RSPEC MOCKS VS STUBS

STUBS

```
kanye.stub(:best_film_clip_of_all_time).and_return(:beyonce)
```

- Stubs won't complain if they're not called
- Use stubs when we're just testing the state e.g. we just care about the end result not about how we get there

STUBS

```
d = double('service', message1:  true)

allow(d).to receive(:message2).and_return(:value)

allow(real_object).to receive(:message).and_return(:value)
```

MOCKS

AKA MESSAGE EXPECTATIONS

myley.should_receive(:twerking).and_return('wtf')

- Mocks or message expectations, will chuck a fit if not called
- This is great because it confirms behavior, that our methods are indeed getting called and returning the values we expect
- tightly coupled to implementation, which is probably the point if you're using mocks over a stub.
- Intent is clear with expect syntax vs foo.should_receive

MOCKS

AKA MESSAGE EXPECTATIONS

```
expect(foo).to receive(:bar)
expect(foo).to receive(:bar).with(:buzz)
expect(foo).to receive(:bar).exactly(3).times
```

DOUBLES

```
thingy = double('thingy')
```

- stands in for another object
- mock('object') and stub('object') are aliases of double and have been removed as of 3.0

ANY QUESTIONS?