Presentation of Cogmaster PROG 101 & PROG 201

Christophe Pallier

Why do cognitive scientists need to learn programming?

Why do cognitive scientists need to learn programming?

- **Theory**
  To understand some fundamental ideas of cognitive science, such as information processing or computation (see, e.g., the *Computational Theory of Mind*)

Why do cognitive scientists need to learn programming?

- **Theory**
  To understand some fundamental ideas of cognitive science, such as information processing or computation (see, e.g., the *Computational Theory of Mind*)

- **Modeling**
  To create simulations of cognitive processes, e.g. neural networks

Why do cognitive scientists need to learn programming?

- **Theory**
  To understand some fundamental ideas of cognitive science, such as information processing or computation (see, e.g., the *Computational Theory of Mind*)

- **Modeling**
  To create simulations of cognitive processes, e.g. neural networks

- **Experimentation**
  To construct stimuli and run experiments

Why do cognitive scientists need to learn programming?

- **Theory**
  To understand some fundamental ideas of cognitive science, such as information processing or computation (see, e.g., the *Computational Theory of Mind*)

- **Modeling**
  To create simulations of cognitive processes, e.g. neural networks

- **Experimentation**
  To construct stimuli and run experiments

- **Data Analyses**
  To analyse behavioral or brain data, or corpora

Why do cognitive scientists need to learn programming?

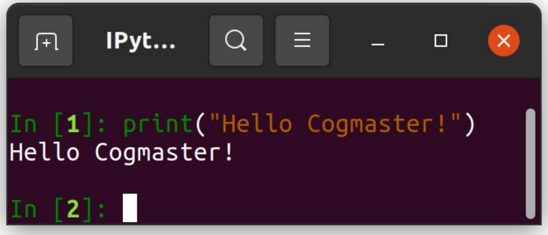
- **Theory**
  To understand some fundamental ideas of cognitive science, such as information processing or computation (see, e.g., the *Computational Theory of Mind*)

- **Modeling**
  To create simulations of cognitive processes, e.g. neural networks

- **Experimentation**
  To construct stimuli and run experiments

- **Data Analyses**
  To analyse behavioral or brain data, or corpora

  - And automate boring tasks (e.g. renaming all files in a folder recursively, ...)

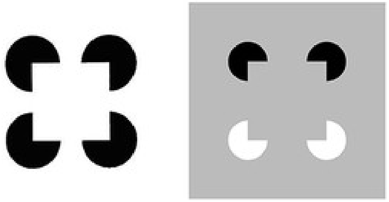


```
In [1]: print("Hello Cogmaster!")
Hello Cogmaster!

In [2]: 
```

PROG 101
Introduction to Programming

- **Audience**: students with little or no prior knowledge of programming.
- **Content**:
  - fundamentals concepts of coding : variables, expressions, functions, control structures, file input/output.
  - The language is *Python* (a good first programming language; knowledge transferable to other languages)
  - Skills: Students will learn to write simple scripts using a code editor, to use the terminal to navigate the file system, to launch scripts,…
- **Timetable:** Courses will take place during S1 (13 sessions of 3 hours)
- **Validation**: PASS/FAIL based on attending the lectures and working on the exercices
- **Instructor:** Henri Vandendriessche (henri.vandendriessche@ens.fr)



PROG 201
Programming Experiments for Psychology and Neuroscience

Content: how to create visual and auditory stimuli, query a database to find words with certain characteristics, program real-time experiments that record participants responses, perform basic data processing to analyse results,…

Programming language: mostly Python (+ pygame + expyriment + pandas + seaborn) (and a bit of R, maybe Javascript this year).

Timetable: S2. During the first half of the semester, the course consists of short lectures followed by hands-on exercises, in the second half of the semester, students have to code an experiment of their choice.

Evaluation: PASS/FAIL based on the realisation of a project (students must provide a running, clean, code, correctly documented on github)

Important note: This lecture is of interest even for students who are not going to need to program experiments as it show how to use revision control software such as git, and clean code methods that promote scientific reproducibility, and train them to improve their coding skills in general…

Instructors: Christophe Pallier & Maxime Cauté

Course materials: check out https://pcbs.rtfd.io/   and https://github.com/chrplr/PCBS