IPYTHON NOTEBOOK SLIDES

Andrew Walker / @walkera101
WHY THIS TALK

• Putting technical slides together is horrible experience
• Most tools aren't version control friendly
• LaTeX slides are a huge hassle to maintain
• IPython notebooks for presentations look even worse
ASSUMED KNOWLEDGE

- Basic Python
- User of IPython
- JSON
- Markdown
- Minimal HTML / CSS
IPYTHON
NBCONVERT
!ipython nbconvert --to slides notebook-to-reveal.ipynb

[NbConvertApp] Converting notebook notebook-to-reveal.ipynb to slides
[NbConvertApp] Writing 214223 bytes to notebook-to-reveal.slides.html
WHAT'S WRONG WITH THE IPYTHON THEME?

- poor use of space
- small font sizes
- aliasing artefacts on fonts
- poor typographical conventions
- optimised for interacting not communicating
- they're ugly
IPYTHON NOTEBOOKS ARE JUST JSON
```json
{
    "cell_type": "code",
    "execution_count": 6,
    "metadata": {
        "collapsed": false
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "hello world\n"
            ]
        }
    ],
    "source": [
        "print 'hello world"
    ]
}
```
NBConverter IS LIKE XSLT FOR JSON

- take a template and data and transform to a new format
- templates are in IPython/nbconvert/templates/
- take a look at slides_reveal.tpl
CUSTOMIZING THE RESULTS

- Custom CSS
- Custom (jinja) template
- Custom (python) configuration file
CUSTOM CSS

# in custom.css
div.inner_cell {
    background-color: red;
}

HOW DOES THIS ALL WORK?
JINJA
from jinja2 import Environment

env = Environment(trim_blocks = True, lstrip_blocks = True)
t = env.from_string('hello {{ name }}')
print t.render(name = 'MPUG')
print t.render(name = 'Ada')

hello MPUG
hello Ada
env = Environment(trim_blocks = True, lstrip_blocks = True)
t = env.from_string(''
{% for slide in slides %}
- Slide {{ slide }}
{% endfor %}'
)print t.render(slides = range(1,4))

<h1>Slide 1</h1>
<h1>Slide 2</h1>
<h1>Slide 3</h1>
MISTUNE
from mistune import markdown

print markdown('# Heading 1')
print markdown('## Heading 2')
print markdown('### Heading 3')

<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
s = '''
- foo
- bar
- baz
'''
print markdown(s)

<ul>
<li>foo</li>
<li>bar</li>
<li>baz</li>
</ul>
CONCLUSION

- Learnt about a few of the challenges facing IPython
- Developed a new IPython reveal.js template
  - New template optimized for communication, rather than interaction
  - Simpler to read, understand and modify
  - Not tied to specific versions of reveal.js