



UFRPE

Universidade
Federal Rural
de Pernambuco



Contrato Pedagógico

Prof. Wyllyams Barbosa Santos

wyllyamss@gmail.com

Introdução à Programação

Agenda

- **Um passo à frente**
- **Objetivos**
- **Ementa**
- **Bibliografia**
- **Metodologia**
- **Avaliações**
- **Responsabilidades**
- **Assinatura do Contrato**
- **Calendário**



UFRPE

Universidade
Federal Rural
de Pernambuco



**“Um passo à frente e você não
está mais no mesmo lugar”**



Um passo à Frente

1º
Semestre

- **Introdução à Programação**

2º
Semestre

- **Laboratório de Programação**
- **Algoritmos e Estruturas de Dados**

3º
Semestre

- **Modelagem e Programação Orientada a Objetos**
- **Fundamentos de Engenharia de Software**

4º
Semestre

- **Processo de Desenvolvimento de Software**
- **Fundamentos de Banco de Dados**

5º
Semestre

- **Gerência de Projetos de Software**
- **Projeto de Banco de Dados**

Conhecer os fundamentos e métodos para construção de algoritmos e programas.



- Algoritmo e linguagens
 - Algoritmo (fluxograma).
 - Instruções de um algoritmo.
 - Tipos de linguagens. (máquina, simbólica, abstrata).
 - Programa, compilação, interpretação e execução.
- Ambiente de programação
 - Conceitos básicos de uma linguagem: variáveis, tipos, estruturas de controle.
 - Declaração de tipo simples.
 - Comandos de entrada e saída.
 - Comando de atribuição.
 - Operadores e expressões.
 - Exercícios em laboratório.

- Tipos estruturados
 - Listas
 - Filas
 - Array unidimensional (vetor).
 - Array bidimensional (matriz).
 - Arquivos.
 - Exercícios em laboratórios
- Subprogramas
 - Função com parâmetro.
 - Sub-routine com parâmetro.
 - Passagem de parâmetro.
 - Exercício em laboratório

- Algoritmos
 - Pesquisa sequencial e binária
 - bubblesort
 - inserção
 - shellsort
 - heapsort
 - quicksort.
- Noções de Complexidade e Paradigmas de Programação
 - Notação o , O , Ω e Θ .
 - Introdução a programação orientada a eventos.
 - Introdução a programação orientada a objetos.
 - Introdução a programação orientada a aspectos.
- Projeto de Implementação.

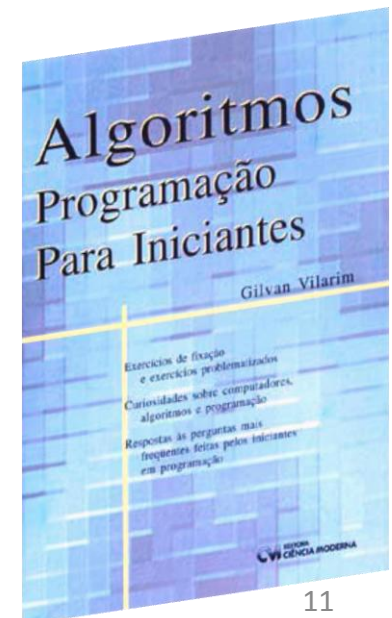
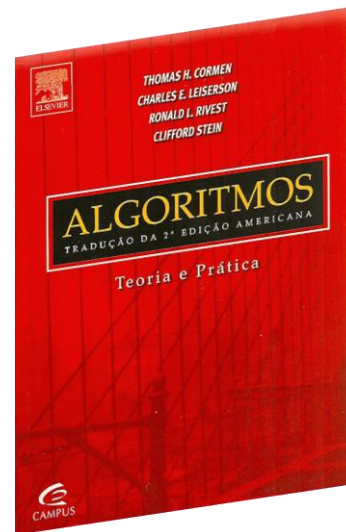
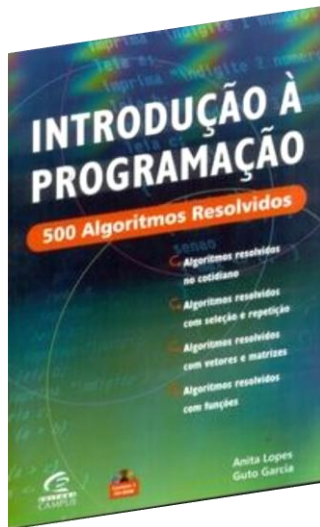
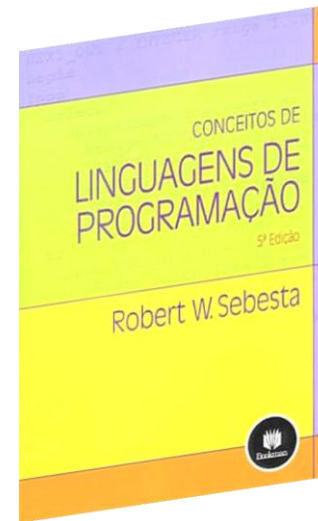
Bibliografia Básica

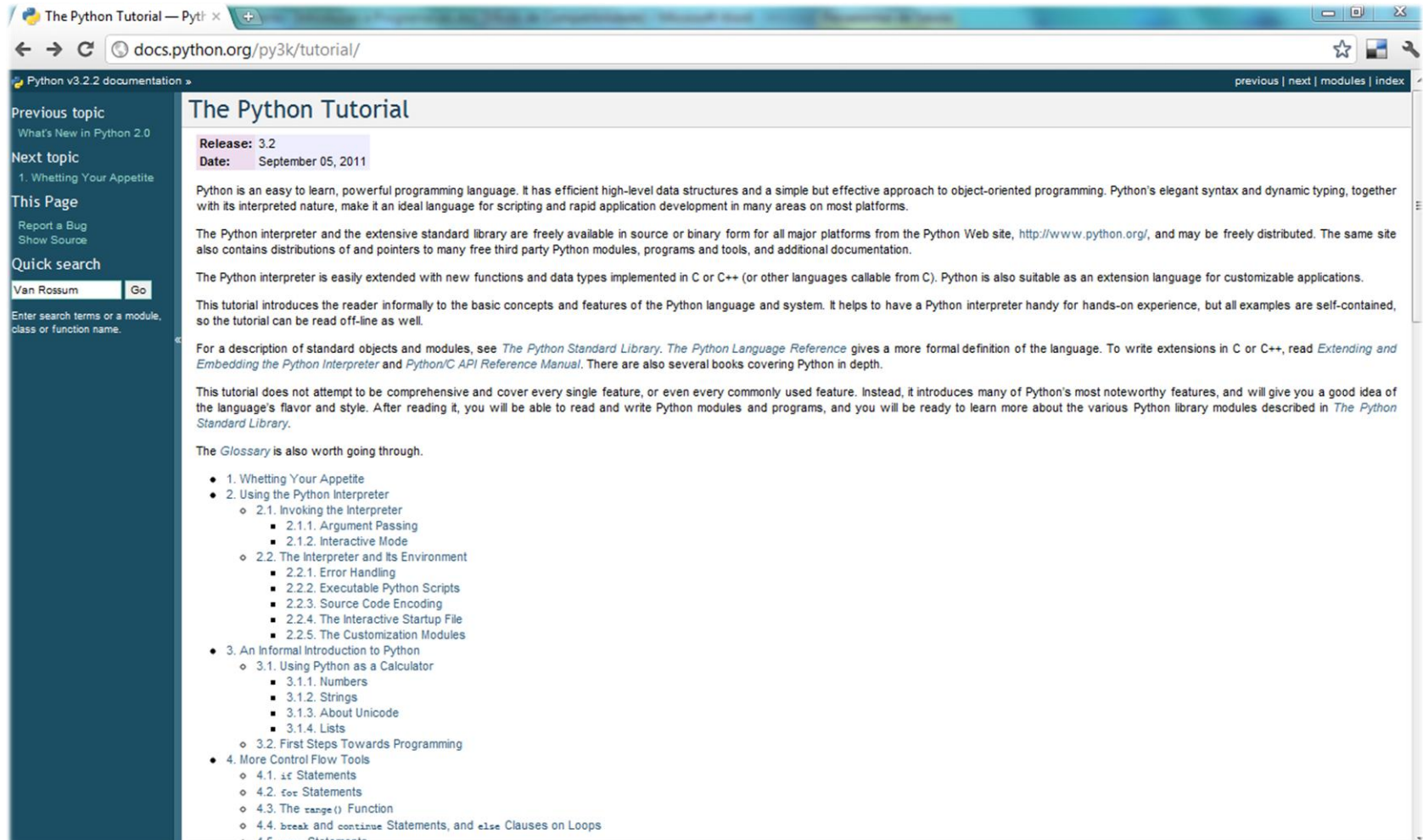
- Lopes, Anita e Garcia, Guto. Introdução a Programação. Campus, 2002.
- Cormen, Thomas H. et. al. Algoritmos: Teoria e Prática. Campus, 2002.
- Sebesta, Robert W. Conceitos de Linguagens de Programação. Bookman, 2001.

Bibliografia Complementar

- Van Rossum, Guido. Tutorial de Python. Disponível gratuitamente em <http://python.org/>, 2004.
- Ziviani, Nivio. Projeto de Algoritmos. Nova Fronteira, 2004.
- Nieto, T. R. Internet & World Wide Web. Como Programar. Bookman, 2003.
- Deitel, Harvey M. et. al. XML Como Programar. Bookman, 2003.
- Farrer, Harry. Algoritmos estruturados. LTC, 1996.
- Vilarim, Gilvan. Algoritmos – Programação para iniciantes. Ciência Moderna, 2004.

Bibliografia





The screenshot shows a web browser window displaying the Python 3.2.2 documentation page. The browser's address bar shows the URL `docs.python.org/py3k/tutorial/`. The page title is "The Python Tutorial".

Release: 3.2
Date: September 05, 2011

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <http://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see *The Python Standard Library*. *The Python Language Reference* gives a more formal definition of the language. To write extensions in C or C++, read *Extending and Embedding the Python Interpreter* and *Python/C API Reference Manual*. There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in *The Python Standard Library*.

The *Glossary* is also worth going through.

- 1. Whetting Your Appetite
- 2. Using the Python Interpreter
 - 2.1. Invoking the Interpreter
 - 2.1.1. Argument Passing
 - 2.1.2. Interactive Mode
 - 2.2. The Interpreter and its Environment
 - 2.2.1. Error Handling
 - 2.2.2. Executable Python Scripts
 - 2.2.3. Source Code Encoding
 - 2.2.4. The Interactive Startup File
 - 2.2.5. The Customization Modules
- 3. An Informal Introduction to Python
 - 3.1. Using Python as a Calculator
 - 3.1.1. Numbers
 - 3.1.2. Strings
 - 3.1.3. About Unicode
 - 3.1.4. Lists
 - 3.2. First Steps Towards Programming
- 4. More Control Flow Tools
 - 4.1. `if` Statements
 - 4.2. `for` Statements
 - 4.3. The `range()` Function
 - 4.4. `break` and `continue` Statements, and `else` Clauses on Loops
 - 4.5. `pass` Statements

Metodologia

Teoria

+

Prática



- 1ª Avaliação
 - Exercícios (0,0 – 3,0)
 - Prova (0,0 – 7,0)
- 2ª avaliação
 - Projeto (0,0 – 5,0)
 - Prova (0,0 – 10,0)
- 3ª avaliação
 - Prova (0,0 – 10,0)
- Final
 - Prova (0,0 – 10,0)

Re **professor** **ponsabi** **alunos** **idade**



“Mediador...

...Provocador!

“A ação de aprender...

não é passiva”!



- Repositório da disciplina: *www.bsi.ufrpe.br*
- Grupo: *ufrpe_intro_prog@googlegroups.com*
- Faltas
- Prazos



Calendário

SEG 15/08	Sem Professor	SEG 17/10	AULA
QUI 17/08	Sem Professor	QUI 19/10	AULA
SEG 22/08	Sem Professor	SEG 24/10	AULA
QUI 24/08	Sem Professor	QUI 26/10	AULA
SEG 29/08	Sem Professor	SEG 31/10	AULA
QUI 31/08	Sem Professor	QUI 02/11	AULA
SEG 05/09	Sem Professor	SEG 07/11	AULA
QUI 07/09	Sem Professor	QUI 09/11	AULA
SEG 12/09	Sem Professor	SEG 14/11	AULA
QUI 14/09	Sem Professor	QUI 16/11	AULA
SEG 19/09	AULA	SEG 21/11	AULA
QUI 21/09	AULA	QUI 23/11	AULA
SEG 26/09	AULA	SEG 28/11	AULA
QUI 28/09	AULA	QUI 30/11	APRESENTAÇÃO DE PROJETOS
SEG 03/10	AULA	SEG 05/12	APRESENTAÇÃO DE PROJETOS
QUI 05/10	AULA APRESENTAÇÃO DE EXERCÍCIO	QUI 07/12	APRESENTAÇÃO DE PROJETOS
SEG 10/10	AULA	SEG 12/12	APRESENTAÇÃO DE PROJETOS
QUI 12/10	PROVA	QUI 14/12	PROVA FINAL



UFRPE
Universidade
Federal Rural
de Pernambuco

