stanford-cs Documentation

Release 0.1.0

Kevin Leptons

Jun 23, 2017

Contents


LELAND STANFORD JUNIOR UNIVERSITY
DIE LUFT DER FREIHEIT WEHT
1891

CHAPTER 1

Introduction

This project is collection of courses from Stanford Computer Science Course.

CHAPTER 2

Courses

cs106b

PROGRAMMING ABSTRACTIONS

Units

#	Name
001	Welcome
002	C++ Functions
003	Big O, Vectors, Grids
004	Strings and Stanford Library
005	Stacks and Queues
006	Sets and Maps
007	Introduction to Recursion
008	Fractals
009	Recursive Backtracking 1
010	Recursive Backtracking 2 and Exhaustive Search
011	Sorting
012	Memoization and Structs
013	Classes
014	Pointers
015	Dynamic Allocation
016	Linked Lists
017	Implementing Vector
018	Binary Heaps
019	Trees
020	Binary Search Trees
021	Graphs
022	Hashing
023	Graphs II: Minimum
024	BFS and DFS
025	Dijkstra and A*
026	Esoteric Data Structures: Skip Lists and Bloom Filters
027	C++ Inheritance and Polymorphism
028	Final Class

References

- http://web.stanford.edu/class/cs106b/

cs107

COMPUTER ORGANIZATION & SYSTEMS

Units

#	Name
001	Welcome
002	Transitioning to C
003	C pointers & arrays
004	More on pointers
005	void*, generics, callbacks
006	Introduce data representation of the Integer family of types
007	More bits and bytes
008	IEEE Floating Point
009	Intro to AMD64, data movement
010	Data layout and access, ALU ops
011	Control
012	Function call
013	Code optimization
014	Managing the heap
015	Build process
016	Memory hierarchy, locality, caching
017	Wrap

References

- https://web.stanford.edu/class/cs107/

cs110

PRINCIPLES OF COMPUTER SYSTEMS

Units

Warning: No PDF slides

#	Name
001	Introductions and Course Syllabus
002	Introduction to Filesystem APIs
003	More Filesystem APIs, Naming and Layering
004	Filesystem wrap, Introduction to fork
005	More on fork, Simple Sychronization
006	Shells, execvp, Pipes, and Interprocess Communication
007	Pipes, subprocess, Interprocess Communication, and Signals
008	Masking Signal, kill, Virtual Memory, OS Schedulers
009	Multiprocessing Wrap, Introduction to Multithreading
010	Race Conditions, Preventing Them, Transition from C to C++ Theads
011	Philosophers, Deadlock, Permission Slips
012	Condition Variables, Semaphores, Reader Writers, and Myth Busters
013	myth-busters and ice-cream-parlors
014	Introduction to Networking
015	Servers, Clients, HTTP, System Calls
016	web-get, System Calls, Server Architectures
017	Networking Wrap For Real, MapReduce
018	Principles of System Design
019	Nonblocking I/O
020	Nonblocking I/O and Event-driven programming

References

- http://web.stanford.edu/class/archive/cs/cs110/cs110.1176/spring-2017/

cs103

MATHEMATICAL FOUNDATIONS OF COMPUTING

Units

#	Name
001	Set Theory
002	Direct Proofs
003	Indirect Proofs
004	Propositional Logic
005	First-Order Logic, Part I
006	First-Order Logic, Part II
007	Binary Relations, Part I
008	Binary Relations, Part II
009	Functions
010	Cardinality
011	Graphs, Part I
012	Graphs, Part II
013	Induction, Part I
014	Induction, Part II
015	Finite Automata, Part I
016	Finite Automata, Part II
017	Finite Automata, Part III
018	Regular Expressions
019	Nonregular Languages
020	Context-Free Grammars
021	Turing Machines, Part I
022	Turing Machines, Part II
023	Turing Machines, Part III
024	Unsolvable Problems, Part I
025	Unsolvable Problems, Part II
026	Complexity Theory, Part I
027	Complexity Theory, Part II
028	The Big Picture

References

- http://web.stanford.edu/class/cs103/

cs109

INTRODUCTION TO PROBABILITY FOR COMPUTER SCIENTISTS

Units

#	Name
001	Counting
002	Permutations and Combinations
003	Axioms of Probability
004	Conditional Probability, Bayes Theorem
005	Independence
006	Random Variables
007	Variance, Bernoulli and Binomial
008	Poisson and More
009	Continuous Random Variables
010	Normal Distribution
011	Joint Distributions
012	Continuous Joint Distributions
013	Properties of Joint Distributions
014	Convolution and Joint Conditional
015	Beta Distributions
016	Great Expectations
017	Covariance and Correlation
018	Samples
019	Central Theoems
020	Parameters and MLE
021	Maximum a Posteriori
022	Naive Bayes
023	Logistic Regression
024	Towards Deep Learning
025	Deep Learning

References

- http://web.stanford.edu/class/cs109/

cs161

DATA STRUCTURE AND ALGORITHMS

Units

#	Name
001	Insertion sort, loop invariants, runtime analyses, asymptotic analysis
002	Merge sort, divide and conquer
003	Recurrence relations, maximum subarray problem
004	Heaps
005	Partition, median of medians algorithms, more recurrence practice
006	Randomized algorithms, quicksort
007	Lower bounds on comparison sorting, counting sort, stacks, queues, linked lists
008	Binary search trees, red-black trees
009	Hashing
010	Graphs, depth-first search, topological sort
011	Strongly connected components, breadth-first-search, Dijkstra's algorithm
012	Dynamic programming
013	Greedy algorithms, activity selection problem, knapsack problem, Floyd-Warshall algorithm
014	Bellman-Ford algorithm, amortized analysis
015	Minimum spanning trees
016	Max flow

References

- http://web.stanford.edu/class/archive/cs/cs161/cs161.1168/

cs131

COMPUTER VISION: FOUNDATIONS AND APPLICATIONS

Units

#	Name	Slide
001	Introduction to Computer Vision	001.pdf
002	Linear Algebra Primer	002.pdf
003	Optional MATLAB Tutorial	None
004	Pixels and Filters	004.pdf
005	Edge detection	005.pdf
006	Feature detectors	006.pdf
007	Difference of Gaussians, SIFT	007.pdf
008	Camera 1	008.pdf
009	Camera 2	009.pdf
010	Camera 3	Same #009
011	Human Vision: Insights and Inspiration for Computer Vision	011.pdf
012	Basic Concepts of Segmentation	012.pdf
013	K-means Clustering	013.pdf
014	Feature tracking	014.pdf
015	Segmentation in Human Vision: Implications for Computer Vision	015.pdf
016	Basic Concepts in Recognition, Nearest Neighbor Match	016.pdf
017	PCA and Eigenfaces	017.pdf
018	Tracking Millions of People	018.pdf
019	Networks and Hierarchical Processing: Object Recognition in Human and Computer Vision	019.pdf
020	Towards the AI Dream	None

References

http://vision.stanford.edu/teaching/cs131_fall1415/index.html http://vision.stanford.edu/teaching/cs131_fall1415/schedule.html

CHAPTER 3

Development

System

Fall down from top to bottom to install essential packages make develop system. No entry help for auto install from a to z now. Try to pass though this section. Building work for UNIX platforms, such as Operating Systems below:

- Linux
- Macintosh

Python

Python is main programming language of this project. Project build on python v3.4, installation for each platform below

- Python on Unix platform
- Python on Macintosh platform

For any one who is newbie with python

- Python command line
- Python language reference
- Python tutorial
- Python documentation

Git

Git is version control system of this project and github is git server. This project located in repo on github Installation for each platform below

- Git on Unix platform

- Git on Macintosh platform

SSH key

If you want to direct write into repository then contact with kevin leptons to add your ssh key into github. If not, you can make pull request to repository at github later. Follow instructions to setup ssh key

```
# step 1: generate your ssh key
# if you early have ssh key, skip this command
# perform this command, then confirm default options by hit enter
# generated public ssh key located in ~/.ssh/id_rsa.pub
ssh-keygen

# step 2: add your ssh to repository at github
# contact kevin, give your public ssh key located in ~/.ssh/id_rsa.pub
# to him and wait for handle
```

Editor

This is option package. Feel free to use your favourite editor. This section suggest some editor for who is have not any one editor or want to discover

- Atom - cross platform
- Vim - cross platform

Source Code

Get source code at github.

```
git clone https://github.com/kevin-leptons/stanford-cs.git
cd stanford-cs
```

File System

Source file

This is list of source files. It will not deleted by *./ctl doc –clear*

File	Descriptions
courses	Stanford courses
script	Tools of development
test	Unit test
dev-requires.txt	Specify develop packages
readme.md	Short manual
setup.py	Command to control development
.gitignore	Specify files which is ignore by git
.travis.yml	Specify how project will be build, test with travis service
.git	Meta data of git repository

Temporary file

Below is list of files which is created during building. It is can be delete by *./ctl doc –clear*

File	Descriptions
build	Build files, result of building project
dest	Build files, result of build document
dist	Build files, result of distribution
gwisp.egg-info	Build files, result of building project
venv	Contains virtual environment
tmp	Store temp files

Standard

Python coding style

This project use PEP8 for python coding style

File naming

File name must easy to read/write in command line. For that, it must pass regex expression

```
^[a-z0-9-_]+(\.[a-z0-9])*$
```

Examples

Valid	Invalid
document.rst	Document.rst
ai-service.py	ai service.py
ai_service.py	AIService.py

For split between word in file name, prefer **-** than _. Use _ only that is specification of other packages

Sample

Package's files	Valid	Invalid
pytest	test_coding.py	test-coding.py
sphinx	dev_install.rst	dev-install.rst

Virtual Environment

Enter virtual environment before developement, because development tools work in that environment.

```
# python virtual environment
./env init
. venv/bin/active
./env install
```

Documentation

Write document

Put document in **doc** directory. Document file must pass regex expression

```
^[a-z0-9_]+\.rst$
```

Samples

Valid	Invalid
introduction.rst	Introdcution.rst
dev.rst	dev
dev_install.rst	dev-install.rst

For write document, see reStructure

Build

```
# This command build document from rst files
# Put result in dest/doc/html
./ctl doc

# Open html document in dest/doc/html/index.html with default
# browser
./ctl doc --view
```

Clear

```
# For new document file added, you must clear document build
# then build again
./ctl doc --clear
```