# Documenting Your Project



Reindert-Jan Ekker

@rjekker http://nl.linkedin.com/in/rjekker



### Overview



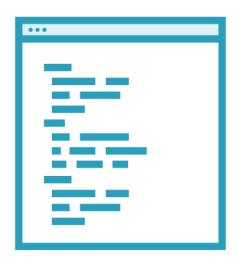
**Docstrings and standards** 

Sphinx: generating HTML from source documentation

reStructuredText



### PFP 257



#### Semantics and conventions for docstrings

#### **Docstrings**

- String as first statement of a module, function, module or class
- Becomes the \_\_doc\_\_ attribute



```
def function(a, b):
    """Do X and return a list."""
```

Simple Docstring Example

Always use """three double quotes""""

Phrase ending in a period

Methods: specify return value



## A Multi-line Docstring

```
def complex(real=0.0, imag=0.0):
    """Form a complex number.
    Keyword arguments:
    real -- the real part (default 0.0)
    imag -- the imaginary part (default 0.0)
    11 11 11
    if imag == 0.0 and real == 0.0:
        return complex_zero
```



### Demo



#### **Sphinx**

- Python documentation generator
- De-facto standard
- reStructuredText -> HTML, PDF, etc.
- Extract docstrings from code



## Running Sphinx



#### First create docs/ folder

- Run commands from in there

sphinx-quickstart

make html

Configuration is stored in conf.py



### reStructuredText



Plain-text based markup

Paragraphs separated by a single blank line

- Indentation matters



### reStructuredText

```
=======
```

### A Chapter

=======

A Section

------

A Subsection

~~~~~~~~

- \* Bulleted list item one
- \* Bulleted list item two

- + Milk
- + Butter

```
`Link text <http://example.com/>`_
http://www.example.com
```



## reStructuredText: Code Examples

```
Code examples start with ::, delimited by indentation. For example see here::
```

```
def hi():
    print("Hello, World!")
```



### Demo



### **Apidoc**

- Sphinx extension
- Generate docs from Python source



```
# Generate docs for mypackage into docs folder
sphinx-apidoc -o docs mypackage
# Get help
sphinx-apidoc --help
# Generate everything from scratch
# (no need for sphinx-quickstart)
sphinx-apidoc --full -o docs mypackage
```

## Apidoc

Extracts docstrings from Python code

Creates reStructuredText with directives for autodoc

Autodoc needs to be enabled in conf.py



```
:class:`Player`
```

:class:`gamedemo.Player`

:meth:`Weapon.attack`

:func:`mymodule.func`

:module:`mymodule`

- **◄** Sphinx allows linking to classes
- This will create hyperlinks into docs generated by autodoc
- Links to methods, functions, modules



## Documenting a Function

```
def is_alive(player):
    """Determine whether the player is alive
    :param player: a Player object
    :return: True when the player is alive, False otherwise
    11 11 11
return player.health > 0
```



## Summary



**Docstrings and standards** 

Sphinx: generating HTML from source documentation

reStructuredText

