

# Boring Object Orientation

Boring is better than interesting

Moshe Zadka – <https://cobordism.com>

2019

# Inheritance-as-API: Examples in the wild

```
# From the Twisted tutorial
class FingerProtocol(basic.LineReceiver):
    def lineReceived(self, user):
        self.transport.write(b"No such user\r\n")
        self.transportloseConnection()
```

## Inheritance-as-API: Examples in the wild

```
# From the Django tutorial
class IndexView(generic.ListView):
    template_name = 'polls/index.html'
    context_object_name = 'latest_question_list'

    def get_queryset(self):
        """Return the last five published questions
        return Question.objects.order_by('-pub_date
```

## Inheritance-as-API: Examples in the wild

# From the Jupyter documentation

```
class EchoKernel(Kernel):
    implementation = 'Echo'
    implementation_version = '1.0'
    language = 'no-op'
    language_version = '0.1'
    language_info = {
        'name': 'Any text',
        'mimetype': 'text/plain',
        'file_extension': '.txt',
    }
    banner = "Echo kernel - as useful as a parrot"

    def do_execute(self, code, silent, store_history,
                  allow_stdin=False):
        if not silent:
            stream_content = {'name': 'stdout', 'text': code}
```

# Inheritance-as-API: Issues

"Shared everything"

# Composition

- ▶ Define *interface*
- ▶ Useful behavior in *referred class*

## Composition: Simple example

```
class IMovable(interface.Interface):  
    x_position = interface.Attribute()  
    y_position = interface.Attribute()  
    def tick():  
        pass
```

## Composition: Simple example

```
@interface.implementer(IMovable)
@attr.s(auto_attribs=True):
class StraightLine:
    dx: float
    dy: float
    x_position: float
    y_position: float
    def tick(self):
        self.x_position += dx
        self.y_position += dy
```



## Composition: Simple example

```
@interface.implementer(IMovable)
@attr.s(auto_attribs=True):
class Sprite:
    _movable: IMovable
    @property
    def x_position(self):
        return self._movable.x_position
    @property
    def y_position(self):
        return self._movable.y_position
    def tick(self):
        return self._movable.tick()
```