Universidade de São Paulo
Faculdade de Medicina de Ribeirão Preto
Depto. de Neurociências e Ciências do Comportamento

Fundamentos de programação computacional para as neurociências

Dr. Rafael Naime Ruggiero
Prof. Dr. João Pereira Leite

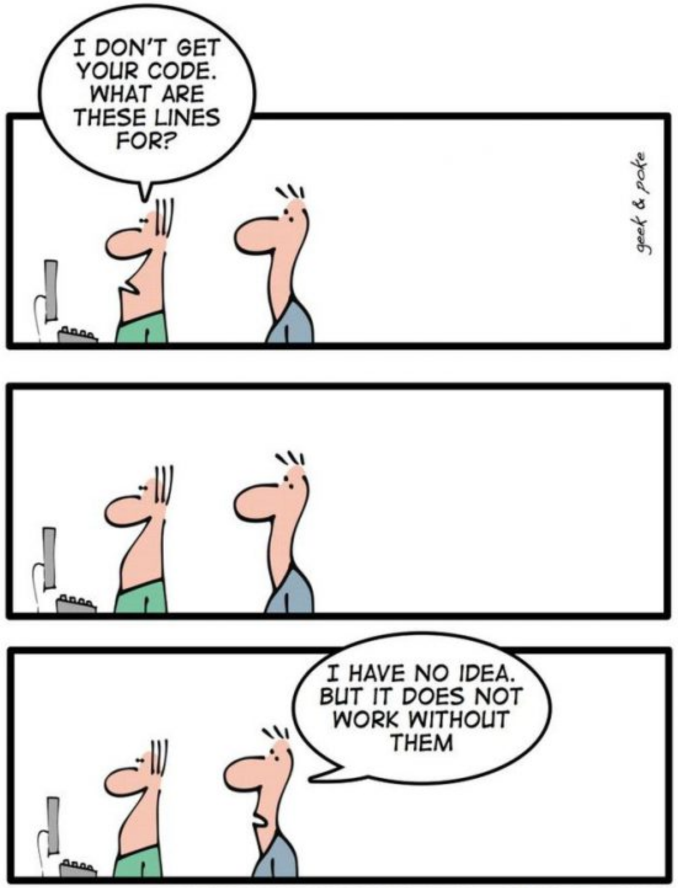
Apresentação do curso

For whom is this course?

For whom is this course?

- Never programmed

- Basic skills on programming


I DON'T GET YOUR CODE. WHAT ARE THESE LINES FOR?
geek & poke
I HAVE NO IDEA. BUT IT DOES NOT WORK WITHOUT THEM

THE ART OF PROGRAMMING - PART 2: KISS

What is the purpose of the discipline?

- Lose the fear of programming
- Make you independent to learn

Will I become a programmer?

Will I become a data analyst?

Will I be able to analyze different types of neuroscience data?

Syllabus

1) Introdução à utilização do Matlab/Octave: Utilização de linguagem de programação para análise de dados, conceitos básicos, variáveis, matrizes e vetores.

2) Fundamentos de programação: estruturas de controle de fluxo. Scripts e funções. Importando e exportando arquivos.

3) Construção de figuras: gráficos de barras, gráficos de linhas, gráficos de distribuição, histogramas.

4) Álgebra matricial e suas aplicações em análise de dados em neurociência.

5) Análise de dados: estatística descritiva, testes paramétricos e não paramétricos, covariância e correlação, regressão linear.

Class structure

Monday (14h)
video lesson
Exercise
test

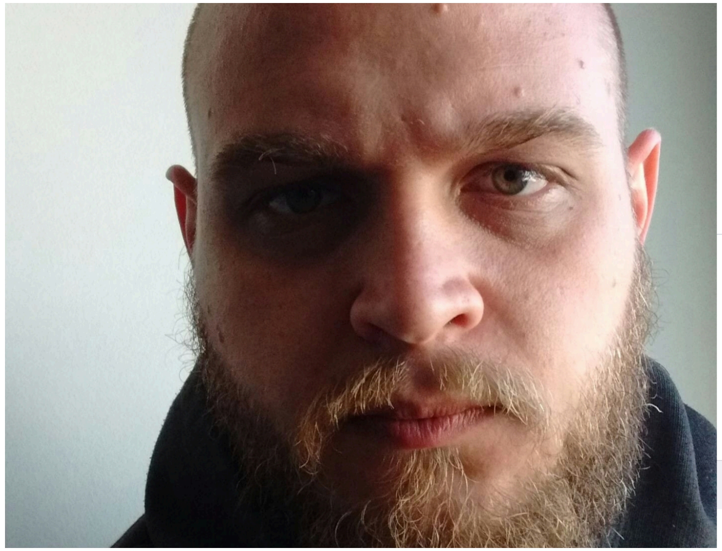
Friday (16- 18h)
Online tutoring

Monday (18h)
test

Hands on!!

Hands up!!

Classes



Leonardo Rakauskas

Danilo Benette Marques

Course presentation

Bibliography

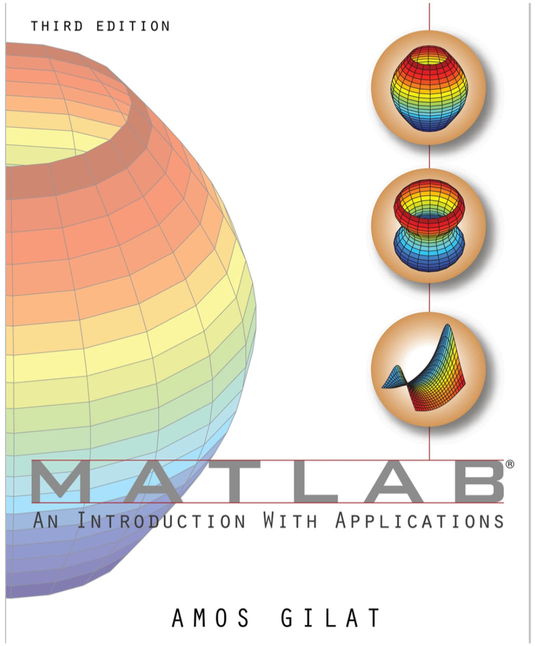
Borgo, M. Soranzo, A. & Grassi, M. (2012). MATLAB for Psychologists. Springer: New York.

Cohen, M.X. (2017). MATLAB for brain and cognitive scientists. The MIT Press: Cambridge.

Hann, B. Valentine, D.T. Essential MATLAB for engineers and scientists. 6th edition. Elsevier: Cambridge.

Rosenbaum, D. A. Vaughan, J. & Wyble, B. (2014). MATLAB for Behavioral Scientists. 2nd edition. Psychology Press: New York.

Wallisch, P. Lusignan, M. Benayoun, M. Baker, T.I. Dickey, A.S. & Hatsopoulos, N. (2014). MATLAB for Neuroscientists: An introduction to scientific computation in MATLAB. 2nd edition. Academic Press: Amsterdan.



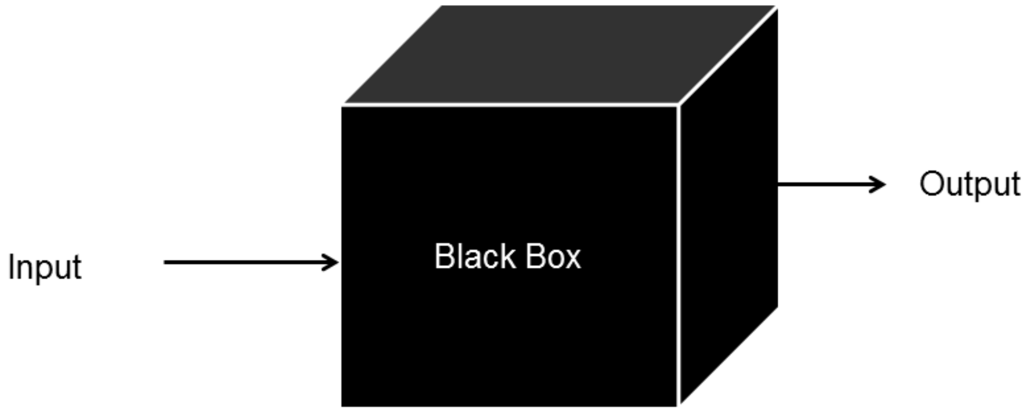
Grading

- Presence
- Weekly test = mean of 6 (MWT)
- Project development (P)

Final Grade = (MWT+P)/2

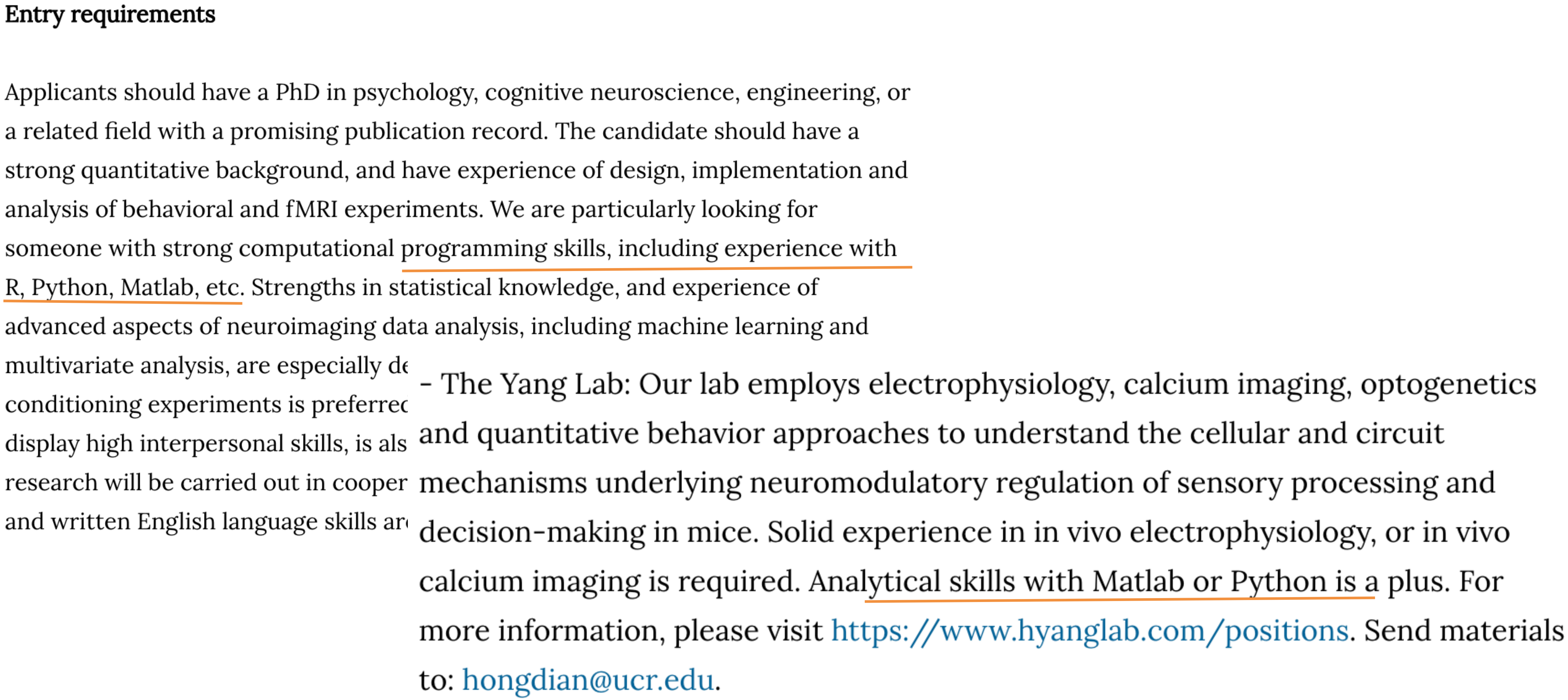
Why programming?

- Learn a new skill
- You can work faster (automatization)
- Discover more creative solutions
- Improves communication
- Get free from the software black boxes
- Essential in some areas
- CV and jobs opportunities



Internal behavior of the code is unknown

Why programming?



Entry requirements

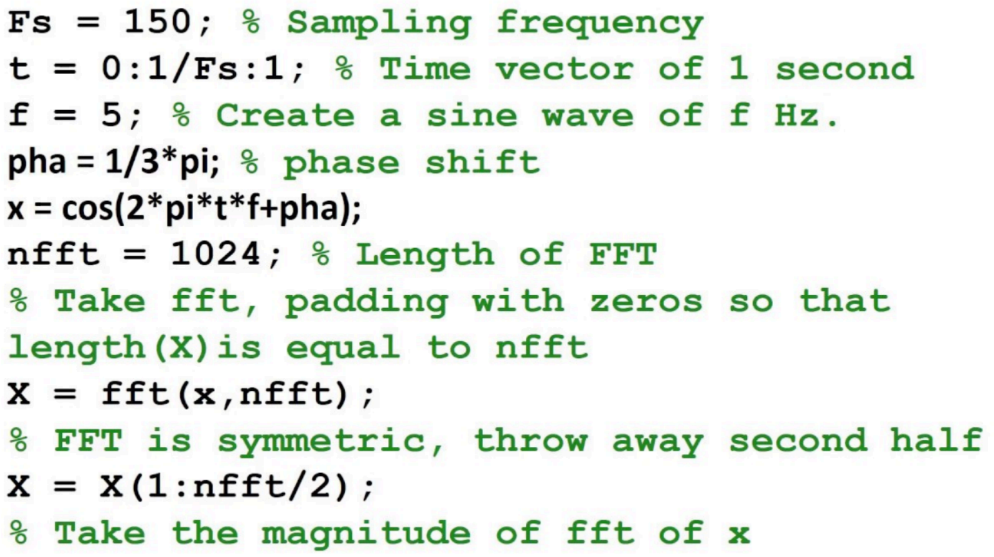
Applicants should have a PhD in psychology, cognitive neuroscience, engineering, or a related field with a promising publication record. The candidate should have a strong quantitative background, and have experience of design, implementation and analysis of behavioral and fMRI experiments. We are particularly looking for someone with strong computational programming skills, including experience with R, Python, Matlab, etc. Strengths in statistical knowledge, and experience of advanced aspects of neuroimaging data analysis, including machine learning and multivariate analysis, are especially de
conditioning experiments is preferrec
display high interpersonal skills, is als
research will be carried out in cooper
and written English language skills ar

- The Yang Lab: Our lab employs electrophysiology, calcium imaging, optogenetics and quantitative behavior approaches to understand the cellular and circuit mechanisms underlying neuromodulatory regulation of sensory processing and decision-making in mice. Solid experience in in vivo electrophysiology, or in vivo calcium imaging is required. Analytical skills with Matlab or Python is a plus. For more information, please visit https://www.hyanglab.com/positions. Send materials to: hongdian@ucr.edu.

Why programming?

- Do data analysis
- Learn data analysis

$$
\begin{aligned}
a_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} s(x)\cos(nx)\,dx = 0, \quad n \geq 0. \\
b_n &= \frac{1}{\pi}\int_{-\pi}^{\pi} s(x)\sin(nx)\,dx \\
&= -\frac{2}{\pi n}\cos(n\pi) + \frac{2}{\pi^2 n^2}\sin(n\pi) \\
&= \frac{2\,(-1)^{n+1}}{\pi n}, \quad n \geq 1.
\end{aligned}
$$



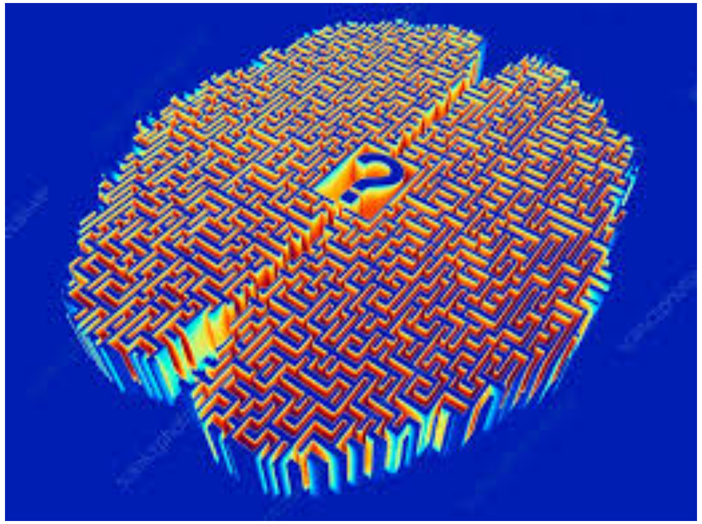
```
Fs = 150; % Sampling frequency
t = 0:1/Fs:1; % Time vector of 1 second
f = 5; % Create a sine wave of f Hz.
pha = 1/3*pi; % phase shift
x = cos(2*pi*t*f+pha);
nfft = 1024; % Length of FFT
% Take fft, padding with zeros so that
length(X)is equal to nfft
X = fft(x,nfft);
% FFT is symmetric, throw away second half
X = X(1:nfft/2);
% Take the magnitude of fft of x
```

Neuroscience

"I Want to Be a Scientist; Do I Also Need to Be a Good Programmer?"

brain is really complex, and technology for measuring it is getting really sophisticated.

point- and-click software tools will impose stronger limits on the experiments you can do and the analyses you can perform.



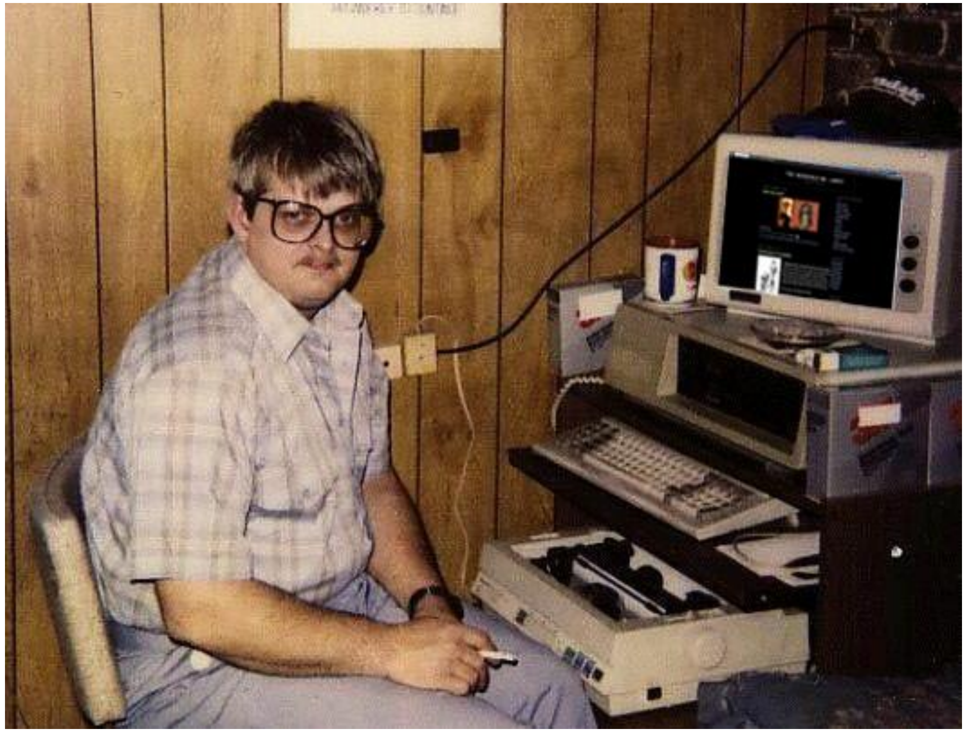
learning to program is learning a skill. Programming is problem solving.

Cohen, 2017.

Why programming?

Programming is for everyone

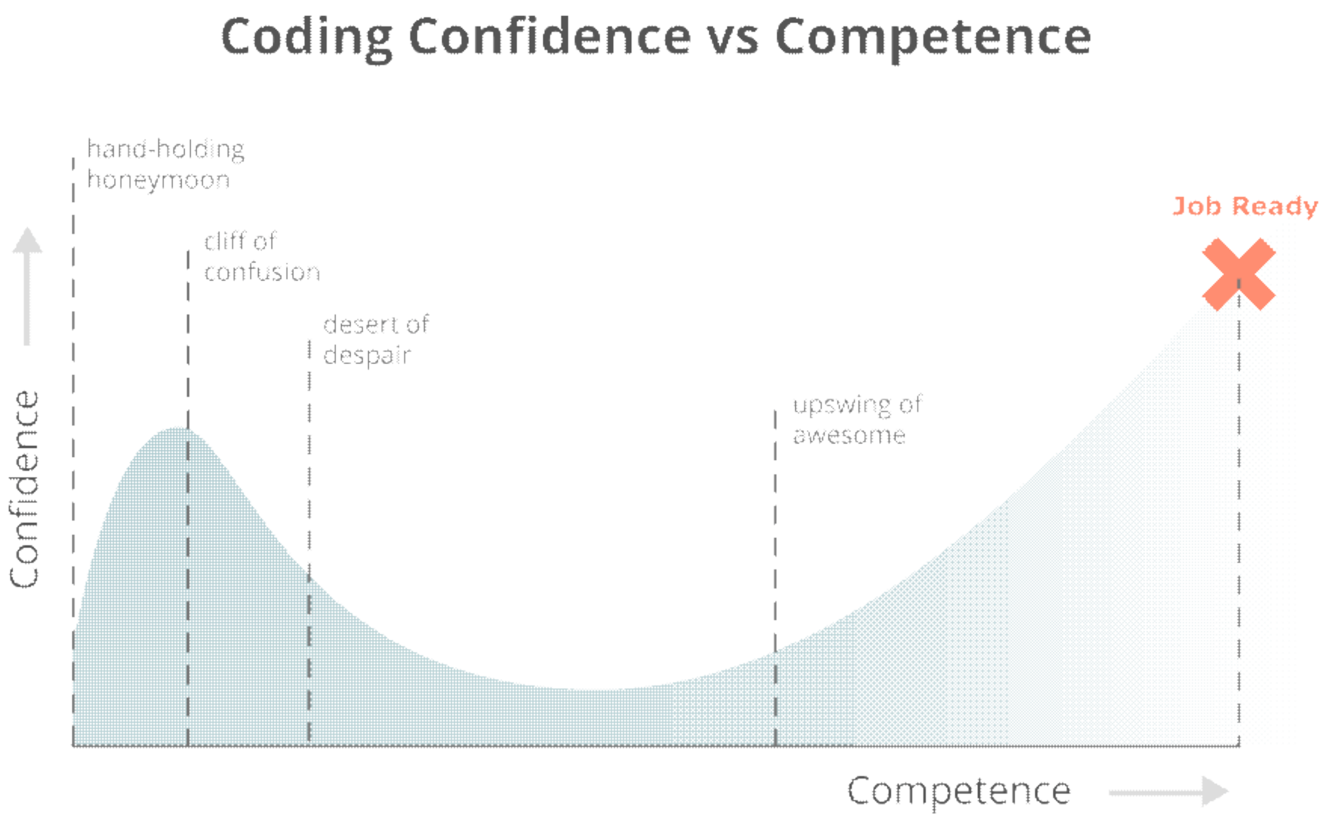
Programming isn’t for everyone

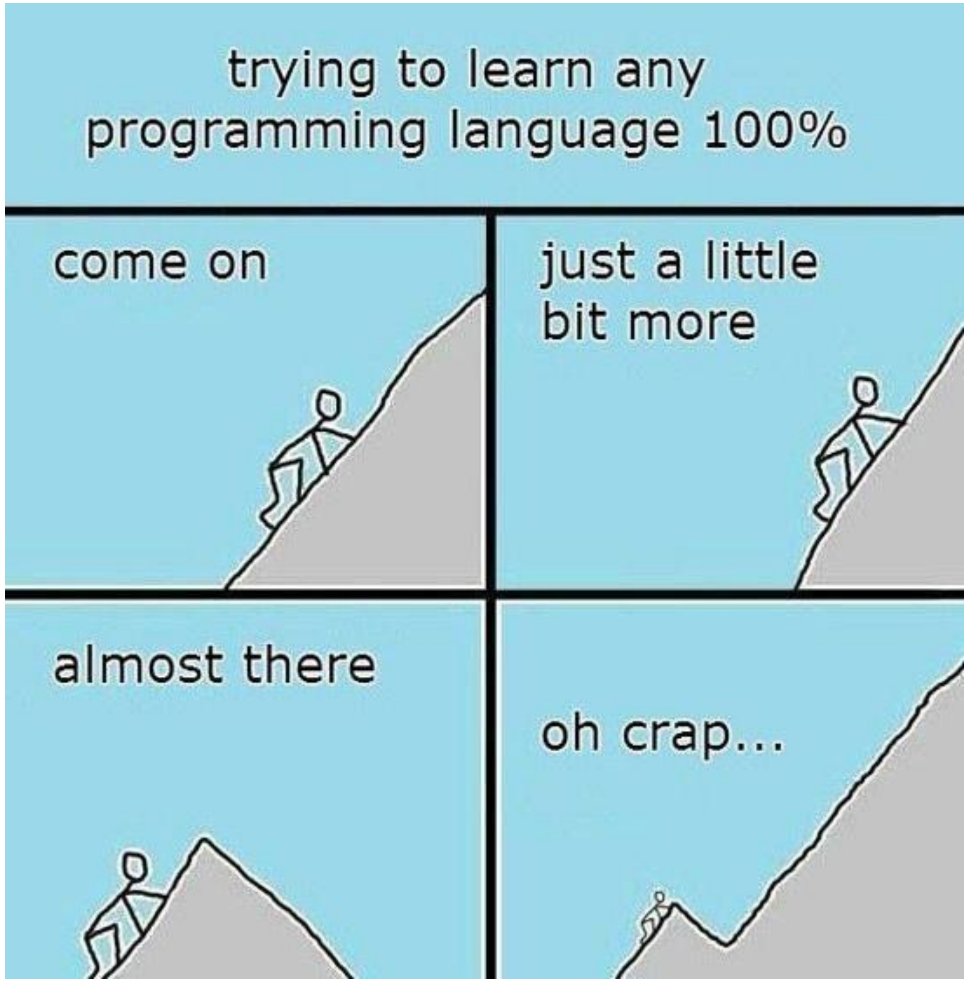


Why programming?

Course presentation

How long?





Why Matlab?

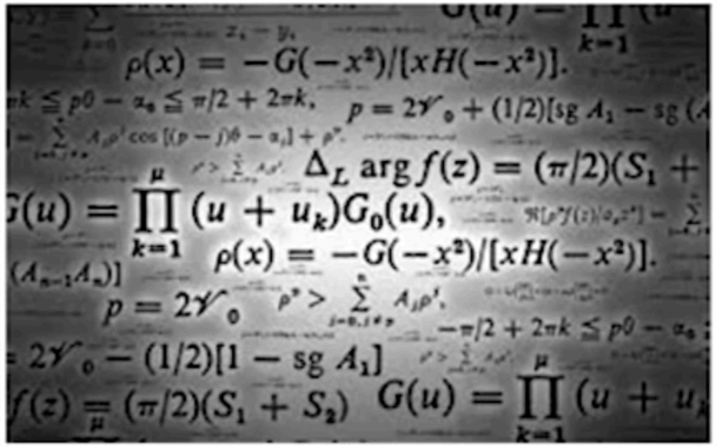
- Widely used
- Built-in functions and toolboxes
- Science and engineering
- Easy to learn (as first language)
- Transpose to other languages (Python, R, Julia, etc)
- Commercial (expensive)

What does a computer do?

- Fundamentally:
  - performs **calculations**

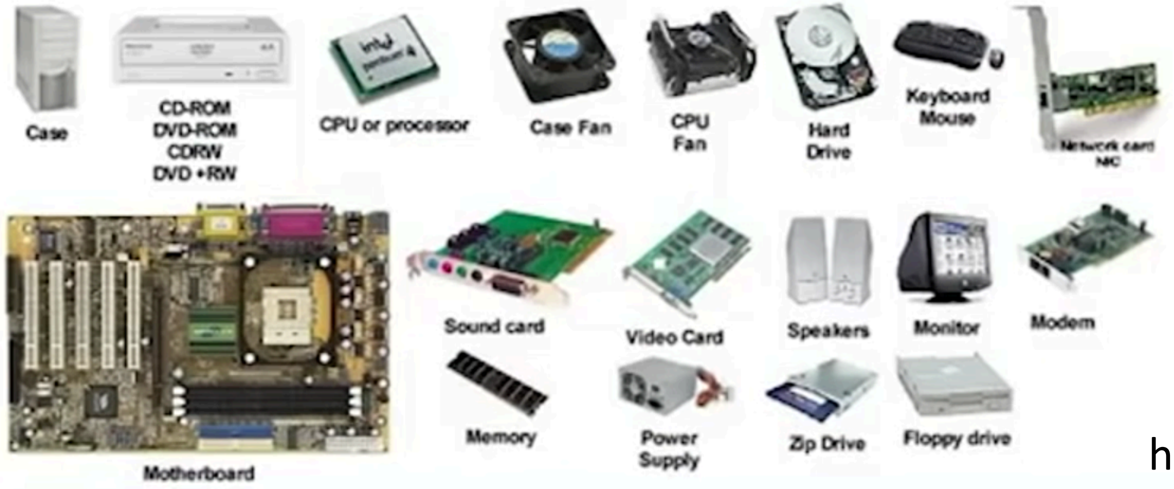
    a billion calculations per second!
  - **remembers** results

    100s of gigabytes of storage!
- What kinds of calculations?
  - **built-in** to the language
  - ones that **you define** as the programmer
- computers only know what you tell them



Computers are machines

- how to capture a recipe in a mechanical process
- **fixed program** computer
  - calculator
- **stored program** computer
  - machine stores and executes instructions



Instructions

Ingredients

- 1 package (1/4 ounce) active dry yeast
- 1 teaspoon sugar
- 1-1/4 cups warm water (110° to 115°)
- 1/4 cup canola oil
- 1 teaspoon salt
- 3-1/2 cups all-purpose flour
- 1 small onion, chopped
- 1 can (15 ounces) tomato sauce
- 3 teaspoons dried oregano
- 1 teaspoon dried basil
- 1 medium green pepper, diced
- 2 cups shredded part-skim mozzarella cheese

Directions

1. In large bowl, dissolve yeast and sugar in water; let stand for 5 minutes. Add oil and salt. Stir in flour, a cup at a time, until a soft dough forms.

2. Turn onto floured surface; knead until smooth and elastic, about 2-3 minutes. Place in a greased bowl, turning once to grease the top. Cover and let rise in a warm place until doubled, about 45 minutes. Meanwhile, cook beef and onion over medium heat until no longer pink; drain.

3. Punch down dough; divide in half. Press each into a greased 12-in. pizza pan. Combine the tomato sauce, oregano and basil; spread over each crust. Top with beef mixture, green pepper and cheese.

4. Bake at 400° for 25-30 minutes or until crust is lightly browned.

Instructions

- Get students list
- Assign a number for each student
- Get the first random number (1 to 10)
- Check student name

Course presentation

Programming

The Three Steps of Programming

Step 1: Think.

Step 2: Write the code.

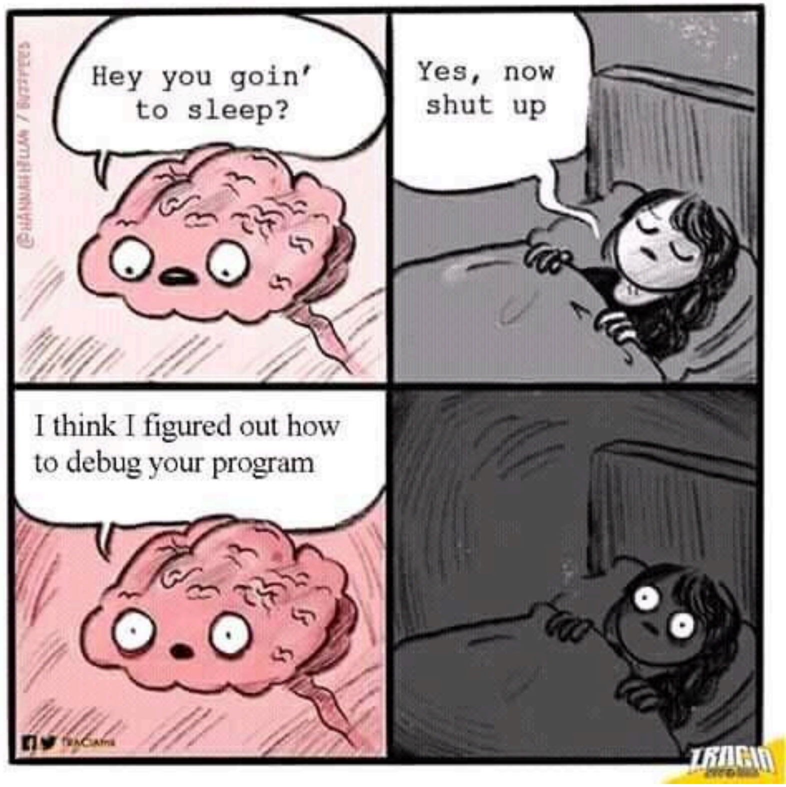
Step 3: Debug.

Step 4: show/share your code

Cohen, 2017.

Programming

Programming languages are languages. They have vocabulary and syntax, they have sentences (lines of code), paragraphs (sections of code), and discourse, and they have styles and ways of thinking.

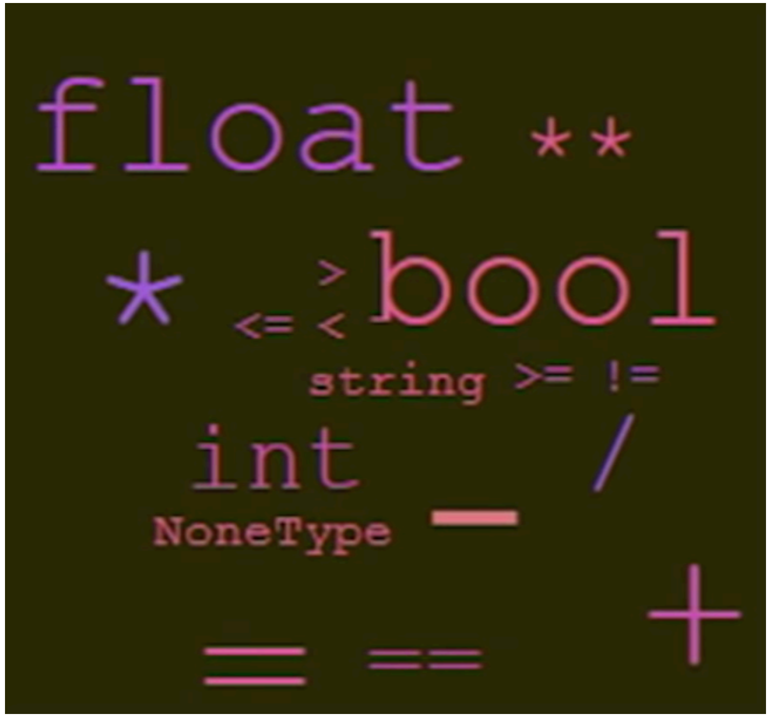


> "No one is born a programmer. The difference between a good programmer and a bad programmer is that a good programmer spends years learning from his or her mistakes, and a bad programmer thinks that good programmers never make mistakes."

Cohen, 2017.

Aspects of language

- **primitive constructs**
  - English: words
  - programming language: numbers, strings, simple operators



Cohen, 2017.

Aspects of language

- **Syntax**
  - English: “table dog boy” → Not syntactically valid

    “boy hugs dog” → syntactically valid
  - Programming language: ‘hi’ 5 → Not syntactically valid

    ‘olá’ or 2+4 → syntactically valid

Aspects of language

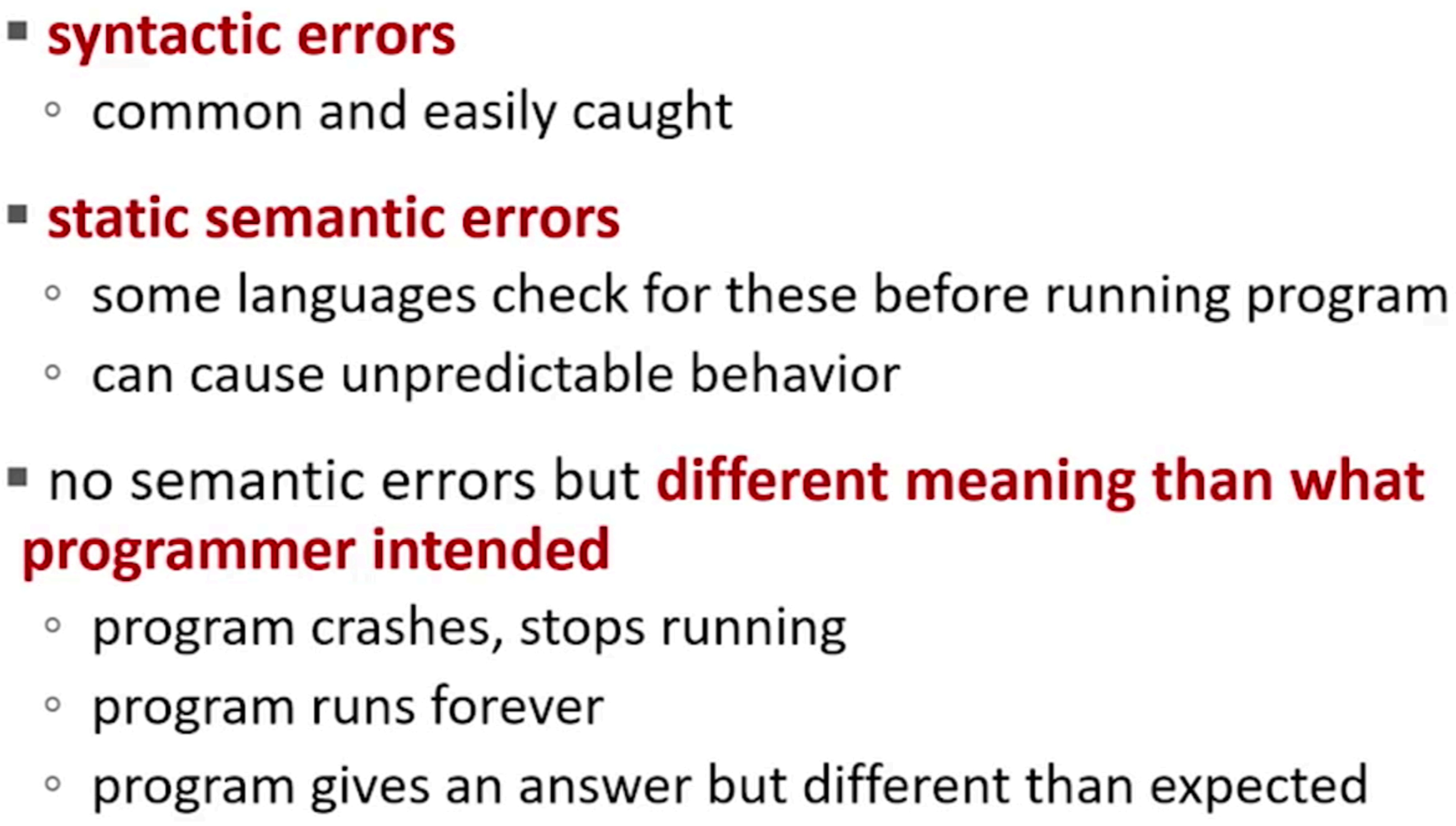
- **Semantics** is which syntactically valid strings have meaning

  - English: “I are hungry” → syntactically valid
But semantic error

  - Programming language: 3.4*2 → syntactically valid

    3+’ola’ → syntactically valid
But semantic error

Where things go wrong



- **syntactic errors**
  - common and easily caught
- **static semantic errors**
  - some languages check for these before running program
  - can cause unpredictable behavior
- no semantic errors but **different meaning than what programmer intended**
  - program crashes, stops running
  - program runs forever
  - program gives an answer but different than expected