Digite IDLE, e deve aparecer para você o link para abrir o programa de que estamos falando. Clique nele. Aí você está pronto para acompanhar as próximas etapas que vamos seguir para escrever seu primeiro código.

Os monitores também prepararam um vídeo mostrando como abrir e começar a usar o IDLE. Ele está disponível aqui. [Python e ferramentas: editor IDLE](#)

Operações aritméticas - Usando o python para fazer contas.

A primeira coisa que vamos aprender a fazer em Python são operações aritméticas. A forma de escrever (sintaxe) da linguagem Python para expressões aritméticas é semelhante à utilizada pelas calculadoras comuns. Escrevemos a expressão desejada intercalando operadores e operandos. Além dos números e dos sinais que representam cada um dos operadores, podemos usar também os parênteses para identificar trechos da fórmula que devem ser resolvidos primeiro.

Os operadores aritméticos mais comuns da linguagem Python são os seguintes:

- + (soma)
- - (subtração)
- * (multiplicação - atenção que é o símbolo de asterisco, não é o x)
- / (divisão)

Para representar a operação de exponenciação (x elevado a y), o Python usa o operador:

- ** (exponenciação)

Existem também dois operadores interessantes, derivados da operação de divisão, quando estamos interessados em divisões inteiras (ou seja, onde paramos a operação de divisão antes de termos casas decimais no resultado):

- // (quociente da divisão inteira)
- % (resto da divisão inteira, operador que chamamos de módulo)

O funcionamento destes operadores é explicado no vídeo:

[Operadores Aritméticos](#)

Exercício: Vamos ver se você entendeu o que acabamos de ver. Considerando o que foi estudado sobre operadores aritméticos e sua ordem de precedência na linguagem Python, em que ordem seriam realizadas as operações das expressões abaixo? Faça as operações nessa ordem e anote o resultado.

a) $2 * 5 \% 4 + 6 / 3 - 1$

b) $3 * 2 ** 2 + 1 * 5 - 4$

Agora, abra o IDLE e use o Shell para saber o resultado de cada uma das expressões. Compare com os resultados que você havia calculado. Foram os mesmos? Caso não tenha sido, significa que o Python fez as operações em uma ordem diferente da que você fez. Tente identificar o que não foi feito como você esperava e porquê.

Agora que já sabemos como escrever expressões aritméticas em Python, iremos finalmente começar a construir nosso primeiro código em Python. Como em toda linguagem, temos que escolher um dos possíveis estilos permitidos para nos expressar. Neste curso, usaremos a programação estruturada modular, que é organizada em pequenos trechos de código, os módulos.

O primeiro recurso do python que vamos aprender para fazer nossos módulos são **funções de programação**. Provavelmente você se lembra do conceito matemático de função.

Na matemática, a intuição por trás da definição de uma função é a descrição de uma maneira de associar a cada valor da(s) entrada(s) um valor de resultado. Considere por exemplo a função f abaixo, que associa a cada número o seu dobro:

$$f(x) = 2 * x$$

A partir da descrição da função f , percebemos que ela tem apenas uma entrada, identificada pela letra x que aparece entre parênteses, ao lado do nome da função. Podemos aplicar a função f a diferentes valores de x para saber o valor que será obtido como resultado. Supondo que o domínio da função f são os números inteiros, podemos calcular o valor de f , por exemplo, para os seguintes valores de x : 2, 100, -1. Temos que $f(2)$ é 4, $f(100)$ é 200, $f(-1)$ é -2.

Relembradas as funções matemáticas, vamos agora falar sobre funções de programação: Assim como na matemática, o conceito de função em programação é a descrição de uma maneira de se obter um valor de resultado a partir dos valores das entradas. Porém na programação esta descrição segue um formato diferente da notação matemática. Veremos como isso acontece no Python.

[Escrevendo uma função em Python](#)

Agora, vamos usar o IDLE para codificar a função de exemplo “dobro”:

[Codificando a função dobro usando o editor do IDLE](#)

Exercício: Escreva uma função em Python que receba um número inteiro como entrada e retorne o sucessor desse número. Use o IDLE para codificar e testar sua função. Aproveite como exemplo o código que já foi escrito para a função dobro, disponível na sua sala de aula virtual.

Algo saiu errado? Nada mais normal! Veja se o vídeo a seguir te ajuda :-)

[Erros comuns em Python](#)

Agora que você já viu como escrever uma função e como lidar com os possíveis erros que podem acontecer durante essa tarefa, é hora de exercitar.

Exercício: baixe o arquivo `soma.py`. Salve em seu computador na pasta que você está usando para guardar seus códigos deste curso (lembre-se de manter seus arquivos organizados para facilitar seus estudos não se perder no futuro). Abra o IDLE. Use o menu superior do IDLE para abrir um arquivo, e abra o arquivo `soma.py`. Neste mesmo arquivo, e usando a função `soma` como exemplo, faça as seguintes funções em python:

- `mult(x,y)`, que retorna o produto de `x` por `y`;
- `divi(x,y)`, que retorna a divisão de `x` por `y`.

Salve seu arquivo com outro nome (por exemplo, `funcoes_mat.py`), não esquecendo de colocar `.py` ao final! Execute e teste suas funções.

Agora que você já deu seus primeiros passos na escrita de código em Python, está na hora de se preparar para as atividades práticas que devem ser feitas essa semana. Assista o vídeo sobre a tarefa da programação:

[A tarefa da programação](#)

Prática em programação

Na parte de atividades do google sala de aula, está uma lista de exercícios, que corresponde à prática desta semana. Ela contempla uma lista de funções que você deve fazer no editor do IDLE, testando sempre no shell. Seu professor dará mais detalhes sobre a entrega destes exercícios.

Material Complementar

Para se aprofundar:

https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o

Vídeo como aprender programação - histórias e dicas

<https://youtu.be/6OkQxxy5mpA>

Livros de Python em português:

<https://wiki.python.org.br/AprendaMais>

Site oficial do Python:

<https://www.python.org/>