1. An HTTP POST request is used to create a new entity, while the PUT request is used to modify an existing one. (Also I believe a post can hold a fence but a put cannot)

2. This is a relative address.

3. An absolute URL specifies the exact entire address of the link as well as the protocol, whereas a relative URL only specifies the URL relative to the which page the link is on.

4. It would generate a GET request.

5. Yes, the querystring is "request\_type=PUT"

6. Possibly a tooltip or other information saying exactly what clicking on the link would do.

7. The web browser retrieves and presents information that the user wishes to see. The database contains the data and relations that the web application can use to generate the requested information.

8. No, there is nothing in the response header about the content that is being returned.

9.

class Troll

attr\_accessor :ugliness, :smelliness, :strength

def initialize(grunt = "UNGAH")

@grunt = grunt

@ugnliness, @smelliness, @strength = nil

end

def self.propegate

Troll.new("eegah")

end

def speak

42.times do

puts @grunt

end

end

def reverse

puts @grunt.reverse

end

end

10. it's missing a method with the name "fight"

11. This isn't really polymorphism. this function is simply checking if a method with that particular name exists.

12. Boolean (True / False) return value.

13. The bang version (do\_this!) is considered the "dangerous" version of do\_this. Often that means that the instance will be changed by the function.

14. Ruby uses dynamic typing. This means that most of its type checking is done at run time instead of compile time.

15. %w( master rails and then try another framework you'll never go back) would yield an array of words. ["master", "rails", "and", "then", "try", "another", "framework", "you'll", "never", "go", "back"]

16. Both code snippets would do the same thing.

17. In ruby, the function does not need an explicit return statement. Ruby will return the value of the last statement by default.

18. before\_create, after\_create, after\_save, after\_commit

19.

|  |  |  |
| --- | --- | --- |
| HTTP method | Controller Action | CRUD Operation |
| Get | Index | Read |
| Get | New | Read |
| Post | Create | Create |
| Get | Edit | Read |
| Put | Update | Update |
| Delete | Destroy | Delete |

20. Browser's generally don't support actual put and delete requests, so rails gets around this by simulating them.

21. The main difference between the development and production environments is that in development the application code is reloaded for each request, whereas in production the code is only reloaded by a server restart.

22. When there would only be one instance, IE only one GeoCoder exists.

23. A helper method is a function that is called from a view. They should be written by the user to keep logic out of the views and in external functions whenever some logic is required.

24. A foreign key constraint must be specified so that AR knows how to assign / infer the proper relationship. This should correspond with the "has\_many" / "belongs\_to" type declarations in the models

25. The queries are valid. Because each flower's bee's hive name and bee's name is displayed on a separate page the queries all provide unique data.