Advanced Brian 2

Runtime and standalone modes

- Runtime
  - Python / Numpy
  - C++ / Weave (doesn’t work on Python 3)
  - Cython
- Standalone
  - C++
  - GPU: GeNN (mostly working)
  - GPU: Moritz et al. (TODO)
  - Android: Java / Renderscript (in progress)
  - Other targets planned (NeMo, OpenCL, FPGA)

Runtime code generation

Select code generation target:

By default, selects best available (weave > cython > numpy)

```
prefs.codegen.target = 'numpy'
prefs.codegen.target = 'weave'
prefs.codegen.target = 'cython'
```

Save preference per-script, per-directory or per-user:

Script: `script.py: prefs.codegen.target = 'weave'`

Directory: `./brian_preferences: codegen.target = 'weave'`

User: `~/.brian/preferences: codegen.tareget = 'weave'`

Additional preferences for things like preferred compiler, compiler arguments, etc.

Demo!

Extending Brian 2: new languages/devices

- Won't go through all the details (depending on time)
- Write a new language generator
  - Syntax translation (using `NodeRenderer`)
  - Translate basic language elements using `CodeGenerator`
    - Data types, scalars, constants, arrays, dynamic arrays
  - Implement templates for supported Brian objects
- Write a new runtime mode
  - Write a language generator if necessary
  - Implement the `CodeObject` (handles compiling, running, etc.)
- Write a new device
  - Write a language generator if necessary
  - Implement a `Device` object
- May not be as much work as it seems! (build on existing C++ code)