



## CONVENTIONS

### Reporting Errors

---

---

---

---

---

---



## ERROR MESSAGES

- Use **gists** for your error messages
- Include your **OS, Ruby Version, and Gems**

---

---

---

---

---

---



## CONVENTIONS

### 2 spaces indent

---

---

---

---

---

---



CONVENTIONS

## Naming Conventions

---

---

---

---

---

---



!CAMELCASE

~~variableName~~

---

---

---

---

---

---



VARIABLES

variable\_name

---

---

---

---

---

---



METHODS

method\_name

---

---

---

---

---

---



CONVENTIONS

Folder Structure

---

---

---

---

---

---

*project/*

lib/

spec/ or test/

Gemfile

Guardfile

Rakefile

README.md

---

---

---

---

---

---




---



---



---



---



---



---




---



---



---



---



---



---




---



---



---



---



---



---



---

---

---

---

---



---

---

---

---

---



---

---

---

---

---



## SUPERHERO

```

3) Superhero should have a nickname
Failure/Error: subject.nick_name.should == "The Caped Crusader"
NoMethodError:
undefined method `nick_name' for #<Superhero:0x007f95532cc690>
# ./spec/06_struct_spec.rb:13:in `block (2 levels) in <top (required)>'

4) Superhero alter_ego should respond_to? alter_ego
Failure/Error: subject.alter_ego.should respond_to :alter_ego
NoMethodError:
undefined method `alter_ego' for #<Superhero:0x007f95532d1280>
# ./spec/06_struct_spec.rb:19:in `block (3 levels) in <top (required)>'

5) Superhero alter_ego should have an alter ego
Failure/Error: subject.alter_ego.should_not be_nil
NoMethodError:
undefined method `alter_ego' for #<Superhero:0x007f95532d1280>
# ./spec/06_struct_spec.rb:19:in `block (3 levels) in <top (required)>'

6) Superhero alter_ego should have the correct alter ego
Failure/Error: subject.alter_ego.should == bruce_wayne
NoMethodError:
undefined method `alter_ego' for #<Superhero:0x007f95532d0420>
# ./spec/06_struct_spec.rb:125:in `block (3 levels) in <top (required)>'

```

---



---



---



---



---



---

## LIVE CODING

---



---



---



---



---



---

## QUESTIONS?

---



---



---



---



---



---

## RESPOND\_TO?

Returns *true* if *obj* responds to the given method. Private methods are included in the search only if the optional second parameter evaluates to true.

<http://rubydoc.info/stdlib/core/1.9.2/Object>

---

---

---

---

---

---

---

## respond\_to?

VS

## kind\_of?

<http://rubydoc.info/stdlib/core/1.9.2/Object>

---

---

---

---

---

---

---

a | |= b

---

---

---

---

---

---

---

**a = a || b**

---

---

---

---

---

---

**a || a = b**

<http://www.rubyinside.com/>

---

---

---

---

---

---



**Glee Box**

---

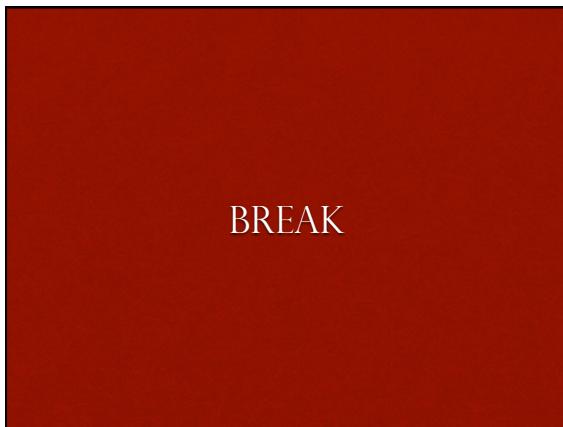
---

---

---

---

---



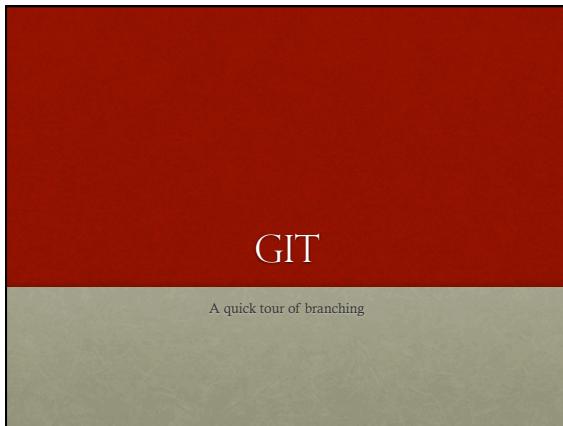
---

---

---

---

---



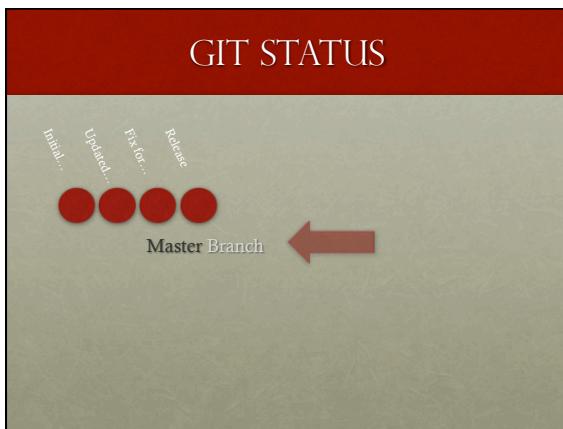
---

---

---

---

---



---

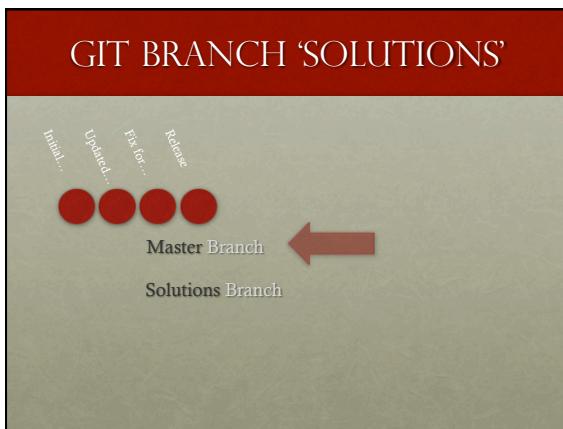
---

---

---

---

---



---

---

---

---

---

---



---

---

---

---

---

---



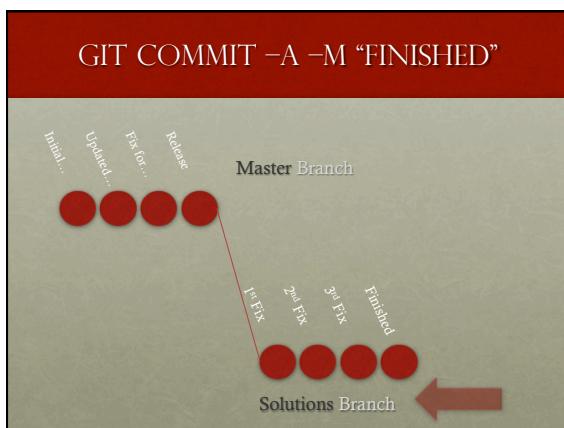
---

---

---

---

---



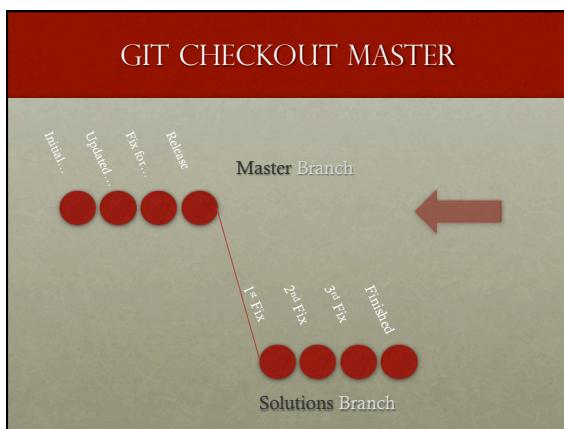
---

---

---

---

---



---

---

---

---

---

## METHODS

```
puts "Name: #{batman.name}"
puts "Nickname: #{batman.nick_name}"
puts "Origin: #{batman.origin}"
puts "Alter Ego: #{batman.alter_ego}"
```

```
puts "Name: #{batman.name}"
puts "Nickname: #{batman.nick_name}"
puts "Origin: #{batman.origin}"
puts "Alter Ego: #{batman.alter_ego}"

puts "Name: #{superman.name}"
puts "Nickname: #{superman.nick_name}"
puts "Origin: #{superman.origin}"
puts "Alter Ego: #{superman.alter_ego}"
```

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## 1.Duplication

---

---

---

---

---

---

```
[ batman, superman ].each do |hero|
  puts "Name: #{hero.name}"
  puts "Nickname: #{hero.nick_name}"
  puts "Origin: #{hero.origin}"
  puts "Alter Ego: #{hero.alter_ego}"
end
```

---

---

---

---

---

---

## 2.Clarity

---

---

---

---

---

---

```
display_superhero batman
```

---

---

---

---

---

---

```
def display_superhero(hero)
  puts "Name: #{hero.name}"
  puts "Nickname: #{hero.nick_name}"
  puts "Origin: #{hero.origin}"
  puts "Alter Ego: #{hero.alter_ego}"
end

[ superman, batman ].each do |hero|
  display_superhero hero
end
```

---

---

---

---

---

---

```
display_superheroes [ superman, batman ]
```

---

---

---

---

---

---

```
def display_superheroes(heroes)
  heroes.each do |hero|
    puts "Name: #{hero.name}"
    puts "Nickname: #{hero.nick_name}"
    puts "Origin: #{hero.origin}"
    puts "Alter Ego: #{hero.alter_ego}"
  end
end

display_superheroes [ superman, batman ]
```

---

---

---

---

---

---

```
display_superheroes superman
```

---

---

---

---

---

---

```
Array()
```

---

---

---

---

---

---

```
def display_superheroes(heroes)
  Array(heroes).each do |hero|
    puts "Name: #{hero.name}"
    puts "Nickname: #{hero.nick_name}"
    puts "Origin: #{hero.origin}"
    puts "Alter Ego: #{hero.alter_ego}"
  end
end
```

---

---

---

---

---

---

display\_superheroes

---

---

---

---

---

---

```
def display_superheroes(heroes = nil)
  Array(heroes).each do |hero|
    puts "Name: #{hero.name}"
    puts "Nickname: #{hero.nick_name}"
    puts "Origin: #{hero.origin}"
    puts "Alter Ego: #{hero.alter_ego}"
  end
end
```

---

---

---

---

---

---

```
def display_superheroes(heroes = superman)
  Array(heroes).each do |hero|
    puts "Name: #{hero.name}"
    puts "Nickname: #{hero.nick_name}"
    puts "Origin: #{hero.origin}"
    puts "Alter Ego: #{hero.alter_ego}"
  end
end
```

---

---

---

---

---

---

```
def display_superheroes(heroes = $heroes)
  Array(heroes).each do |hero|
    puts "Name: #{hero.name}"
    puts "Nickname: #{hero.nick_name}"
    puts "Origin: #{hero.origin}"
    puts "Alter Ego: #{hero.alter_ego}"
  end
end
```

---

---

---

---

---

---

\$ heroes = []

---

---

---

---

---

---

\*Available in all scopes. Use with discretion.

```
SuperHeroes = []
```

Constants with a Capital Letter

---

---

---

---

---

---

```
display_superheroes superman, wonderwoman,  
batman, ...
```

---

---

---

---

---

---

```
def display_superheroes(*heroes)  
  heroes.each do |hero|  
    puts "Name: #{hero.name}"  
    puts "Nickname: #{hero.nick_name}"  
    puts "Origin: #{hero.origin}"  
    puts "Alter Ego: #{hero.alter_ego}"  
  end  
end
```

---

---

---

---

---

---

\* splat

All remaining parameters are simply placed in an array

---

---

---

---

---

---

Finding

---

---

---

---

---

---

find\_superheroes\_with\_nemesis joker

---

---

---

---

---

---

[ batman, robin ]

---

---

---

---

---

---

```
def find_superheroes_with_nemesis(nemesis)
  SuperHeroes.find_all do |hero|
    hero.nemesis == nemesis
  end
end
```

---

---

---

---

---

---

find\_superheroes\_with\_origin 'Krypton'

---

---

---

---

---

---

```
[ superman, supergirl ]
```

---

---

---

---

---

---

```
def find_superheroes_with_origin(origin)
  SuperHeroes.find_all do |hero|
    hero.origin == origin
  end
end
```

---

---

---

---

---

---

## Named Parameters

---

---

---

---

---

---

```
find_superheroes { :origin => 'Krypton',
                   :nemesis=> joker }
```

---

---

---

---

---

---

```
find_superheroes :origin => 'Krypton',
                  :nemesis => joker
```

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |key,value|
    SuperHeroes.find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end
```

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |origin,'Krypton'|
    SuperHeroes.find_all do |hero|
      hero[:origin] == 'Krypton'
    end
  end.flatten.uniq
end

[ [ superman, supergirl ] ]
```

---

---

---

---

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |:nemesis, joker|
    SuperHeroes.find_all do |hero|
      hero[:nemesis] == joker
    end
  end.flatten.uniq
end

[ [ superman, supergirl ], [ batman, robin ] ]
```

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |key,value|
    SuperHeroes.find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end
```

---

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |key,value|
    SuperHeroes.find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end

[ superman, supergirl, batman, robin ]
```

---

---

---

---

---

---

```
def find_superheroes(criteria)
  criteria.map do |key,value|
    SuperHeroes.find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end

[ superman, supergirl, batman, robin ]
```

---

---

---

---

---

---

## Bonus

```
find_superheroes :origin => 'Gotham City',
                 :nick_name => 'Boy Wonder'
```

```
[ robin ]
```

---

---

---

---

---

---

## Defining a Methods for one object

```
def find_superheroes(criteria)
  criteria.map do |key,value|
    SuperHeroes.find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end

[ superman, supergirl, batman, robin ]
```

```
def SuperHeroes.find_by(criteria)
  criteria.map do |key,value|
    find_all do |hero|
      hero[key] == value
    end
  end.flatten.uniq
end
```

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

SuperHeroes.find\_by :name => 'Supergirl'

---

---

---

---

---

---

Questions?

---

---

---

---

---

---