Welcome to CS106A

- Instructor - Nick Parlante
  CS senior lecturer, previously at Google
- Head TA - Elyse Cornwall
- Course page for everything: cs106a.stanford.edu
- CS106A teaches basic coding and problem solving
- No prior experience required

How Does This Class Work?

- Start the very beginning
- Careful to go step by step
- Like swimming, you can't just read about it
- Learn by doing
  "reps" - Nick's favorite word
  Technology to practice ideas right away
- In just a couple weeks
- You will be amazed at what you can build
- Partly because computers are powerful
- Partly because coding is easier than you think

Why Take CS106A?

- 1. Understand What Computers Can Do
  The nature of computers is code

Understand what computers can and cannot do
Not some hocus-pocus mythology of computers

- 2. Solve Real Code Problems
  Python is powerful
  In 10 weeks, you'll learn enough to solve real problems
  Handy if you take CS106A early enough to use it
- 3. Hidden Agenda
  You might find you like it
  Take CS106B
  We have a whole department of this stuff!

Other CS Courses

(pasted from syllabus)

CS106A is the main, first course in programming and computer science, for people who with zero experience.

CS106B is the second course, teaching more advanced programming and computer science for people who know basic programming.

CS105 is a more lightweight introduction to CS ideas, but without as much coding as CS106A.

CS193q is a 1 unit seminar that teaches Python very very quickly - geared for people who already know how to program but want to learn Python.

CS106AX is an "accelerated" variant of CS106A taught this quarter for people who already have some programming experience. It's more advanced and moves more quickly, covering both the Javascript and Python languages.

Foreshadowing

The story arc or a typical CS106A project...

Step-1 - Code is going to look bad

- Angry Emoji 😡
- Code will look kind of awful
  This will be evident right away!
- Weird syntax
  e.g. `MMDDYYY`
- Run won't work right
- Unhelpful error messages
- Computer systems can be weirdly stupid and inflexible
  You see this trying to enter data into web forms
  Like you type a comma in the wrong place and everything breaks

Aside: Why is syntax inflexible?

- Computers are powerful but mechanical and kind of stupid
- e.g. `MMDDYYY`
- The format is cryptic, but it is not **difficult**
- This is the basic deal of computer systems
  You, as a programmer, have an idea for the real world
  You express the idea in simple syntax so the computer can understand
- This unlocks the power of the computer

Step-2 - And Then it's Going To Work

- The badness is largely **superficial**
- Push through the superficial syntax
  It's weird, but not difficult
  Weird but not difficult = CS106A week 1
- We will show you how A connects to B, B connects to C

- Show you how to fix some common problems
- And then your code is going to work
- Delightful!
- "That's it?" you will say
  "That's all it takes?"

Course Details

We will go through the main course details quickly here, then we'll start coding. Please see the syllabus for more information.

Links on Course Page

- cs106a.stanford.edu - everything is here
  e.g. office hours
- Elyse Cornwall - our super head TA
  Email her if you need a problem solved, extension
  Email before classtime for quick response
- Python Guide free reference Nick is writing
  link on course page
- Videos (on canvas) available after lecture within about 2 hours
  The videos will be sufficient for you to follow along with the course
- We like in-person attendance
  We try to run a dynamic lecture
  Our room *almost* has enough seats for everyone

Python Language

- Using Python3

- Python is "programmer friendly"
  get things done easily
- Not just for CS people
- Very popular for data, all sorts of things

Education Theory - Learn By Doing

- Education research, Carl Wieman (Stanford)
  Do a little activity with what you just saw
- Lab/Exercises within lecture - reinforce
- "Reps" - Nick's Favorite Word
  Learn by working a few examples
  Tech makes it easy to have many examples ready

Lecture Structure and Exercises

- I'll provide notes like this each day
- Including code exercises
- The lecture exercises always have a **Show Solution** button
- Lecture goes pretty fast
- After lecture, you should, say, 60% understand the ideas
- You will likely need to review the examples to solidify your understanding
  Try to solving a lecture exercise yourself
  Perhaps done right after class
  Perhaps done when the homework comes out on these topics
- Ultimately, you want to be able to solve the lecture exercises without peeking at a solution

Weekly Section

- Small group section
- Weekly review of pre-homework problems
- Section leader will also grade all your homeworks
- Section times on Wed - Thu - Fri
- Watch Wed lecture first, as section builds on this
- Signups Thu week-1

Typical Week Schedule

- Give out homework (HW) on Thu
- Lecture on that topic through Fri
- Section, say, Thu on HW topic
  Section is ~12 people, get individual help
- HW is due Tue or Wed following week
- We are careful to give you the weekend
- Advice: Start the HW Sun eve, see how it looks

Late Days

- Theory:
  We will give you a little more time to get it right
  Within limits!
- Say homework is due Wed 11:55pm
- 2% on time bonus

- 48 hour grace period (so .. Fri 11:55)
- Then 15% off per day
- Ask Elyse for extra time in exceptional circumstances
  Before classtime Fri is best
- Can get partial credit for partial solution
- Advice: shoot for the Wed deadline

Office Hours - Lair

- CS106A provides a lot of help - The Lair
- The Lair has a huge team of section leaders, providing help 7:00-11:00 Sun-Thu eve.
  Starts Sun at the end of week-1
  See "Getting Help" on course page
- Has "online" option for online students

Office Hours - Nick and Elyse

- Nick and Elyse - right after class
  Just bring your laptop down and ask
- Then we have hours most days - see course page

Ed Forum

- Ed is a class forum for students to ask and students and staff to answer questions
- Don't post a copy of your code into Ed
- If you have a weird issue, you can see if someone has a solution on Ed

- If you need someone go over your code, use the office hours and the Lair

Honor Code

- See long form in syllabus
- 1. You can talk to other students, share ideas, plans, insights
- 2. You need to write and debug your own code
- 3. Avoid looking at another student's code
- 4. Please - don't put your solutions on the Internet
- Honor Code problems stem from a moment of weakness
- If you are in a real fix, talk to Nick to work something out, pass this class Instead of making a big mistake
- We have automated tools for finding issues
- We may not scan for Honor Code problems until the end of the quarter
- Week 10 retractions

Nick's Theory of the 10 Week Quarter

- We're going to go quite fast at the start of the quarter
- Reduce the load at the very end
- This first material is **weird** but not that difficult
- Just hang in there, this will give us more time later