Say hello to Python

Recap of the course so far

- 4+ weeks of functional with OCaml
- Next: 3 weeks of OO with Python

News

- Midterm Grades Will Be Up Soon
  - Stay tuned for viewing hours

- PA4 Due Tomorrow (Fri 5p)
  - Ocaml

- PA5 Is Up, Due 5/13
  - Python
OO at the highest level

• What is OO programming?

• Answer:
  - Objects
  - message sends
  - dynamic dispatch

Just to whet your appetite

• Say we have objects, like cars, ducks, pig, cell_phones

• Say we have a message name: make_some_noise

• Each object has its own implementation for make_some_noise: these are traditionally called methods.

  - car: vroom vroom,
  - pig: oink oink,
  - duck: quack quack

• Can send make_some_noise to any object. Depending on the actually run-time object, we’ll get a different noise!
**OO programming**

- **Message:**
  - the name of an operation
- **Method:**
  - the implementation of an operation
- **Dynamic dispatch:**
  - dynamic type of object determines
  - which method is run for given message send

- These are the core ideas of OO

---

**This brings us to Python...**

- We’ll use Python as our vehicle for OO programming
- Fun and useful language

---

**Python**

- Python has a very relaxed philosophy
  - if something "can be done" then it is allowed.
- Dynamic types + Everything is an object
  - very flexible
  - very intuitive code

---

**No static types**

- No static type system to "prohibit" operations.
- No more of that OCaml compiler giving you hard-to-decipher error messages!
No static types: but what instead?

- **Dynamic typing**
- At runtime, every "operation" is translated to a method call on the appropriate object. If the object supports the method, then the computation proceeds.
- **Duck-typing**: if it looks like a duck, quacks like a duck, then it is a duck!

Similarities to ML

- **Uniform model**
  - everything is an object, including functions
- **Pass functions around**
  - functions are objects!
- **Supports functional programming**
  - map and fold

Other cool things about Python

- A lot of stuff that you may first think is a "language feature" is actually just translated under the hood to a method call...
- Very widely used, supported.
- Has libraries for all sorts of things.

Ok, let’s start playing with Python!

- Like Perl, python is a "managed" or "interpreted" language that runs under the python environment, i.e. not compiled to machine code.
- Makes it convenient to rapidly write, check-in and test code!
Ways to run Python code

• At an interactive Python prompt
  - like "read-eval-print" loop of ML
• As shell scripts
• As stand-alone programs
  - run from the shell.

Let’s fire it up!

• Ok, let’s give it a try...
• See lec11.py file for the rest...