

# USANDO PYTHON PARA A INTRODUÇÃO À PROGRAMAÇÃO EM CURSOS UNIVERSITÁRIOS

**Carla A. D. M. Delgado**

*Departamento de Ciência da Computação - UFRJ*

# INTRODUÇÃO À PROGRAMAÇÃO NA UNIVERSIDADE: CS1 E CS2

Cursos introdutórios de programação são de fundamental importância para qualquer estudante na área de ciências exatas. Porém estes cursos sofrem com:

Altas taxas de abandono e reprovação

Falta de motivação

Curva de aprendizado não linear

Medo, pânico e trauma rondam o aprendizado de programação :-)

# COMO ENSINAR PROGRAMAÇÃO PARA OS NOVATOS?

Uma ampla discussão já se estendo por anos... Comece com:

## Opção 1) **Programação imperativa estruturada**

**Obstáculo:** ausência de uma linha mestra que guie o processo de programação

**Risco:** baixa motivação, incapacidade de construir programas maiores

## Opção 2) **Programação Orientada a objetos (POO)**

**Obstáculo:** alto nível de abstração, sintaxe extensa

**Risco:** ofuscar conceitos básicos, dificultar o entendimento

# NOSSA PROPOSTA (DELGADO ET AL, WEI-CSBC 2016)

Alinhada com a ideia de “voltar ao básico”, enfatizando:

- Resolução de problemas

- Decomposição em procedimentos

- Domínio de skills básicas

Através de:

- CS1 baseado na abordagem imperativa-estruturada, focando na construção de módulos concisos de código e deixando P00 para CS2

# O QUE FAZEMOS

Adoção de **Python** como linguagem de programação

Iniciamos com a construção de funções em Python

Usamos modularização como estratégia de programação

Deixamos interação com o usuário para o final do curso

Restringimos o uso de comandos de entrada e saída à função principal do programa

# SLIDES UTILIZADOS NA PRIMEIRA SEMANA

## Executando funções interativamente

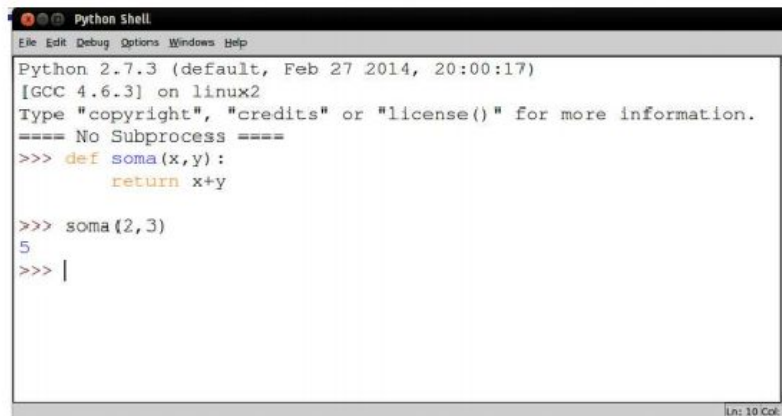
### Como definir uma função em Python

```
# Como se define uma função em Python
def nome-função(lista-parâmetros):
    return valor de retorno
```

### Soma

- Nome da Função: `soma`
- Parâmetros: `x,y`
- Valor de Retorno: `x+y`

```
# Esta é a função soma que dados os valores
# de x e y retorna o valor de x + y
def soma(x,y):
    return x+y
```



```
Python Shell
File Edit Debug Options Windows Help
Python 2.7.3 (default, Feb 27 2014, 20:00:17)
[GCC 4.6.3] on linux2
Type "copyright", "credits" or "license()" for more information.
==== No Subprocess ====
>>> def soma(x,y):
        return x+y

>>> soma(2,3)
5
>>> |
```

# BENEFÍCIOS

- Mais tempo para praticar o conceito de funções
- Mais tempo para conhecer as particularidades da linguagem de programação sendo utilizada
- Conexão com habilidades e conhecimentos pré existentes:
  - Os estudantes já foram previamente expostos ao conceito de funções matemáticas
- Permite trabalhar melhor tarefas cognitivas relacionadas ao desenvolvimento de programas
  - desenho de soluções, legibilidade, reutilização)

# OUTROS BENEFÍCIOS

Noções de entrada e saída de um algoritmo são tratadas conceitualmente e não confundidas com entrada e saída via console (ou por outros meios).

Podemos praticar o desenvolvimento e chamada e funções incrementalmente, sem a preocupação de escrever um programa completo.

Funções pequenas são mais simples de tester e “*debugar*”.



# CURSO DE C# SEMANA A SEMANA

1	O que é programação; Funções	7	Comando de repetição - while
2	Funções	8	Comando de repetição- for
3	Tipos de Dados, Strings Estrutura de decisão - if	9	Loops aninhados; Matrizes
4	Variáveis e atribuição	10	Dicionários
5	Manipulação de String, tuplas, listas	11	Entrada e saída básica
6	Listas	12	Construção de programas complexos

Material coberto a cada semana de curso. Uma prova intermediária acontece entre as semanas 6 e 7, e as provas finais ocorrem após a semana 12.

# HTTP://LADYBUGCODINGSCHOOL.COM

PRINCIPAL

QUEM SOMOS

METODOLOGIA

ARTIGOS

DICAS

CONTATO

## MATERIAL

- » [Python - Computação 1](#)
- » [Python - Computação 2](#)

## NOTÍCIAS

09 de abril de 2016

### [Pyladies na UFRJ](#)

Dias 12 e 13 de Abril acontece na UFRJ o evento Pyladies, para incentivar mulheres a aprender programação. Mais informações sobre o evento em [AQUI!](#)

06 de janeiro de 2016

### [Metodologia Ladybug](#)



## PYTHON - COMPUTAÇÃO 1

- » 1- Introdução e Função : [Teórica](#) , [Prática \(parte a\)](#) e [Prática \(parte b\)](#) (Atualizado em 11/04/16)
- » 2- Função : [Teórica](#) , [Prática](#) (Atualizado em 15/11/15)
- » 3- Tipos de dados, Strings, Estruturas Condicionais : [Teórica](#) , [Prática](#) (Atualizado em 02/05/16)

# CONCLUSÕES

A ênfase na construção e uso de funções desde o início de um curso introdutório, associado ao uso de uma linguagem ao mesmo tempo simples e poderosa como o Python

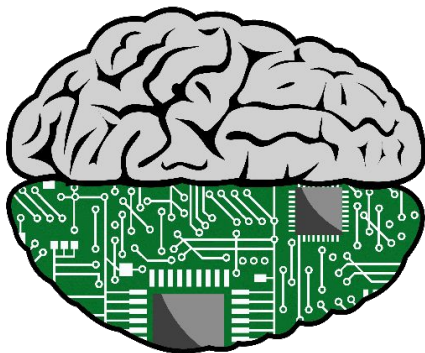
- reduz o tempo e esforço necessários para desenvolver as habilidades de programação;
- proporciona uma base sólida para a futura construção de outras habilidades em estágios futuros
  - Estudo de caso: projeto algpedia

# OUTRAS CONCLUSÕES

Altas taxas de reprovação e abandono, dúvidas quanto à qualidade dos cursos CS1 e CS2 são problemas relacionados à mudanças nos cenários do ensino superior, e a solução está fora do alcance de cada professor isoladamente!

- Mudanças de turma, reprovações e a preservação da continuidade do aprendizado são demandas importantes.
- É necessário repensar os fundamentos destes cursos: **objetivos, conteúdo, linguagem de programação, estratégias de ensino e material de referência.**

# COLLABORATIVE ENCYCLOPEDIA OF ALGORITHMS



## AlgPedia

<http://algpedia.dcc.ufrj.br/>

**Thais N. Vianna, Carla A. D. M. Delgado**

*Programa de Pós Graduação em Informática PPGI - UFRJ*

**João Carlos P. Silva, Carlos Eduardo da S. Martins,  
Fernando R. Gouvêa**

*Departamento de Ciência da Computação - UFRJ*

# PYTHON + DJANGO

## Sorting Algorithms

About: Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to insertion sort.[1] It can be practical if the input is usually in sorted order but may occasionally have some out-of-order elements nearly in position. [More Information]

Implementations : 

PYTHON 

```
def bubbleSort(alist):
    for passnum in range(len(alist)-1,0,-1):
        for i in range(passnum):
            if alist[i]>alist[i+1]:
                temp = alist[i]
                alist[i] = alist[i+1]
                alist[i+1] = temp
```



AlgPedia's goal is to become a reference for programmers, providing relevant information about algorithms and their implementations.



# TRABALHOS FUTUROS

Analisar as notas dos estudantes antes e depois desta abordagem ser adotada.

Nossa análise preliminar revelou um fenômeno interessante: a polarização das notas foi atenuada, o que sugere que a curva de aprendizado esteja mais suave.

***Obrigada pela atenção***