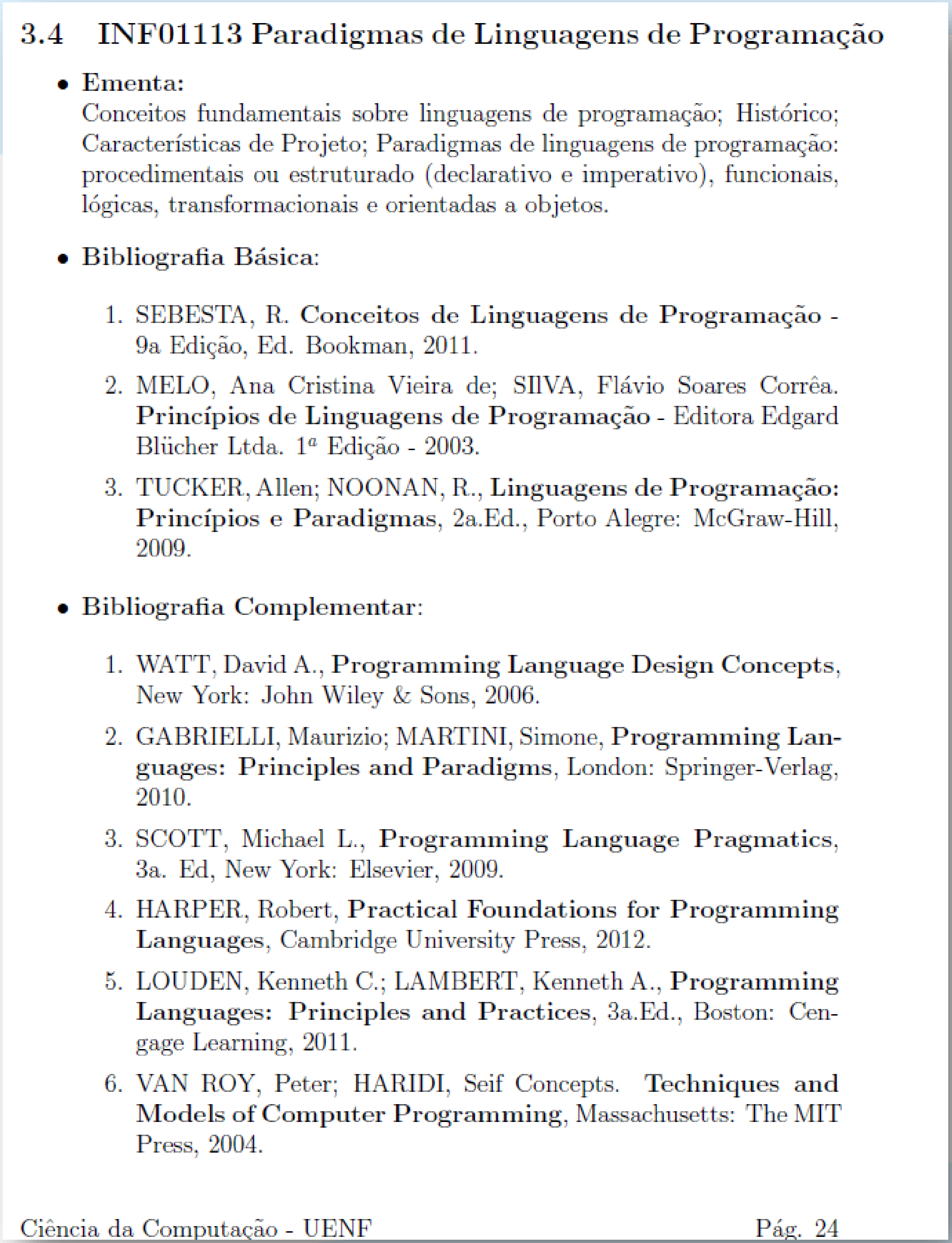
CIÊNCIA DA COMPUTAÇÃO UENF

Paradigmas de Linguagens de Programação

Informações

Prof. Ausberto S. Castro V.
ascv@uenf.br

Orquestra do Paradigmas



3.4 INF01113 Paradigmas de Linguagens de Programação

- **Ementa:**
  Conceitos fundamentais sobre linguagens de programação; Histórico; Características de Projeto; Paradigmas de linguagens de programação: procedimentais ou estruturado (declarativo e imperativo), funcionais, lógicas, transformacionais e orientadas a objetos.

- **Bibliografia Básica:**

  1. SEBESTA, R. **Conceitos de Linguagens de Programação** - 9a Edição, Ed. Bookman, 2011.
  2. MELO, Ana Cristina Vieira de; SIlVA, Flávio Soares Corrêa. **Princípios de Linguagens de Programação** - Editora Edgard Blücher Ltda. 1ª Edição - 2003.
  3. TUCKER, Allen; NOONAN, R., **Linguagens de Programação: Princípios e Paradigmas**, 2a.Ed., Porto Alegre: McGraw-Hill, 2009.

- **Bibliografia Complementar:**

  1. WATT, David A., **Programming Language Design Concepts**, New York: John Wiley & Sons, 2006.
  2. GABRIELLI, Maurizio; MARTINI, Simone, **Programming Languages: Principles and Paradigms**, London: Springer-Verlag, 2010.
  3. SCOTT, Michael L., **Programming Language Pragmatics**, 3a. Ed, New York: Elsevier, 2009.
  4. HARPER, Robert, **Practical Foundations for Programming Languages**, Cambridge University Press, 2012.
  5. LOUDEN, Kenneth C.; LAMBERT, Kenneth A., **Programming Languages: Principles and Practices**, 3a.Ed., Boston: Cengage Learning, 2011.
  6. VAN ROY, Peter; HARIDI, Seif Concepts. **Techniques and Models of Computer Programming**, Massachusetts: The MIT Press, 2004.

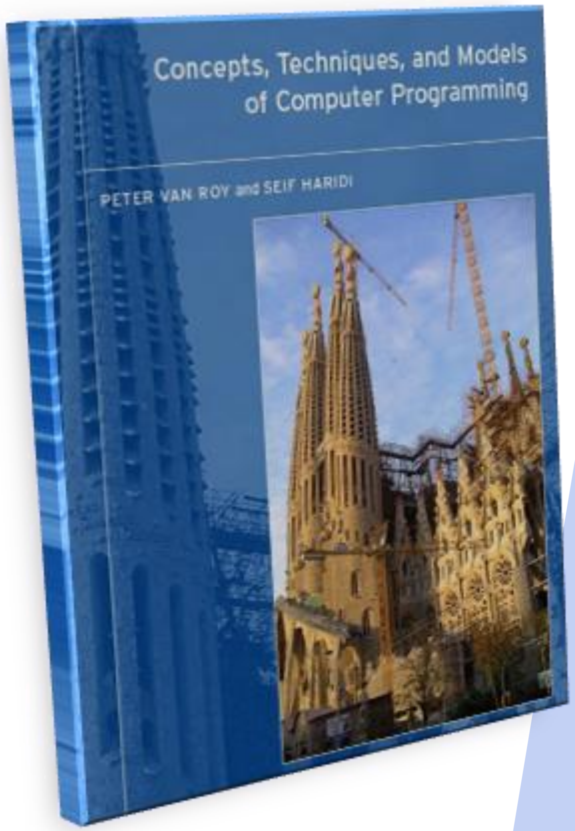
Ciência da Computação - UENF Pág. 24

Ementa da disciplina como aparece no Projeto Pedagógico do Curso

Bibliografia Básica

Bookman; 11 edição
2018

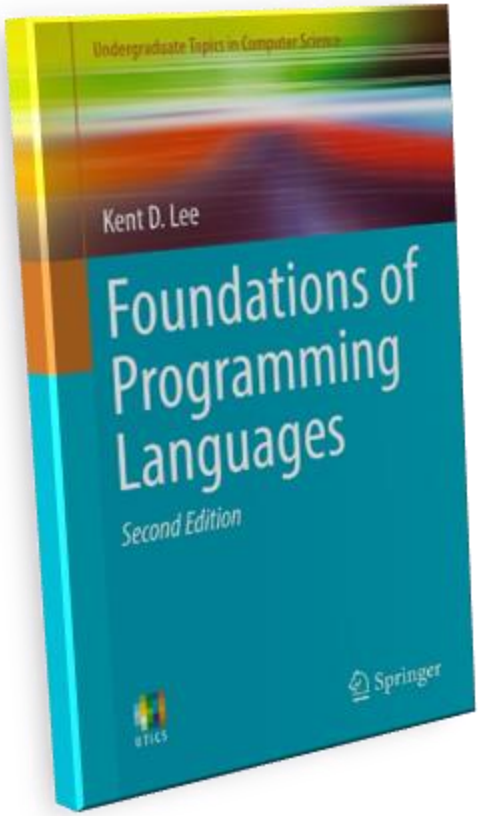
David A. Watt
Programming Language Concepts and Paradigms
PRENTICE HALL INTERNATIONAL SERIES IN COMPUTER SCIENCE
C.A.R. HOARE SERIES EDITOR


Concepts, Techniques, and Models of Computer Programming
PETER VAN ROY and SEIF HARIDI

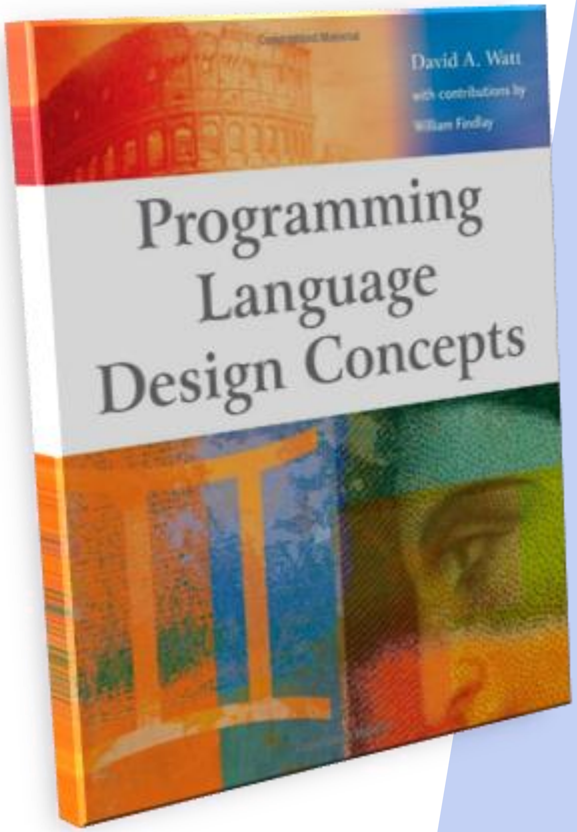
Bibliografia Complementar

Pearson Education; 12 edition
2019

Springer; 2 edition
2017


Undergraduate Topics in Computer Science
Kent D. Lee
Foundations of Programming Languages
Second Edition
Springer
UTICS

Wiley; 1 edition (May 21, 2004)



http://www.levenez.com/lang/

Avaliação

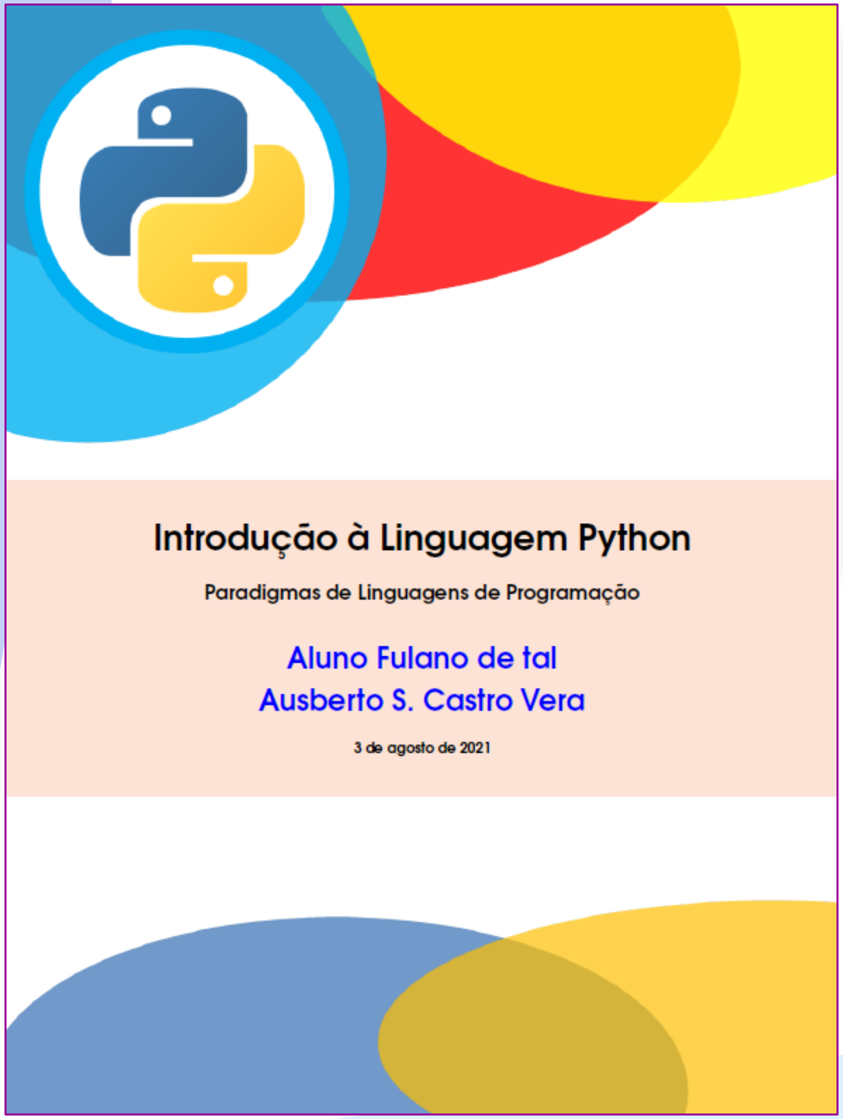
Média:

1. Laboratório (L) (4) **Peso 1**
   - Completo Nota 2.5
   - Incompleto* Nota 1,0
2. Provas Escritas (P) **Peso 2**
   - Pelo menos duas
3. Trabalho Individual (T) **Peso 3**
   - Relatório (LaTeX)

* Incompleto = pelo menos 2/3 do total

Trabalho Individual

- Relatório de uma linguagem de Programação escrito em LaTeX



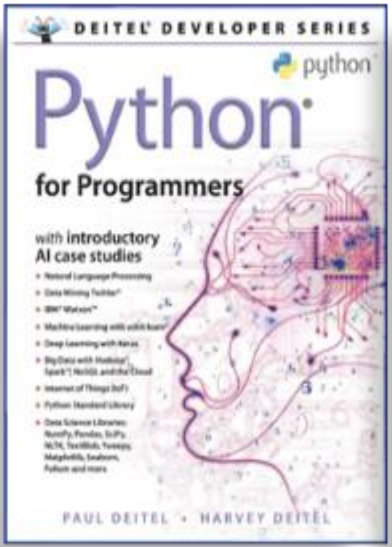
Introdução à Linguagem Python

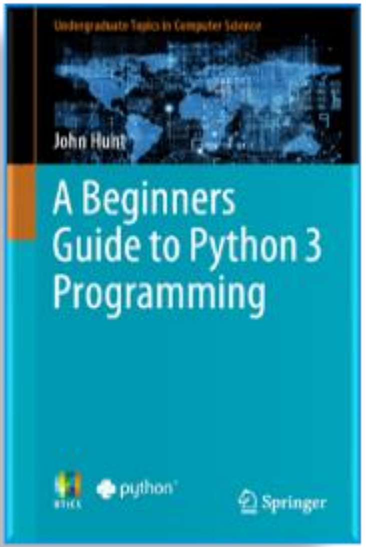
Paradigmas de Linguagens de Programação

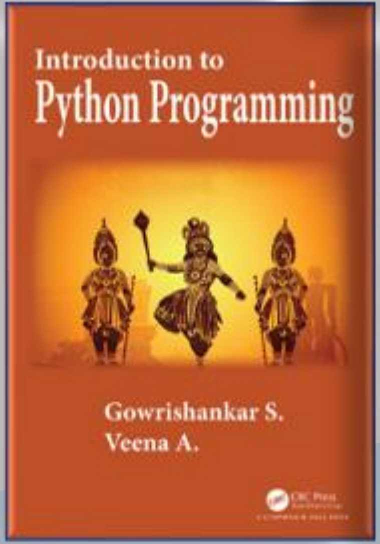
Aluno Fulano de tal

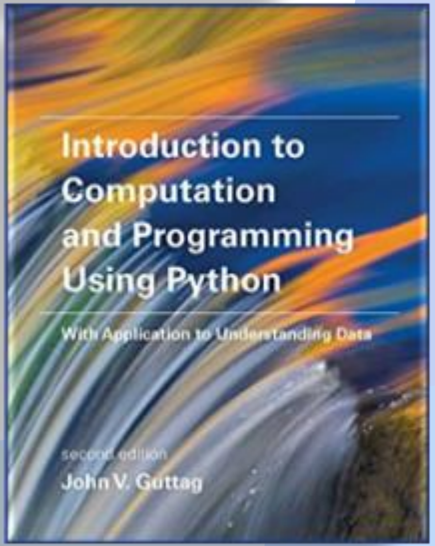
Ausberto S. Castro Vera

3 de agosto de 2021


DEITEL DEVELOPER SERIES
python
Python
for Programmers
with introductory AI case studies
PAUL DEITEL • HARVEY DEITEL

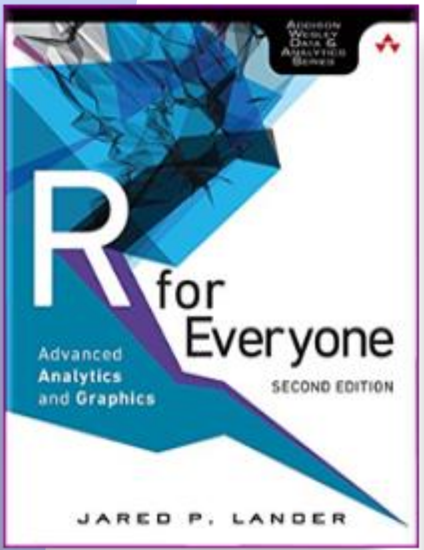

Undergraduate Topics in Computer Science
John Hunt
A Beginners Guide to Python 3 Programming
python
Springer

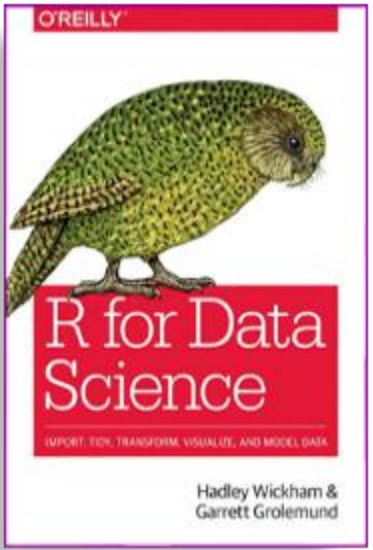


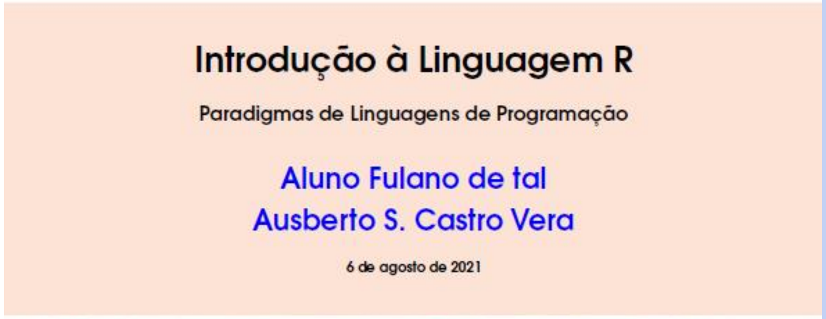

Introduction to Computation and Programming Using Python
With Application to Understanding Data
second edition
John V. Guttag

Trabalho Individual

- Relatório de uma linguagem de Programação escrito em LaTeX


Addison Wesley Data & Analytics Series
R for Everyone
Advanced Analytics and Graphics
SECOND EDITION
JARED P. LANDER


O'REILLY
R for Data Science
IMPORT, TIDY, TRANSFORM, VISUALIZE, AND MODEL DATA
Hadley Wickham & Garrett Grolemund

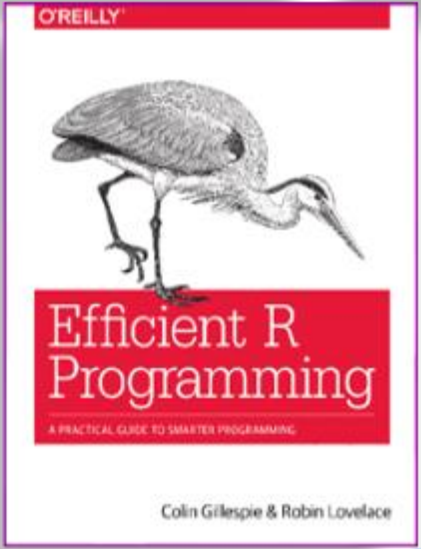


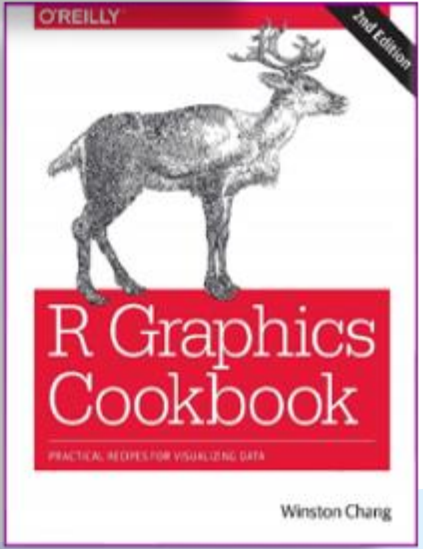
Introdução à Linguagem R

Paradigmas de Linguagens de Programação

Aluno Fulano de tal
Ausberto S. Castro Vera

6 de agosto de 2021


O'REILLY
Efficient R Programming
A PRACTICAL GUIDE TO SMARTER PROGRAMMING
Colin Gillespie & Robin Lovelace



Trabalho Individual

- Relatório de uma linguagem de Programação escrito em LaTeX

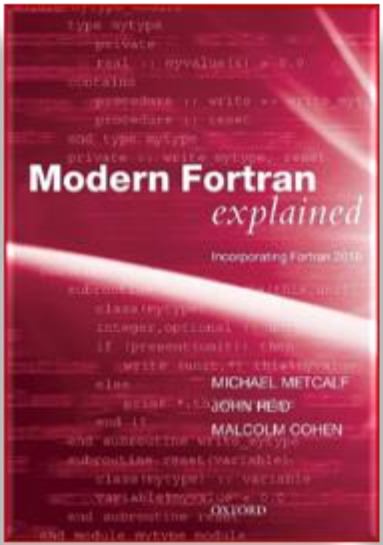
Introdução à Linguagem FORTRAN

Paradigmas de Linguagens de Programação

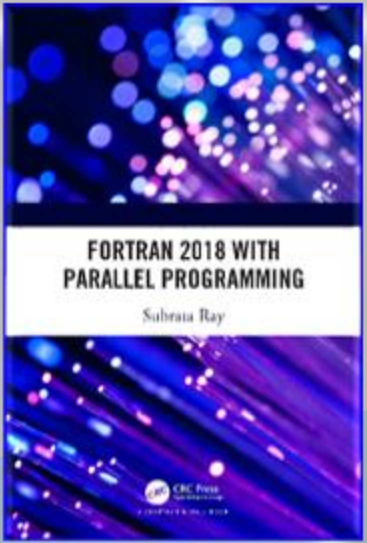
Aluno Fulano de tal

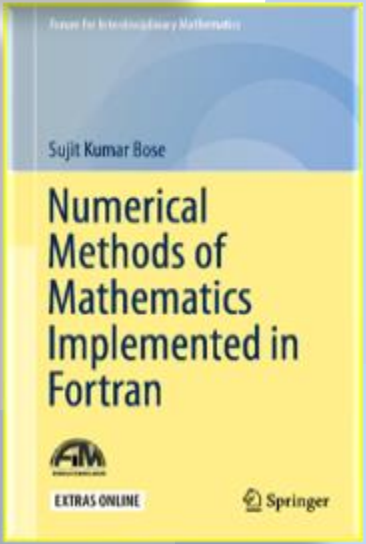
Ausberto S. Castro Vera

4 de julho de 2021



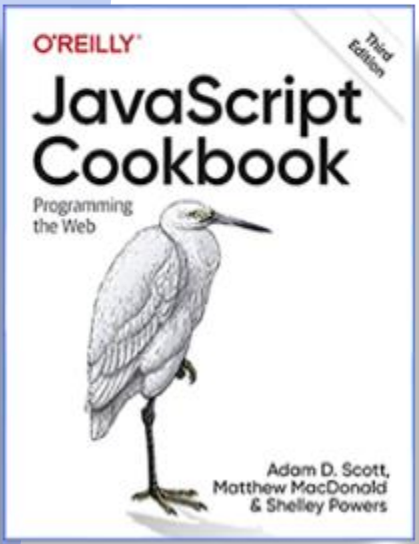
Ian Chivers · Jane Sleightholme
Introduction to Programming with Fortran
Fourth Edition
Springer



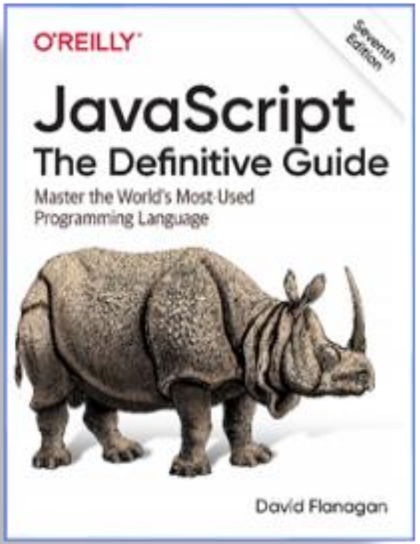


Trabalho Individual

- **Relatório de uma linguagem de Programação escrito em LaTeX**

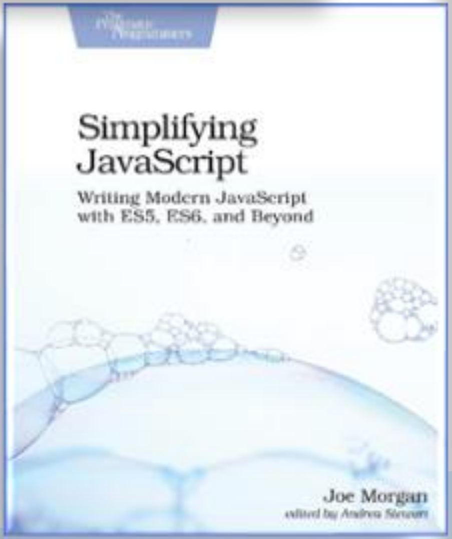


O'REILLY
Third Edition
JavaScript Cookbook
Programming the Web
Adam D. Scott, Matthew MacDonald & Shelley Powers



O'REILLY
Seventh Edition
JavaScript
The Definitive Guide
Master the World's Most-Used Programming Language
David Flanagan

O'REILLY
JavaScript Everywhere
Building Cross-Platform Applications with GraphQL, React, React Native, and Electron
Adam D. Scott



Simplifying JavaScript
Writing Modern JavaScript with ES5, ES6, and Beyond
Joe Morgan

Introdução à Linguagem JavaScript
Paradigmas de Linguagens de Programação
Aluno Fulano de tal
Ausberto S. Castro Vera
4 de julho de 2021

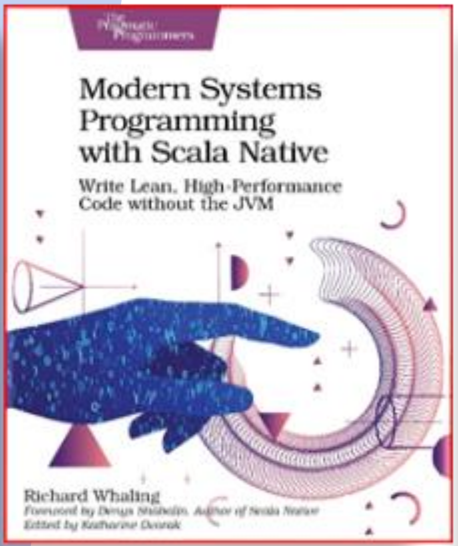
Trabalho Individual – datas importantes

- **Introdução + Bibliografia**
  - Autor, datas, motivações, livros e artigos (docs PDF)
  - NÃO considerar páginas web
  - Entrega: até 31 de Agosto de 2021
- **Aspectos BÁSICOS da linguagem**
  - Estruturas, comandos, funções, etc.
  - Entrega: até 21 de setembro de 2021
- **Aspectos AVANÇADOS da linguagem**
  - Módulos, funções, objetos, etc.
  - Entrega: até 14 de outubro de 2021
- **Aplicações**
  - 5 aplicações: calculadora, quicksort, gráfico, ...
  - Entrega: até 04 de novembro de 2021
- **Ferramentas e Conclusões**
  - Compiladores, interpretadores, editores, IDEs
  - Entrega FINAL: até 30 de novembro 2021

Entrega completa: 10,0
Entrega incompleta: 5,0

Trabalho Individual - Opcional

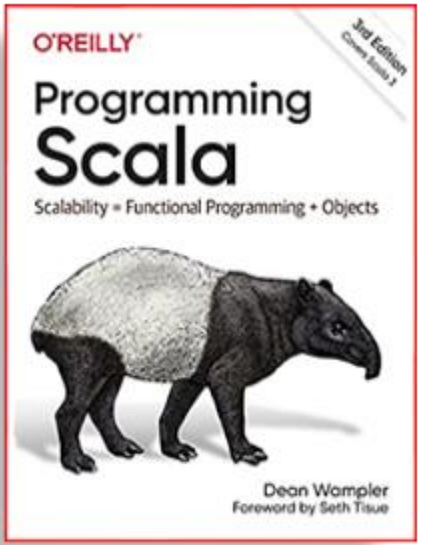
- Relatório de uma linguagem de Programação escrito em LaTeX


The Pragmatic Programmers

Modern Systems Programming with Scala Native

Write Lean, High-Performance Code without the JVM

Richard Whaling


O'REILLY

3rd Edition

Programming Scala

Scalability = Functional Programming + Objects

Dean Wampler

Foreword by Seth Tisue

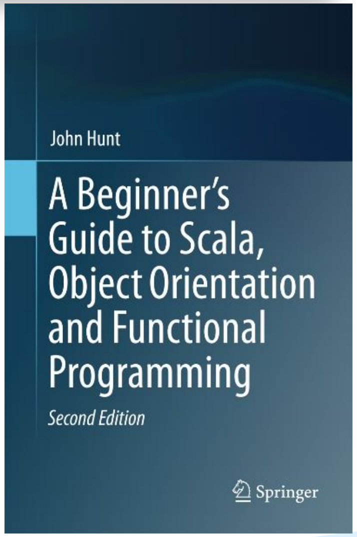
Introdução à Linguagem Scala

Paradigmas de Linguagens de Programação

Aluno Fulano de tal

Ausberto S. Castro Vera

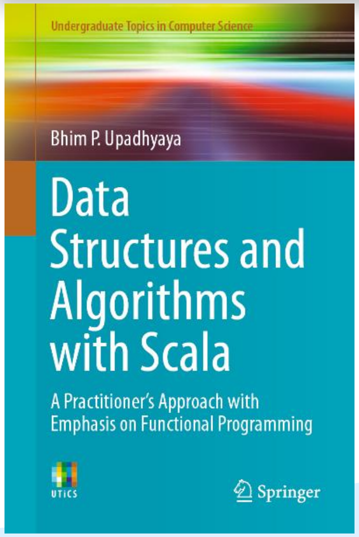
10 de agosto de 2021


John Hunt

A Beginner's Guide to Scala, Object Orientation and Functional Programming

Second Edition

Springer


Undergraduate Topics in Computer Science

Bhim P. Upadhyaya

Data Structures and Algorithms with Scala

A Practitioner's Approach with Emphasis on Functional Programming

UTICS

Springer

Trabalho Individual - Links

https://www.python.org/

https://www.r-project.org/

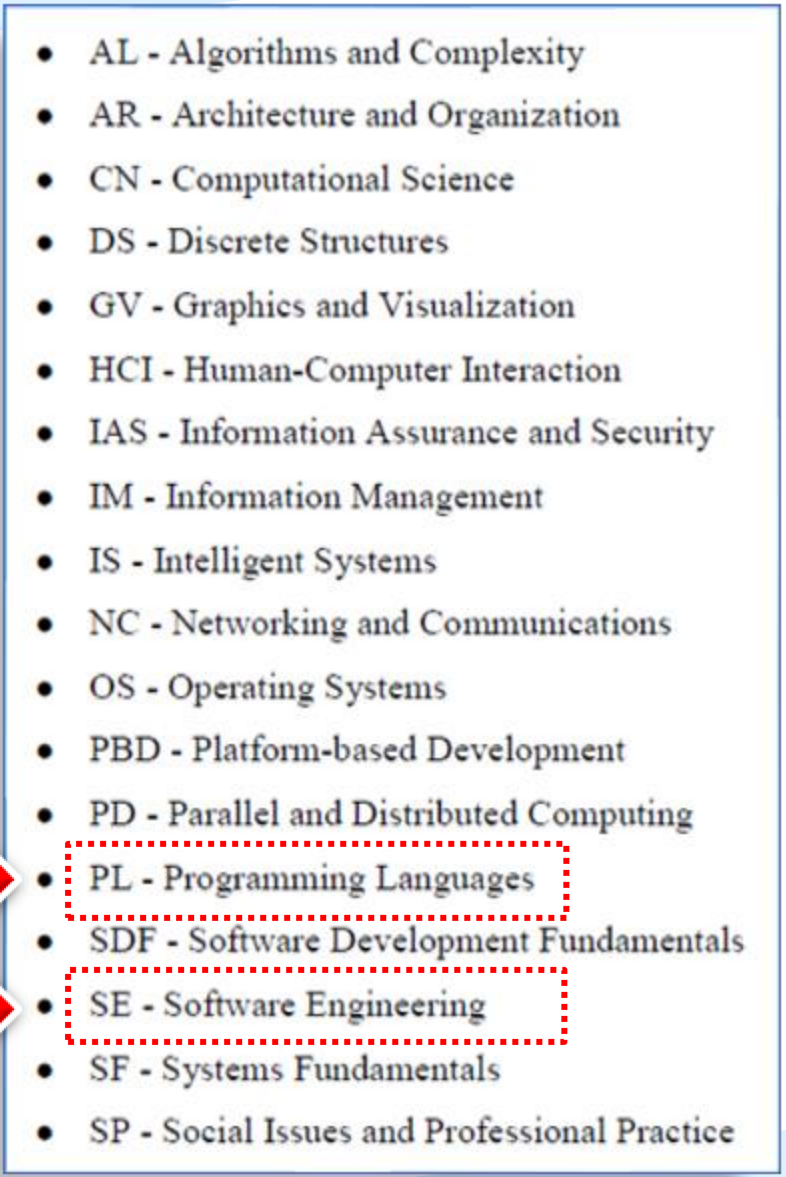
https://www.fortran.com/
https://www.fortrantutorial.com/basics/

https://www.javascript.com/

https://en.wikipedia.org/wiki/JavaScript

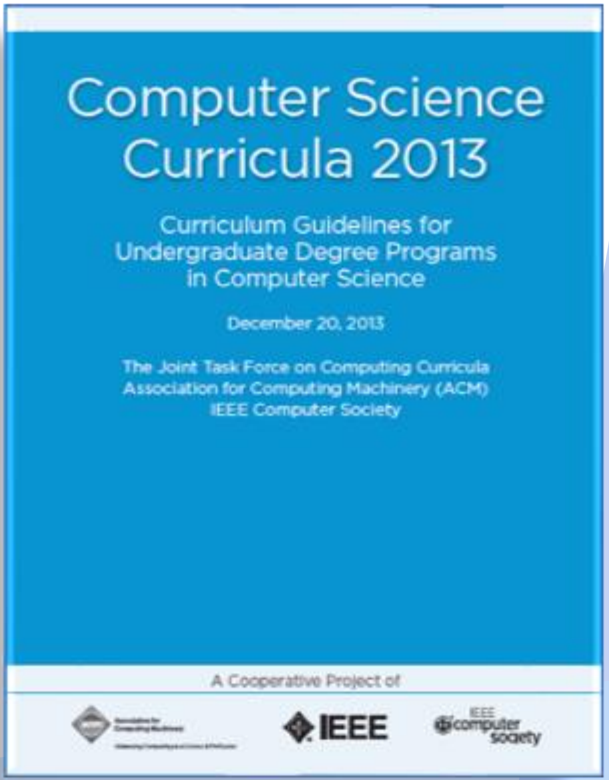
https://www.scala-lang.org/

Ciência da Computação



- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visualization
- HCI - Human-Computer Interaction
- IAS - Information Assurance and Security
- IM - Information Management
- IS - Intelligent Systems
- NC - Networking and Communications
- OS - Operating Systems
- PBD - Platform-based Development
- PD - Parallel and Distributed Computing
- PL - Programming Languages
- SDF - Software Development Fundamentals
- SE - Software Engineering
- SF - Systems Fundamentals
- SP - Social Issues and Professional Practice

18 ÁREAS



Computer Science Curricula 2013

Curriculum Guidelines for Undergraduate Degree Programs in Computer Science

December 20, 2013

The Joint Task Force on Computing Curricula
Association for Computing Machinery (ACM)
IEEE Computer Society

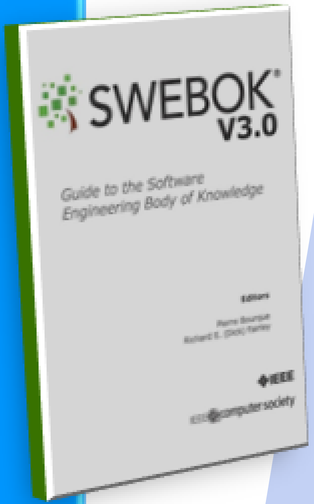
A Cooperative Project of

PL – Programming Languages

- PL/Object-Oriented Programming
- PL/Functional Programming
- PL/Event-Driven and Reactive Programming
- PL/Basic Type Systems
- PL/Program Representation
- PL/Language Translation and Execution
- PL/Syntax Analysis
- PL/Compiler Semantic Analysis
- PL/Code Generation
- PL/Runtime Systems
- PL/Static Analysis
- PL/Advanced Programming Constructs
- PL/Concurrency and Parallelism
- PL/Type Systems
- PL/Formal Semantics
- PL/Language Pragmatics
- PL/Logic Programming

SE-Engenharia de Software

1. Software Requirements
2. Software Design
3. Software Construction
4. Software Testing
5. Software Maintenance
6. Software Configuration Management
7. Software Engineering Management
8. Software Engineering Process
9. Software Engineering Models and Methods
10. Software Quality
11. Software Engineering Professional Practice
12. Software Engineering Economics
13. Computing Foundations
14. Mathematical Foundations
15. Engineering Foundations



Chapter 13: Computing Foundations

- **1. Problem Solving Techniques**
  - 1.1. Definition of Problem Solving
  - 1.2. Formulating the Real Problem
  - 1.3. Analyze the Problem
  - 1.4. Design a Solution Search Strategy
  - 1.5. Problem Solving Using Programs
- **2. Abstraction**
  - 2.1. Levels of Abstraction
  - 2.2. Encapsulation
  - 2.3. Hierarchy
  - 2.4. Alternate Abstractions
- **3. Programming Fundamentals**
  - 3.1. The Programming Process
  - 3.2. Programming Paradigms
- **4. Programming Language Basics**
  - 4.1. Programming Language Overview
  - 4.2. Syntax and Semantics of Programming Languages
  - 4.3. Low-Level Programming Languages
  - 4.4. High-Level Programming Languages
    4.5. Declarative vs. Imperative Programming Languages
- **5. Debugging Tools and Techniques**
  - 5.1. Types of Errors
  - 5.2. Debugging Techniques
  - 5.3. Debugging Tools
- **6. Data Structure and Representation**
  - 6.1. Data Structure Overview
  - 6.2. Types of Data Structure
  - 6.3. Operations on Data Structures

Chapter 13: Computing Foundations

- **7. Algorithms and Complexity**
  - 7.1. Overview of Algorithms
  - 7.2. Attributes of Algorithms
  - 7.3. Algorithmic Analysis
  - 7.4. Algorithmic Design Strategies
  - 7.5. Algorithmic Analysis Strategies
- **8. Basic Concept of a System**
  - 8.1. Emergent System Properties
  - 8.2. Systems Engineering
  - 8.3. Overview of a Computer System
- **9. Computer Organization**
  - 9.1. Computer Organization Overview
  - 9.2. Digital Systems
  - 9.3. Digital Logic
  - 9.4. Computer Expression of Data
  - 9.5. The Central Processing Unit (CPU)
  - 9.6. Memory System Organization
  - 9.7. Input and Output (I/O)
- **11. Operating Systems Basics**
  - 11.1. Operating Systems Overview
  - 11.2. Tasks of an Operating System
  - 11.3. Operating System Abstractions
  - 11.4. Operating Systems Classification
- **12. Database Basics and Data Management**
  - 12.1. Entity and Schema
  - 12.2. Database Management Systems (DBMS)
  - 12.3. Database Query Language
  - 12.4. Tasks of DBMS Packages
  - 12.5. Data Management
  - 12.6. Data Mining
- **13. Network Communication Basics**
  - 13.1. Types of Network
  - 13.2. Basic Network Components
  - 13.3. Networking Protocols and Standards
  - 13.4. The Internet
  - 13.5. Internet of Things
  - 13.6. Virtual Private Network (VPN)

Chapter 13: Computing Foundations

- **14. Parallel and Distributed Computing**
  - 14.1. Parallel and Distributed Computing Overview
  - 14.2. Difference between Parallel and Distributed Computing
  - 14.3. Parallel and Distributed Computing Models
  - 14.4. Main Issues in Distributed Computing
- **15. Basic User Human Factors**
  - 15.1. Input and Output
  - 15.2. Error Messages
  - 15.3. Software Robustness
- **16. Basic Developer Human Factors**
  - 16.1. Structure
  - 16.2. Comments
- **17. Secure Software Development and Maintenance**
  - 17.1. Software Requirements Security
  - 17.2. Software Design Security
  - 17.3. Software Construction Security
  - 17.4. Software Testing Security
  - 17.5. Build Security into Software Engineering Process
  - 17.6. Software Security Guidelines

Ciência da Computação 2020

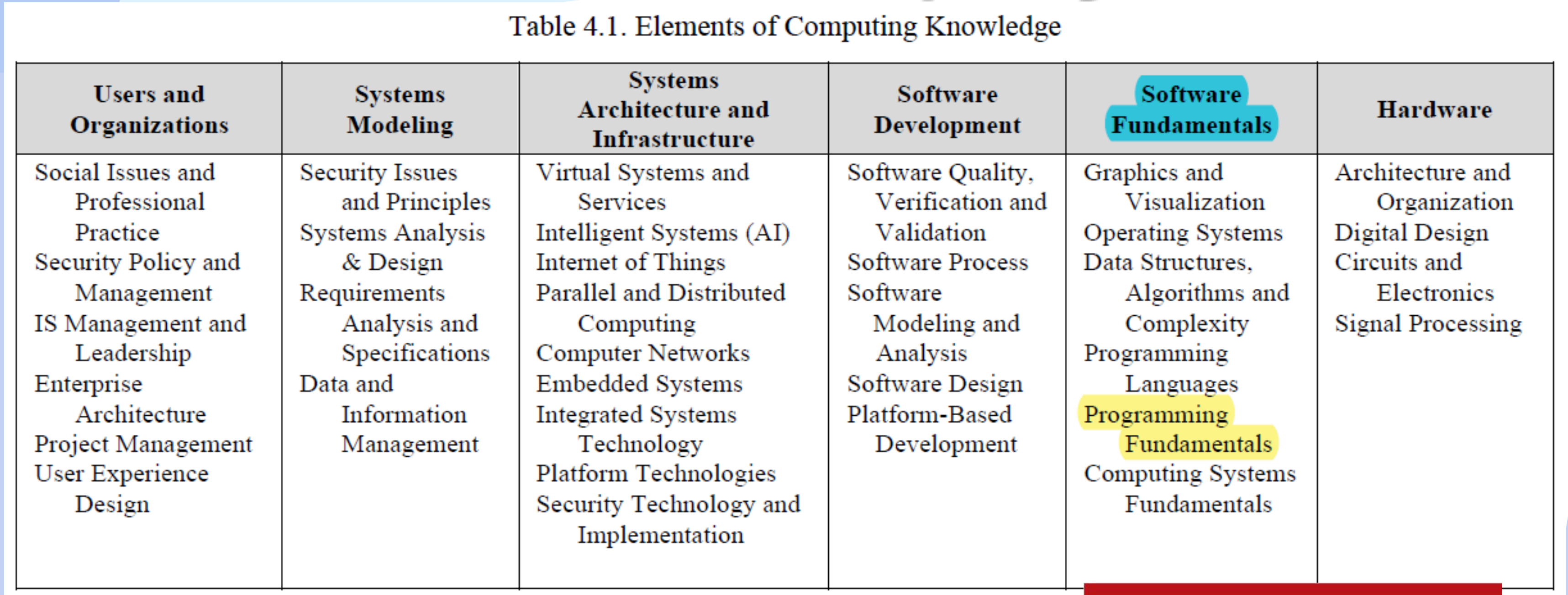
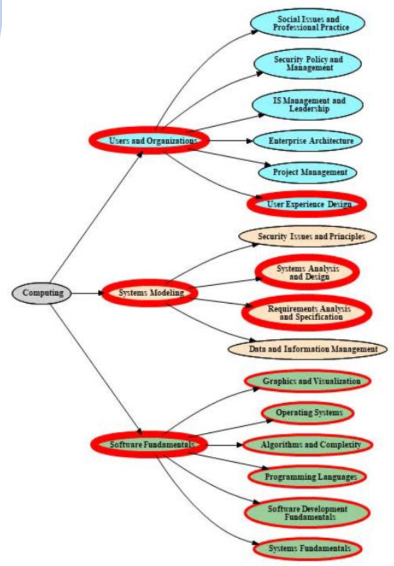
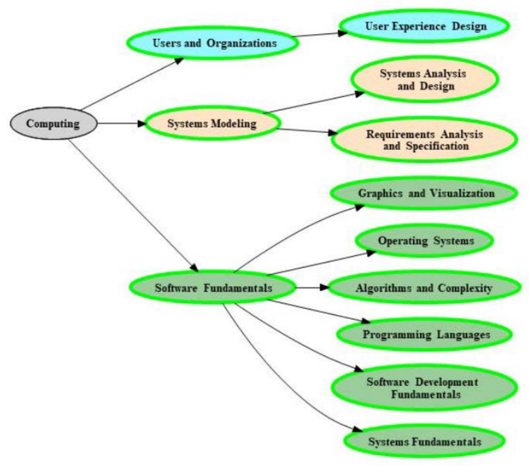
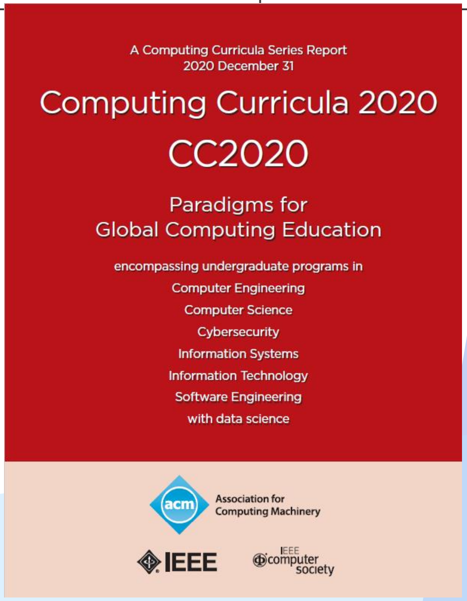


Table 4.1. Elements of Computing Knowledge

Users and Organizations	Systems Modeling	Systems Architecture and Infrastructure	Software Development	Software Fundamentals	Hardware
Social Issues and Professional Practice Security Policy and Management IS Management and Leadership Enterprise Architecture Project Management User Experience Design	Security Issues and Principles Systems Analysis & Design Requirements Analysis and Specifications Data and Information Management	Virtual Systems and Services Intelligent Systems (AI) Internet of Things Parallel and Distributed Computing Computer Networks Embedded Systems Integrated Systems Technology Platform Technologies Security Technology and Implementation	Software Quality, Verification and Validation Software Process Software Modeling and Analysis Software Design Platform-Based Development	Graphics and Visualization Operating Systems Data Structures, Algorithms and Complexity Programming Languages Programming Fundamentals Computing Systems Fundamentals	Architecture and Organization Digital Design Circuits and Electronics Signal Processing


Computing
Users and Organizations
Social Issues and Professional Practice
Security Policy and Management
IS Management and Leadership
Enterprise Architecture
Project Management
User Experience Design
Systems Modeling
Security Issues and Principles
Systems Analysis and Design
Requirements Analysis and Specification
Data and Information Management
Software Fundamentals
Graphics and Visualization
Operating Systems
Algorithms and Complexity
Programming Languages
Software Development Fundamentals
Systems Fundamentals


Computing
Users and Organizations
User Experience Design
Systems Modeling
Systems Analysis and Design
Requirements Analysis and Specification
Software Fundamentals
Graphics and Visualization
Operating Systems
Algorithms and Complexity
Programming Languages
Software Development Fundamentals
Systems Fundamentals



A Computing Curricula Series Report
2020 December 31

Computing Curricula 2020

CC2020

Paradigms for
Global Computing Education

encompassing undergraduate programs in
Computer Engineering
Computer Science
Cybersecurity
Information Systems
Information Technology
Software Engineering
with data science

Association for Computing Machinery

IEEE

IEEE computer society

lifehacker

Five Best Programming Languages for First-Time Learners

Java

RUBY Programming

python

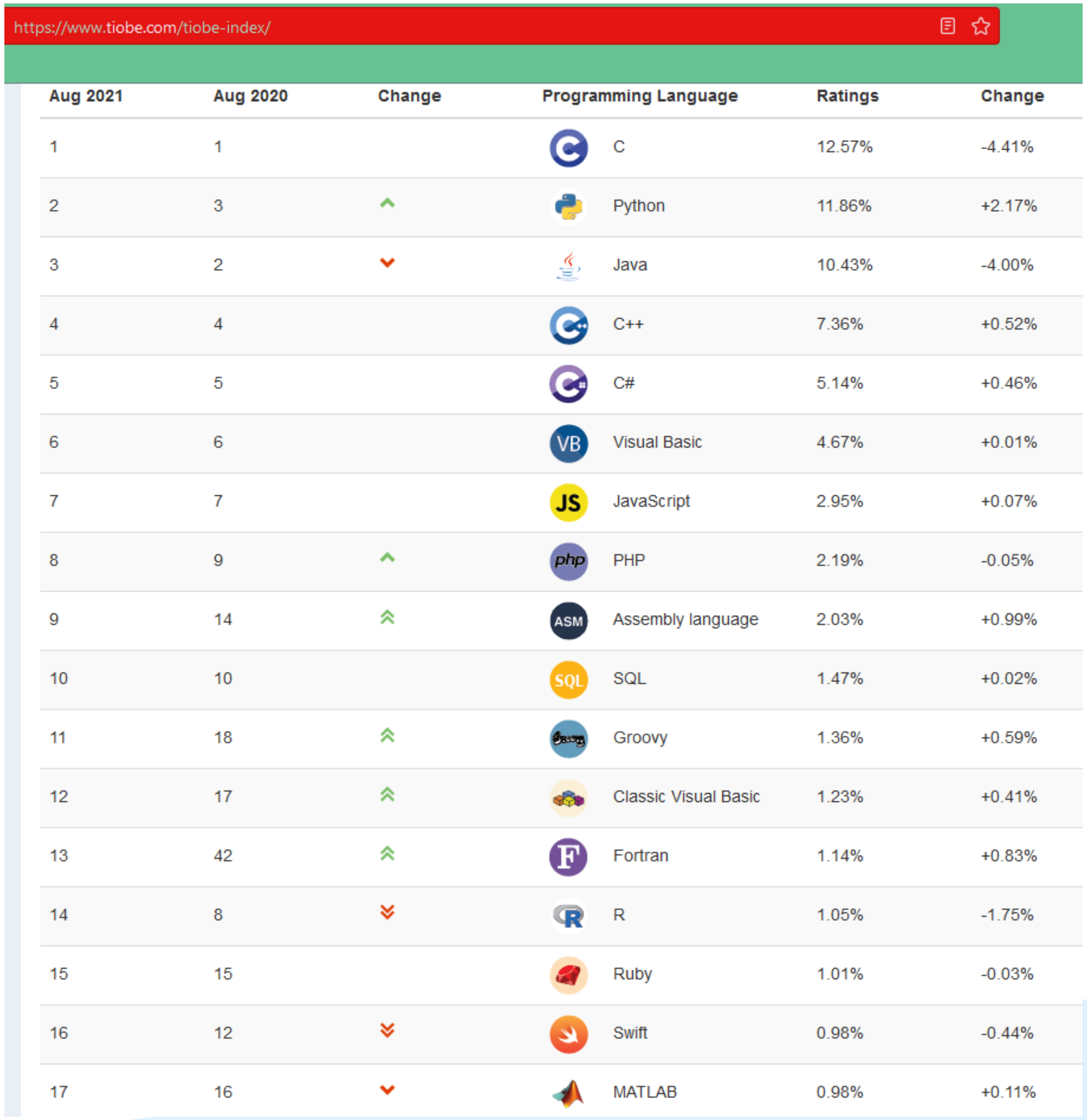
THE C PROGRAMMING LANGUAGE

C++

JavaScript

http://lifehacker.com/five-best-programming-languages-for-first-time-learners-1494256243

TIOBE Index for Agust 2021



https://www.tiobe.com/tiobe-index/

Aug 2021	Aug 2020	Change	Programming Language	Ratings	Change
1	1		C	12.57%	-4.41%
2	3	⌃	Python	11.86%	+2.17%
3	2	⌄	Java	10.43%	-4.00%
4	4		C++	7.36%	+0.52%
5	5		C#	5.14%	+0.46%
6	6		Visual Basic	4.67%	+0.01%
7	7		JavaScript	2.95%	+0.07%
8	9	⌃	PHP	2.19%	-0.05%
9	14	⏫	Assembly language	2.03%	+0.99%
10	10		SQL	1.47%	+0.02%
11	18	⏫	Groovy	1.36%	+0.59%
12	17	⏫	Classic Visual Basic	1.23%	+0.41%
13	42	⏫	Fortran	1.14%	+0.83%
14	8	⏬	R	1.05%	-1.75%
15	15		Ruby	1.01%	-0.03%
16	12	⏬	Swift	0.98%	-0.44%
17	16	⌄	MATLAB	0.98%	+0.11%

The 9 Best Programming Languages to Learn in 2021

1. JavaScript	• + HTML+ CSS
2. Swift	• IOS
3. Scala	• concorrente
4. Go	• Google
5. Python	• User-friendly
6. Elm	• Fast-executing
7. Ruby	• web
8. C#	• Microsoft, VR, Xamarin
9. Rust	• Mozilla

FULLSTACK ACADEMY

https://www.fullstackacademy.com/blog/nine-best-programming-languages-to-learn

FULLSTACK ACADEMY

Which Programming Language to Learn Based on Your Career Goals

Front-end web development	Back-end web development	Mobile development
JavaScript	JavaScript	Swift
Elm	Scala	Java
TypeScript	Python	Objective C
	Go	JavaScript
	Ruby	

Game development	Desktop applications	Systems programming
Unity	Scala	Go
TypeScript	Go	Rust
	Python	

What Programming Languages Engineers and Employers Love—and Hate

IEEE SPECTRUM

Skills	San Francisco Bay Area	New York	Toronto	Paris	London	Developers
1. Go	3	2	8	1	3	7%
2. Scala	4	7	13	14	2	3%
3. Ruby	2	1	2	15	4	10%
4. TypeScript	1	5	1	2	1	12%
5. Kotlin	5	10	12	8	9	2%
6. JavaScript	6	4	3	4	5	62%
7. Objective-C	10	3	10	11	13	3%
8. PHP	7	6	6	3	8	12%
9. Java	9	12	9	7	7	42%
10. HTML	8	9	5	6	6	36%
11. Swift	11	8	14	12	15	6%
12. Python	12	11	7	13	10	42%
13. C++	13	14	15	10	14	14%
14. C	14	15	11	5	12	9%
15. C#	15	13	4	9	11	17%
16. R	16	16	16	16	16	2%

Source: Hired

Top 10 IT Skills and Tech Skills for 2021

1. IT Support
2. Remote IT Jobs
3. Artificial Intelligence (AI)
4. Cybersecurity
5. Project Management
6. Software Development
7. Cloud Computing
8. Augmented Reality (AR) and Virtual Reality (VR)
9. Data Science
10. Business Intelligence (BI) Analysis

technojobs
The IT & Technical Job Site

Qual Linguagem?

2020

1. Mobile Development
2. Artificial Intelligence
3. Python
4. Data Science
5. Cybersecurity
6. Cloud/Amazon Web Services
7. Blockchain
8. Virtual Reality
9. IT Support
10. Internet of Things (IoT)

Top 10 In-Demand Programming languages to learn in 2020

Popularidade, salario, demanda de trabalho, Usos

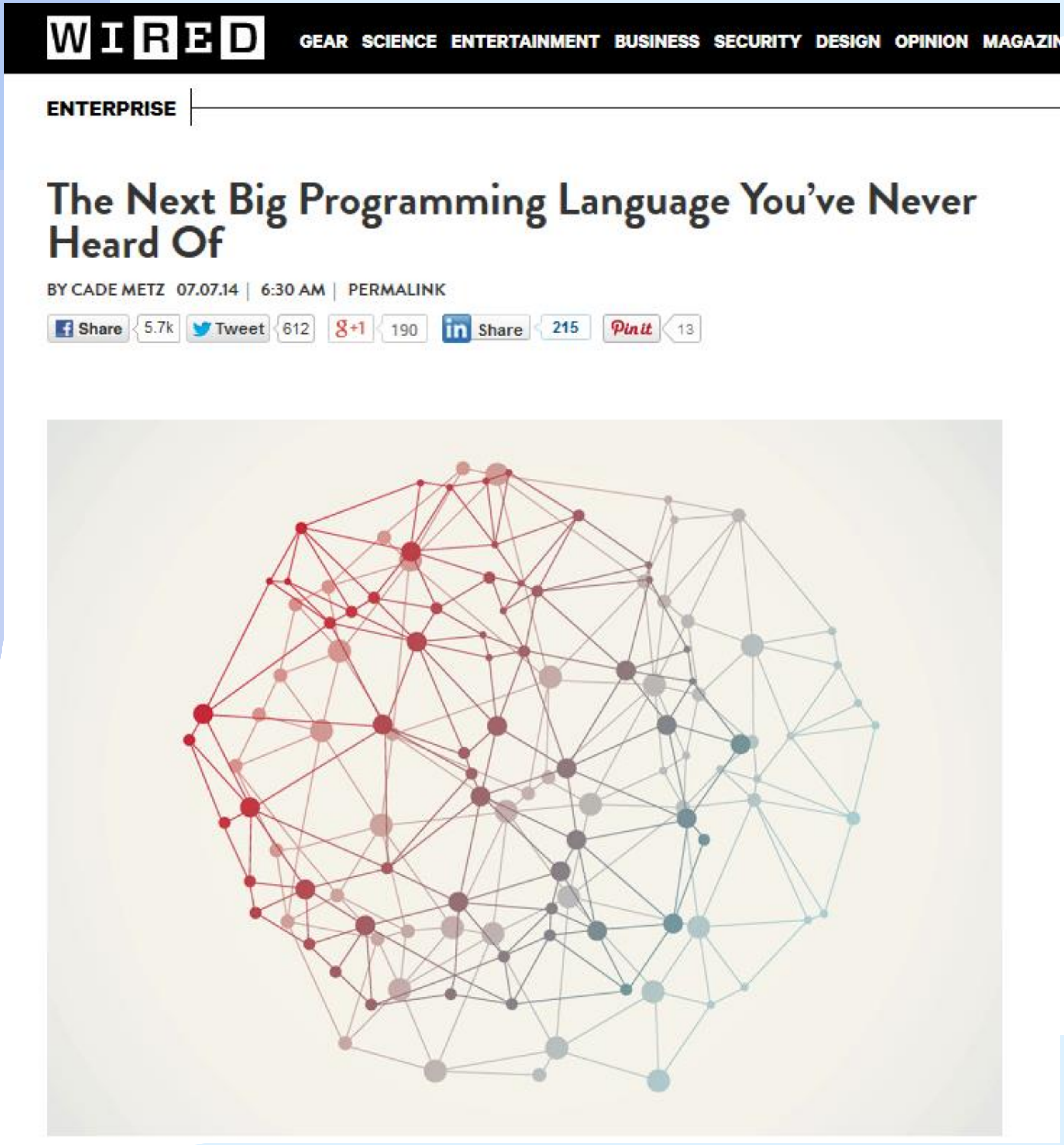
1. Python
2. JavaScript
3. Java
4. C#
5. C
6. C++
7. PHP
8. Swift
9. Go
10. Ruby

Top 10: Most In-Demand Programming Languages 2021

1. ***JavaScript (62%)***
2. ***Java (59%)***
3. ***Python (48%)***
4. ***C# (40%)***
5. ***PHP (32%)***
6. ***C++ (27%)***
7. ***Typescript (24%)***
8. ***C (15%)***
9. ***Kotlin (15%)***
10. ***Swift (14%)***

Empresas RH para TI: Entrevistados estão em busca de candidatos com essa habilidade

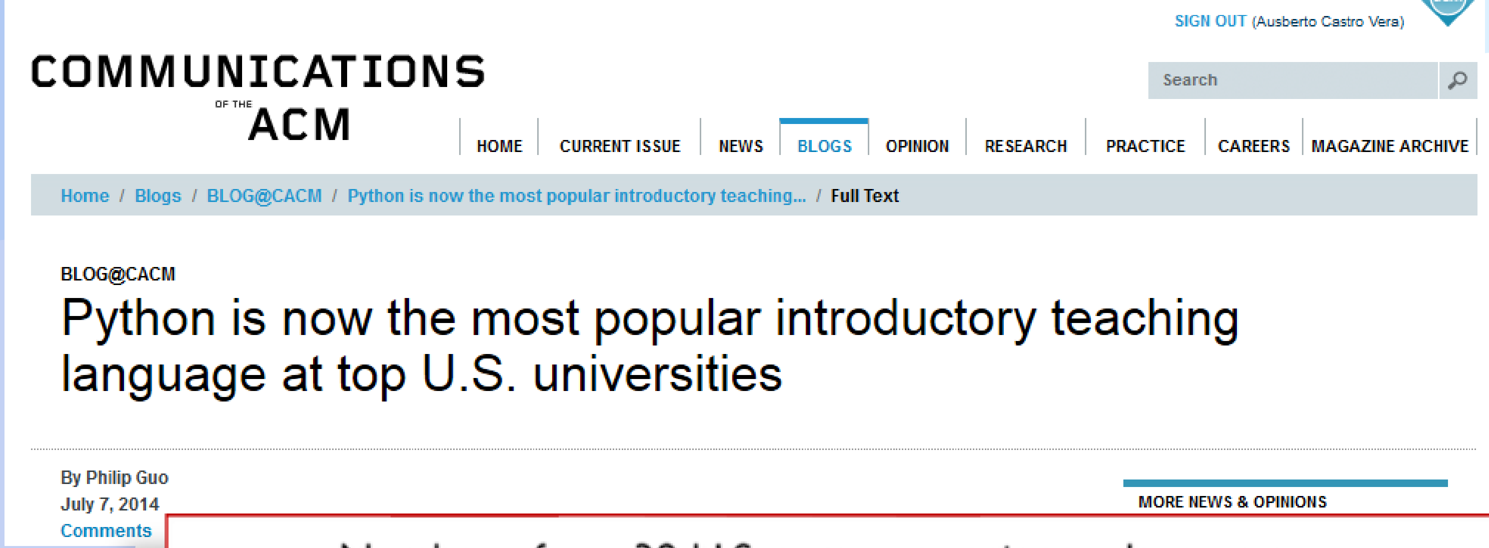
2021 CodinGame annual survey


WIRED
GEAR SCIENCE ENTERTAINMENT BUSINESS SECURITY DESIGN OPINION MAGAZIN
ENTERPRISE
Share 5.7k
Tweet 612
8+1 190
in Share 215
Pin it 13

The Next Big Programming Language You've Never Heard Of

BY CADE METZ 07.07.14 | 6:30 AM | PERMALINK

http://dlang.org/



SIGN OUT (Ausberto Castro Vera)

COMMUNICATIONS OF THE ACM

Search

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTICE | CAREERS | MAGAZINE ARCHIVE

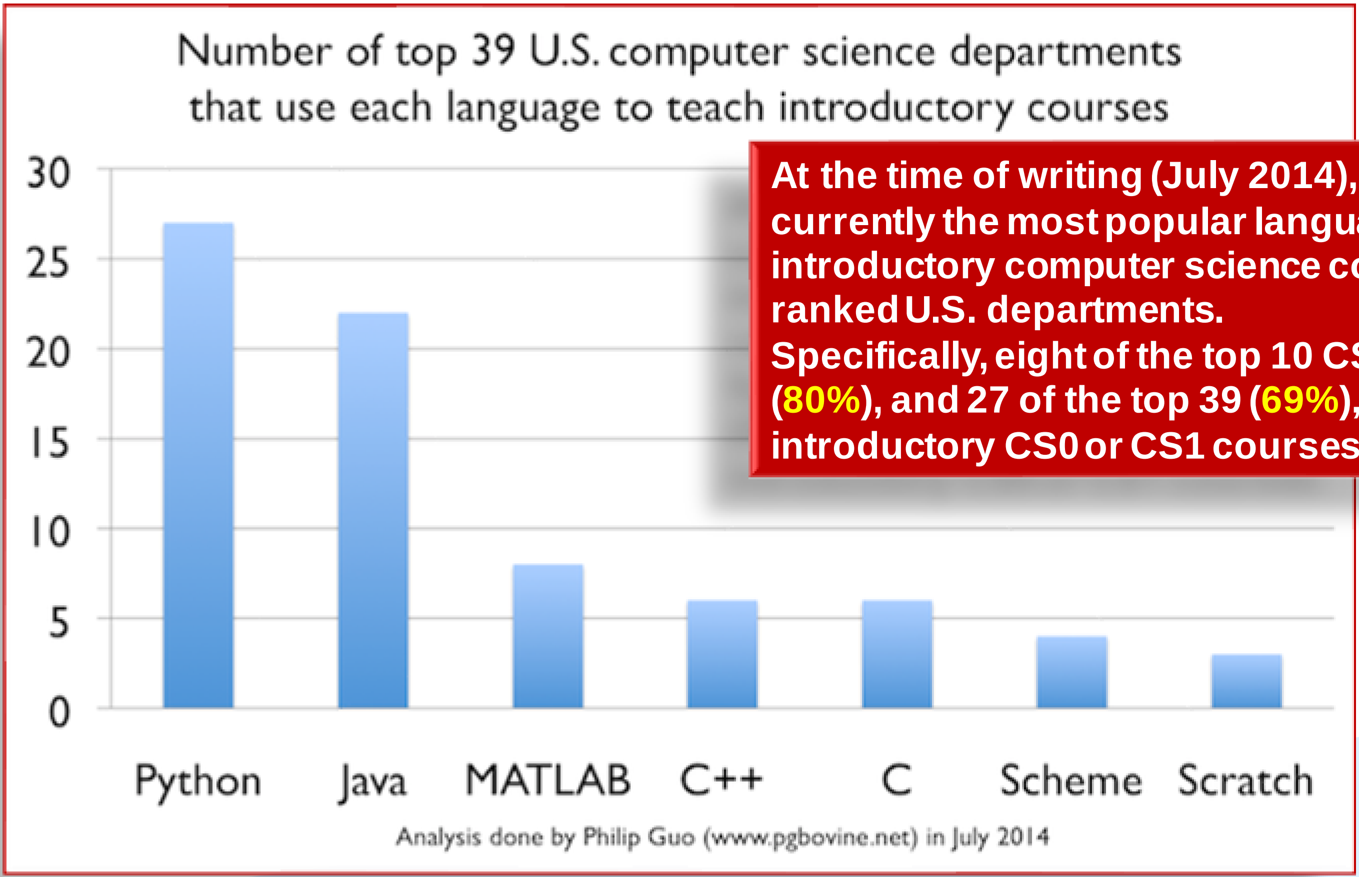
Home / Blogs / BLOG@CACM / Python is now the most popular introductory teaching... / Full Text

BLOG@CACM

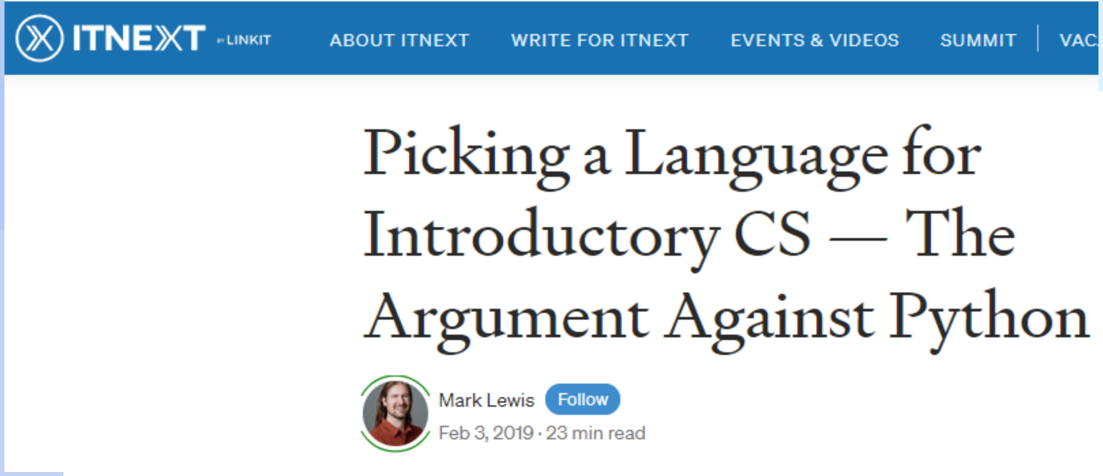
Python is now the most popular introductory teaching language at top U.S. universities

By Philip Guo
July 7, 2014
Comments

MORE NEWS & OPINIONS


Number of top 39 U.S. computer science departments that use each language to teach introductory courses

At the time of writing (July 2014), Python is currently the most popular language for teaching introductory computer science courses at top-ranked U.S. departments.
Specifically, eight of the top 10 CS departments (80%), and 27 of the top 39 (69%), teach Python in introductory CS0 or CS1 courses.



ITNEXT ABOUT ITNEXT WRITE FOR ITNEXT EVENTS & VIDEOS SUMMIT VAC

Picking a Language for Introductory CS — The Argument Against Python

Mark Lewis Follow
Feb 3, 2019 · 23 min read

Python is that it is "simple" and easy for students to learn

https://itnext.io/picking-a-languages-for-introductory-cs-the-argument-againstpython-4331cca26cfa

Então, quais fatores devem ser considerados ao escolher uma linguagem para a instrução CS1? Minha lista curta para esta postagem é a seguinte:

- **Capacidade de cobrir conceitos importantes.**
- **Aplicar boas práticas para que os alunos não desenvolvam hábitos ruins.**
- **Baixa sobrecarga para pequenos programas.**
- **Capacidade de resolver problemas razoáveis sem introduzir muita complexidade.**
- **Relatórios de erros iniciais.**
- **Facilite o aprendizado de outras linguagens de programação.**
- **Capacidade de escrever coisas que sejam interessantes no primeiro semestre.**
- **Pontos de bônus: ser aberto, multiplataforma e de preferência de código aberto.**

Uma coisa que devo apontar aqui é que estou considerando especificamente CS1 com a intenção de que os alunos em CS1 pretendam adquirir um conhecimento mais profundo de programação, desenvolvimento de software e ciência da computação em geral.

Scala

Python: a possibilidade de os alunos adquirirem hábitos ruins que a linguagem não impede.

Prof. Dr. Ausberto S. Castro Vera
Ciência da Computação
UENF-CCT-LCMAT
Campos, RJ

ascv@uenf.br
ausberto.castro@gmail.com