

PLANT SHOP

Tash Gibson
Ruby Terminal app
T1A3

PLANT SHOP

Overview:

The plant shop app is designed to find a suitable indoor plant for the user.

- size of plant,
- ease of care,
- light levels.

The user will be asked some criteria about the growing conditions and how much care they want to put in. This will result in a suggestion of an indoor plant, with a price.



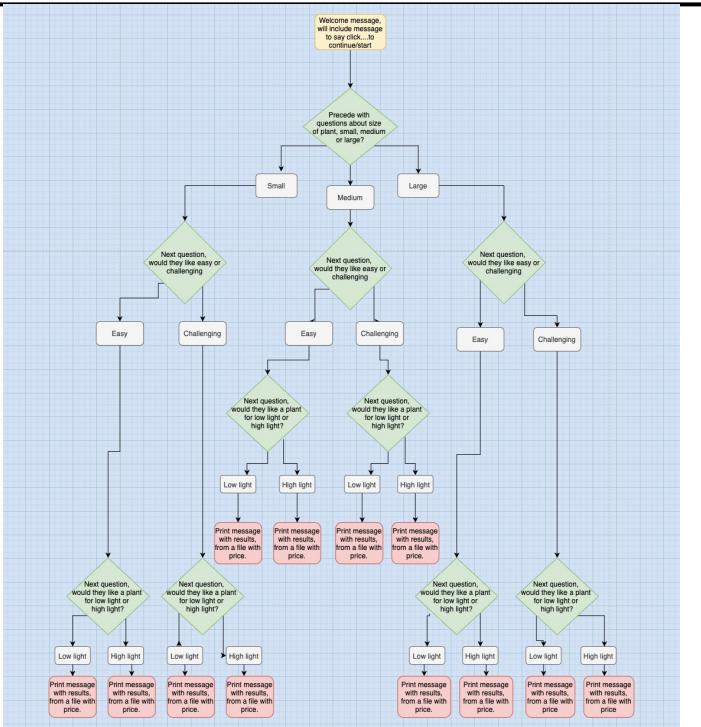
PLANT SHOP

The app

- When the user opens the app there will be a welcome message and the user will be asked for their name.
- They will be asked some questions about the plant requirements.
- The answers will then be matched to a plant.
- The user will be given a print of a plant suggestion with a price.
- There will be an option of all the plants printed out.



PLANT SHOP FLOW CHART



FEATURES

List of features

- List of questions
- How I will suggest a plant for the user
- Print out of plant suggestion with price
- Optional print out of all plants and their prices

How the app is used

- The app functions by the user responding to prompts to answer questions, there may be some extra questions for some optional additional information to be printed.



PLANT SHOP

Method with the welcome message and user name input.
Method for error handling.

```
def welcome
    puts "Welcome to the indoor plant shop.".colorize
        (:yellow)
    puts "What is your name?"
    print ">"
    user_name = gets.chomp()
    validate_name(user_name)
    puts ("Hello " + user_name + ", please answer
        some questions to find your perfect plant match.")
    user_name
end

#specify an exception type
def validate_name(name)
    name = name.strip
    raise ArgumentError, "Name must not be empty" if
        name.empty?
    name
end
```



PLANT SHOP

Method containing questions for the user.

```
def questions(plant_selection)
  puts "What size plant do you want, small, medium
or large?"
  print ">"
  plant_selection.answers[0] = gets.chomp
  puts "Do you want easy or challenging?"
  print ">"
  plant_selection.answers[1] = gets.chomp
  puts "Do you want a plant for high light or low
light?"
  print ">"
  plant_selection.answers[2] = gets.chomp
end
```



PLANT SHOP

Class with an array of hashes – containing plant lists.

```
class Plantlist
attr_reader :plant_choice
def initialize

@plant_choice = [ { :size => "small", :care => "easy", :light => "low",
:plant_name => "Sansevera spp / Snake Plant", :price => "$25"},

{ :size => "small", :care => "challenging", :light => "low", :plant_name =>
"Adiantum raddianum / Maidenhair fern", :price => "$18"},

{ :size => "small", :care => "easy", :light => "high", :plant_name =>
"Epipremnum aureum / Devil's Ivy", :price => "$18"},

{ :size => "small", :care => "challenging", :light => "high", :plant_name =>
"Calathea spp / Calathea", :price => "$25"},

{ :size => "medium", :care => "easy", :light => "low", :plant_name =>
"Monstera deliciosa / Swiss Cheese Plant", :price => "$30"},

{ :size => "medium", :care => "challenging", :light => "low", :plant_name =>
"Zanzibar Gem / Zuzu plant", :price => "$40"},

{ :size => "medium", :care => "easy", :light => "high", :plant_name =>
"Philodendron Xanandu / Xanandu", :price => "$35"},

{ :size => "medium", :care => "challenging", :light => "high", :plant_name =>
"Nephrolepis exelata / Boston fern", :price => "$25"},
```



PLANT SHOP

Method to match users answers to the corresponding hash.

```
# calling a plant selection to print to user

def plant_suggest
object = Plantlist.new
pp object.plant_choice.select { |hash| (hash.values
& answers) == answers}

end
```



PLANT SHOP

Class containing methods to save users input to a File.

```
class PlantSpecks
  attr_accessor :answers
  def initialize(name)
    @name = name
    @file_path = "./PlantSpecks/#{name}.txt"
    @answers = []
  end

  def size
    puts "#{@name}'s Do you want a ({1}small), ({2}medium), ({3}large) plant?"

    if File.exist?(@file_path)
      File.open(@file_path, 'r') do |file|
        file.each_line do |line|
          puts "#{file.lineno}, #{line}"
        end
      end
    else
  
```





PLANT SHOP

Overview of the code and important parts

- Class: array of hashes containing plants and corresponding values.
- Class: creates a file to store user details plus a method to compare users answers to the plant hashes.
- Testing: error handling, rspec tests.
- Method: getting user input.



PLANT SHOP

Challenges, ethical issues and favourite parts.

- Challenges: RUBY
- Ethical issues: Imposter syndrome – will I catch on quickly enough? Need to practice logic.
- Favourite parts: Project planning – especially the trello board and flowchart and when the code works.