

# Hooks & Tags

LEVEL 4 •

#### Hooks

```
spec/models/zombie spec.rb
describe Zombie do
  let(:zombie) { Zombie.new }
  it 'is hungry' do
     zombie.hungry!
     zombie.should be_hungry
  end
  it 'craves brains' do
    zombie.hungry!
    zombie.should be_craving_brains
  end
end
```

don't repeat yourself



#### Hooks

spec/models/zombie\_spec.rb

```
describe Zombie do
  let(:zombie) { Zombie.new }
  before { zombie.hungry! }
  it 'is hungry' do
    zombie.should be_hungry
  end
  it 'craves brains' do
    zombie.should be_craving_brains
  end
end
```

```
run before each example
     before(:each)
   run once before all
     before(:all)
    run after each
     after(:each)
      run after all
      after(:all)
```



#### Hooks in context

spec/models/zombie\_spec.rb

```
describe Zombie do
  let(:zombie) { Zombie.new }
  before { zombie.hungry! }
  it 'craves brains' do
    zombie.should be_craving_brains
  end
  context 'with a veggie preference' do
    it 'still craves brains' do
      zombie.hungry!
      zombie.vegetarian = true
    end
    it 'craves vegan brains' do
      zombie.hungry!
      zombie.vegetarian = true
    end
  end
```



#### Hooks in context

spec/models/zombie\_spec.rb

```
describe Zombie do
  let(:zombie) { Zombie.new }
  before { zombie.hungry! } 
  it 'craves brains' do
    zombie.should be_craving_brains
  end
  context 'with a veggie preference' do
    it 'still craves brains' do
      zombie.vegetarian = true
    end
    it 'craves vegan brains' do
      zombie.vegetarian = true
    end
  end
```

available within contexts



#### Hooks in context

```
spec/models/zombie_spec.rb
```

```
describe Zombie do
  let(:zombie) { Zombie.new }
  before { zombie.hungry! }
  it 'craves brains' do
    zombie.should be_craving_brains
  end
  context 'with a veggie preference' do
    before { zombie.vegetarian = true }
    it 'still craves brains' do
    end
    it 'craves vegan brains' do
    end
  end
```

available to all examples within this context



```
describe Zombie do
  it 'should not have a pulse' do
    zombie = Zombie.new
    zombie.pulse.should == false
  end
end
```

spec/models/vampire\_spec.rb

spec/models/zombie spec.rb

```
describe Vampire do
  it 'should not have a pulse' do
    vampire = Vampire.new
    vampire.pulse.should == false
  end
end
```

Duplication!



spec/models/zombie\_spec.rb

describe Zombie do
 it\_behaves\_like 'the undead'
end

used to call shared examples

spec/models/vampire\_spec.rb

end

describe Vampire do
 it\_behaves\_like 'the undead'
end

let's build our shared example

spec/support/shared\_examples\_for\_undead.rb

shared\_examples\_for 'the undead' do
 it 'does not have a pulse' do
 subject.pulse.should == false
end

TESPEE C

refers to the implicit subject

spec/models/zombie\_spec.rb

```
describe Zombie do
  it_behaves_like 'the undead' do
  let(:undead) { Zombie.new }
  end
end
```

spec/support/shared\_examples\_for\_undead.rb

```
shared_examples_for 'the undead' do
  it 'does not have a pulse' do
  undead.pulse.should == false
  end
end
```

we can access the 'undead' we defined in 'let'



spec/models/zombie\_spec.rb

```
describe Zombie do
  it_behaves_like 'the undead', Zombie.new
end
```

spec/support/shared\_examples\_for\_undead.rb

```
shared_examples_for 'the undead' do lundead!
  it 'does not have a pulse' do
    undead.pulse.should == false
  end
end
```



```
spec/models/zombie_spec.rb
```

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', focus: true do
        it 'still craves brains'
        it 'prefers vegan brains', vegan: true
    end
end
end
end
```

```
$ rspec --tag focus spec/lib/zombie_spec.rb
Zombie
  with a veggie preference
    still craves brains
    prefers vegan brains
Finished in 0.00125 seconds
2 examples, 0 failures
```

only: focus examples ran



```
spec/models/zombie_spec.rb
```

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', focus: true do
        it 'still craves brains'
        it 'prefers vegan brains', vegan: true
    end
  end
end
```

```
spec/spec_helper.rb
```

```
RSpec.configure do | config|
config.filter_run focus: true
config.run_all_with_everything_filtered = true

runs everything if none match
end
```



spec/models/zombie\_spec.rb

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', focus: true do
        it 'still craves brains'
        it 'prefers vegan brains', vegan: true
    end
  end
end
```

```
$ rspec spec/lib/zombie_spec.rb
Zombie
  with a veggie preference
    still craves brains
    prefers vegan brains
Finished in 0.00125 seconds
2 examples, 0 failures
```

still filtered to : focus



```
spec/models/zombie_spec.rb
```

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', slow: true do
       it 'still craves brains'
       it 'prefers vegan brains'
    end
end
end
end
```

```
$ rspec --tag ~slow spec/lib/zombie_spec.rb
Zombie
wants brains
```

Finished in 0.00125 seconds 1 examples, 0 failures

slow examples didn't run



spec/models/zombie spec.rb

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', slow: true do
      it 'still craves brains'
      it 'prefers vegan brains'
    end
  end
end
```

```
skip slow examples in default runs
spec/spec helper.rb
RSpec.configure do Iconfigl
   config.filter_run_excluding slow: true
   config.run_all_with_everything_filtered = true
end
```



spec/models/zombie\_spec.rb

```
describe Zombie do
  context 'when hungry' do
    it 'wants brains'
    context 'with a veggie preference', slow: true do
       it 'still craves brains'
       it 'prefers vegan brains'
    end
end
```

```
$ rspec --tag slow spec/lib/zombie_spec.rb
Zombie
  with a veggie preference
    still craves brains
    prefers vegan brains
Finished in 0.00125 seconds
2 examples, 0 failures
```

still filtered to : focus

