Algoritmos e Programação de Computadores

Introdução & Plano de Desenvolvimento

Profa. Sandra Avila
Instituto de Computação (IC/Unicamp)

MC102 Turma KLMN, 28 Fevereiro, 2018

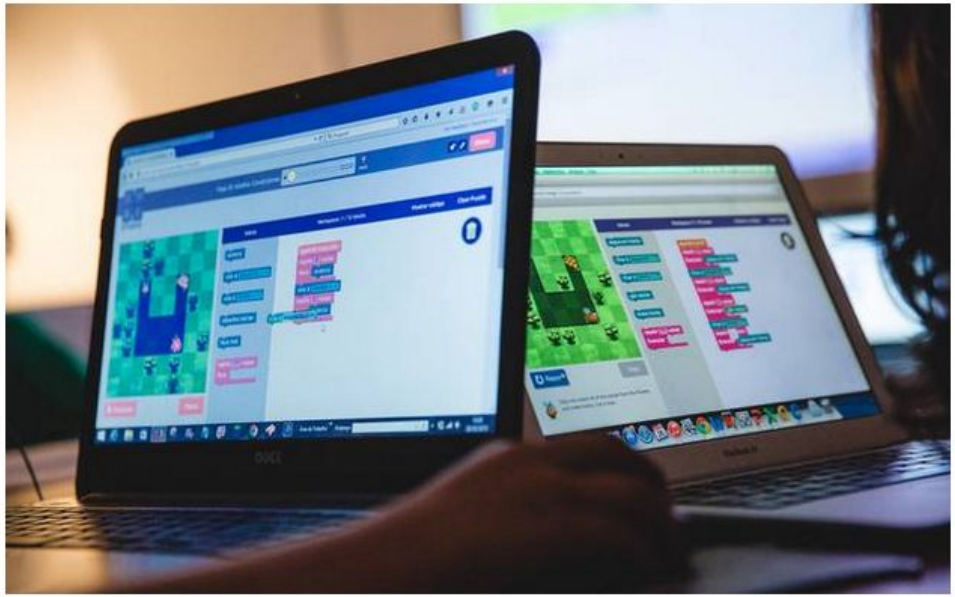
EXPRESSO

Por que programar é o novo 'aprender inglês'

Beatriz Montesanti 02 Abr 2017 (atualizado 04/Abr 18h17)

Escolas no Brasil e no exterior começam a adotar programação entre as disciplinas do currículo

FOTO: OLABI MAKERSPACE/FLICKR / CREATIVE COMMONS



UMA A CADA QUATRO ESCOLAS AMERICANAS OFERECEM AULAS DE PROGRAMAÇÃO

GRÁFICOS

CARREIRA - VOCÊ S/A

Por que aprender programação é tão crucial quanto saber ler

Pesquisador do MIT Media Lab defende que aprender programação é importante para qualquer profissional

Por **Rafael Carvalho**
21 ago 2015, 11h00

Programação: caminho para tornar as pessoas fluentes em novas tecnologias, segundo especialista do MIT (Thinkstock/)

Por que é tão importante aprender programação?

Escrito por HostGator Brasil
25 de janeiro de 2018 | Comente

Diferente do que muito gente pensa, você não precisa ser um gênio para aprender a programar. Lembra de quando você não sabia ler? As letras eram como desenhos ou

O que é Programação de Computadores?

Neste curso, vamos aprender a criar **algoritmos** e **programas** para resolver problemas.

Bolo de Chocolate Super Rápido

Ingredientes:

4 colheres de sopa de chocolate em pó
2 colheres de sopa de margarina
3 xícaras de chá de farinha
2 xícaras de chá de açúcar
1 xícara de chá de leite
4 ovos
2 colheres de sopa de fermento em pó (não muito cheias)

Modo de Fazer:

Misture todos os ingredientes no liquidificador e bata por cerca de 4 minutos.
Desligue o liquidificador, acrescente o fermento em pó, misture bem com a colher, e coloque a mistura em uma forma untada.
Leve ao fogo (pré-aquecido) por cerca de 40 minutos. Cubra com brigadeiro e chocolate granulado.

www.diasdesol.com.br

Por que é importante Aprender a Programar?

Por que é Importante?

- Resolução de problemas
- Visão de futuro
- Senso crítico e criatividade
- Mercado de trabalho

Resolução de Problemas

“Usar a programação como ferramenta para encontrar respostas e soluções para os desafios **acrescenta muito na capacidade de raciocínio lógico** — e essa característica é essencial em qualquer profissional.”

Visão de Futuro

“Saber programação significa entender como a tecnologia funciona. Esse conhecimento não só vai expandir a sua visão de futuro, como também **vai torná-lo capaz de trabalhar melhor em meio a tanta inovação**.”

Senso Crítico e Criatividade

“Ao aprender como as tecnologias funcionam, **ganhamos maior senso crítico, liberdade e criatividade**. Isso porque, em vez de se conformar com as aplicações prontas, podemos criá-las para atender às nossas necessidades.”

Mercado de Trabalho

“Todas as habilidades que estão embutidas no aprendizado de programação são exigidas pelas empresas na hora de contratar um profissional:

- Clareza, rapidez e fluidez nos pensamentos
- Raciocínio lógico
- Organização

Um pouco da história

“O primeiro computador”: https://youtu.be/wyZPsCQd7Uo

Canal Nerdologia

Plano de Desenvolvimento

Plano de Desenvolvimento

- Informações Básicas
- Critérios de Avaliação
- Atendimento & Referências

Informações Básicas

- Carga horária da disciplina
- Ementa
- Linguagem de programação
- Divulgação de informações

Carga Horária da Disciplina

- Duração: 90 horas
- Distribuição semanal
  - 4 horas de aulas teóricas
  - 2 horas de aulas de laboratório
- Frequência mínima: 75% (veja o Regimento Geral de Graduação)
  https://www.dac.unicamp.br/portal/graduacao/regimento-geral

Ementa

- Conceitos básicos de organização de computadores
- Construção de **algoritmos** e sua representação em pseudocódigo e linguagens de alto nível
- Desenvolvimento sistemático e **implementação de programas**
- Estruturação, depuração, testes e documentação de programas
- **Resolução de problemas**

Linguagem de Programação

- Python (versão 3)
- Criada por Guido Van Rossum
- Primeiro *release* em 1991
- Comunidade dinâmica
- Muitas bibliotecas e recursos disponíveis

Divulgação de Informações

- Página web da disciplina
  - http://www.ic.unicamp.br/~mc102
- Páginas web específica para a turma KLMN
  - http://www.ic.unicamp.br/~sandra/teaching/2018-1-mc102klmn
- E-mails encaminhados ao endereço fornecido pela DAC
- Fique atenta(o)!

Critérios de Avaliação

- Atividades conceituais
- Tarefas de laboratórios
- Provas teóricas

Atividades Conceituais

- Visam indicar o **grau de compreensão** dos conceitos básicos
- **Questionários Online**:
  - Exercícios de múltipla escolha
  - Podem ser realizados em laboratório ou em casa
- **Tarefas Presenciais**:
  - Realizadas em laboratório, sob supervisão
  - Aguarde divulgação de datas e outras orientações

Atividades Conceituais

- Ferramenta de apoio:
  - **Moodle** (http://www.ggte.unicamp.br/eam)
  - Login: email da DAC
  - Curso: MC102 - 1S2018 - Algoritmos e Programação de Computadores

Atividades Conceituais — Avaliação

- Proposta de n atividades ao longo do semestre
- Nota AC_i, $1 \leqslant i \leqslant n$, será proporcional ao número de questões respondidas pela(o) aluna(o)
- Questionários Online: Peso 1
- Tarefas Presenciais: Peso 3
- Média M_{AC}: média ponderada das atividades conceituais

Tarefas de Laboratório

- Implementação de problemas em Python
- Enunciado disponível na página da disciplina, com orientações, restrições e prazos
- Auxílio ao desenvolvimento durante as aulas de laboratório
- Correção automática
  - Ferramenta de apoio: SuSy
  - **Testes abertos & Testes fechados**

Tarefas de Laboratório — Avaliação

- Proposta de m tarefas
- Nota L_i, $1 \leqslant i \leqslant m$, calculada de acordo com o estipulado no enunciado da tarefa i.
- Peso $LP_i \in \{1, 2, 3\}$
- Média M_L: média ponderada das tarefas de laboratório

Provas Teóricas

Prova	Peso	Data	Horário
P_1	2	20 de Abril	14h–16h
P_2	3	20 de Junho	14h–16h

- Individuais e sem consulta
- Média M_P: média ponderada das provas teóricas

Critérios de Avaliação

- Média ponderada dos elementos:

$$M_{Elem} = (0.6 \times M_P) + (0.3 \times M_L) + (0.1 \times M_{AC})$$

M_P : média ponderada das **provas teóricas**

M_L : média ponderada das **tarefas de laboratório**

M_{AC}: média ponderada das **atividades conceituais**

- Média antes do exame: $M = min(M_{Elem}, M_P, M_L)$

Critérios de Avaliação — Média Final F

Frequência $\geqslant 75\%$:

- $M \geqslant 5$: **aprovação** por nota e frequência com $F = M$
- Se $2.5 \leqslant M < 5$:
  - Exame E no dia 13 de julho das 14h–16h, $F = (M+E)/2$
    - $F \geqslant 5.0$: **aprovação** por nota e frequência
    - $F < 5.0$: reprovação por nota
- Se $M < 2.5$: reprovação por nota com $F = M$

Frequência $< 75\%$: reprovação por frequência com $F = M$

Observações

- Nos **dias de prova** será necessária a apresentação de **documento oficial com foto**.
- Não há possibilidade de troca de horário de provas e/ou do exame final.
- Não haverá reposição de tarefas de laboratório.
- Todas as tarefas são individuais.
- Qualquer **tentativa de fraude** implicará em nota **0.0 (zero)** na disciplina para todas as pessoas envolvidas.
- O sistema SuSy possui detector de plágio.

Apoio

- Atendimento (Dúvidas)
- Material de apoio & Referências bibliográficas

Atendimento

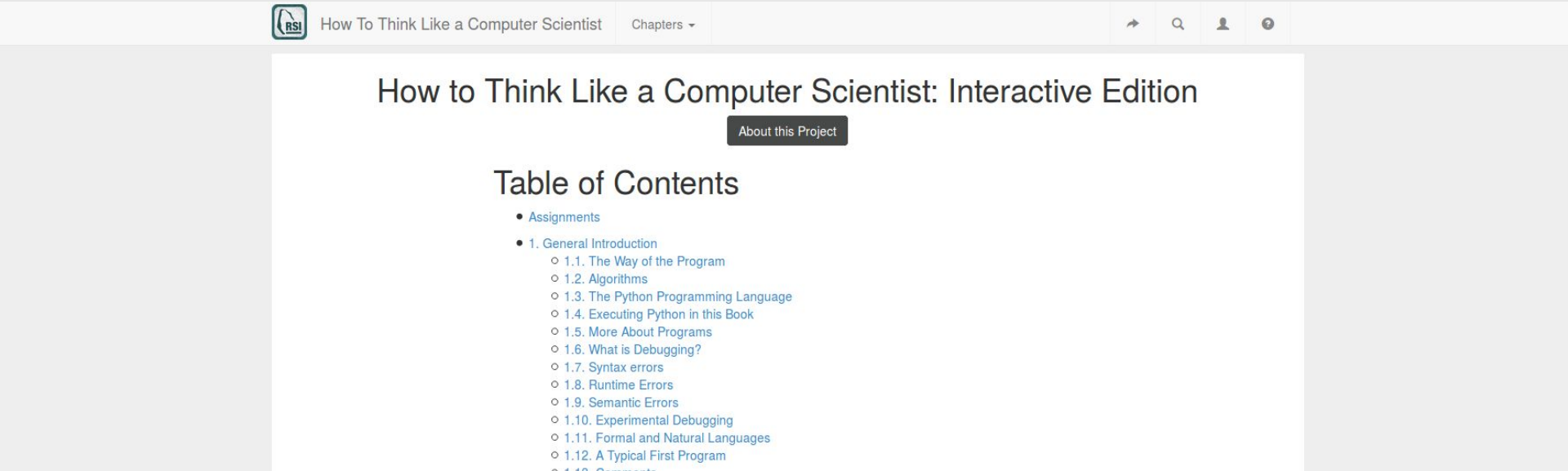
- Equipe de monitoras(es)
  - PEDs: alunas(os) de pós-graduação
  - PADs: alunas(os) de graduação
- Atendimento durante as aulas de laboratório
- Atendimento extra: veja página web da disciplina
- Aulas extras de revisão: veja página web da disciplina

Atendimento

- Equipe
  - Celso Augusto R. L. Brennand (PED, Doutorado)
  - Alceu Emanuel Bissoto (PED, Mestrado)
  - Matheus Martins Susin (PED, Mestrado)
  - Renato Noronha Máximo (PAD, Graduação)
  - William Ryuji Massuda (PAD, Graduação)

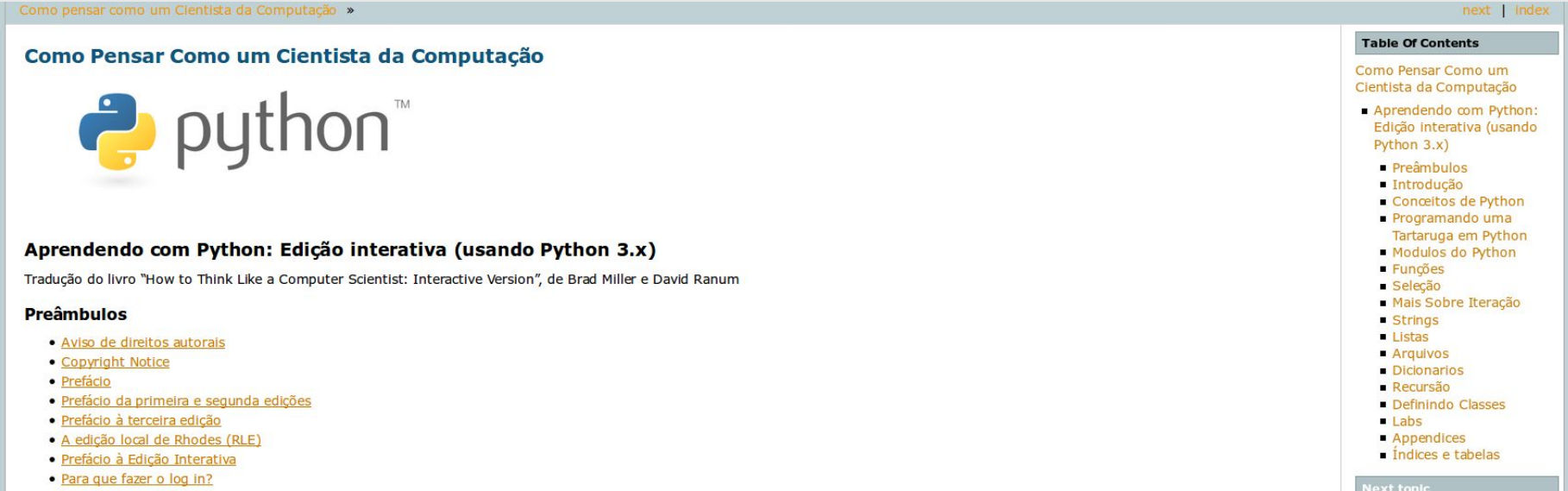
Material de Apoio & Referências Bibliográficas

“How to Think Like a Computer Scientist: Interactive Edition”, de Brad Miller e David Ranum: **https://runestone.academy/runestone/static/thinkcspy/index.html**



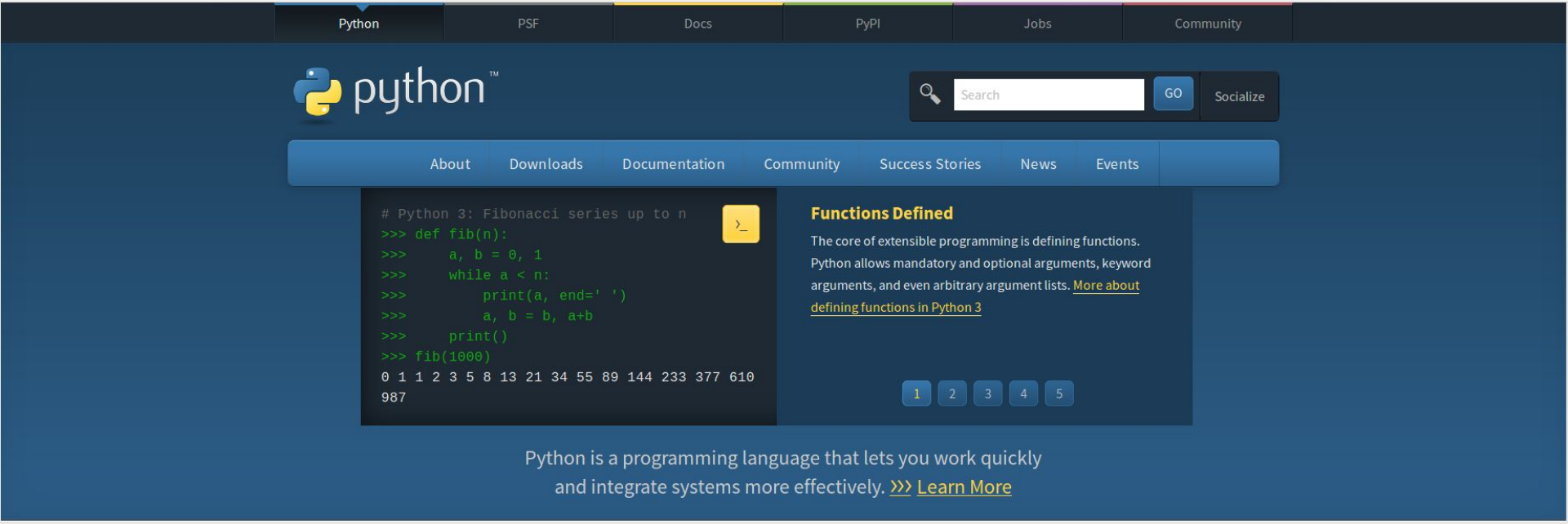
Material de Apoio & Referências Bibliográficas

“How to Think Like a Computer Scientist: Interactive Edition”, de Brad Miller e David Ranum: **https://panda.ime.usp.br/pensepy/static/pensepy**



Material de Apoio & Referências Bibliográficas

Veja material nas páginas oficiais: **www.python.org**



Material de Apoio & Referências Bibliográficas

Veja material nas páginas oficiais: **www.python.org.br**

Programador(a)