



# Philosophize this!

---

A terminal app  
Armstrong Ashton-  
Lewis

# The Idea

---

- A philosophy app that can help users to learn about some of the major philosophers of human history.



# How best to implement?

---

- Perfect opportunity for a Trello board
- Utilize the rubric and comb through to make sure you are hitting all points while keeping things as simple as possible
- KISS

Boards Jump to... Trello + ⓘ 🔔 AA

Board Philosophize\_This Army's army Free Workspace visible Invite

Documentation To Do's

- Flow chart (May 20)
- design a software development plan (May 20)
- Develop a list of features that will be included in the application (May 20)
- Develop a diagram which describes the control flow of your application. (May 20)
- Develop an outline of the user interaction and experience for the application. (May 20)
- Design help documentation which includes a set of instructions which accurately describe how to use and install the application. (May 20)

Code To Do's

- Apply DRY (Don't Repeat Yourself) coding principles to all code produced. (May 23)
- Apply all style and conventions for the programming language consistently to all code produced. (May 23)
- Creates an application which runs without error and has features that are consistent with the development plan. (May 23)
- Design TWO tests which check that the application is running as expected. (May 23)
- Utilise source control throughout the development of the application by: (May 23)

Presentation To Do's

- Present your Terminal application to the class (May 24)
- A walk-through of your Terminal application, its features and how it used (May 24)
- A walk-through of the logic of your Terminal application and code (May 24)
- A review of your development/build process including challenges, ethical issues, favourite parts, etc (May 24)
- An overview of your Terminal application (May 24)
- An overview of your code (May 24)

In progress

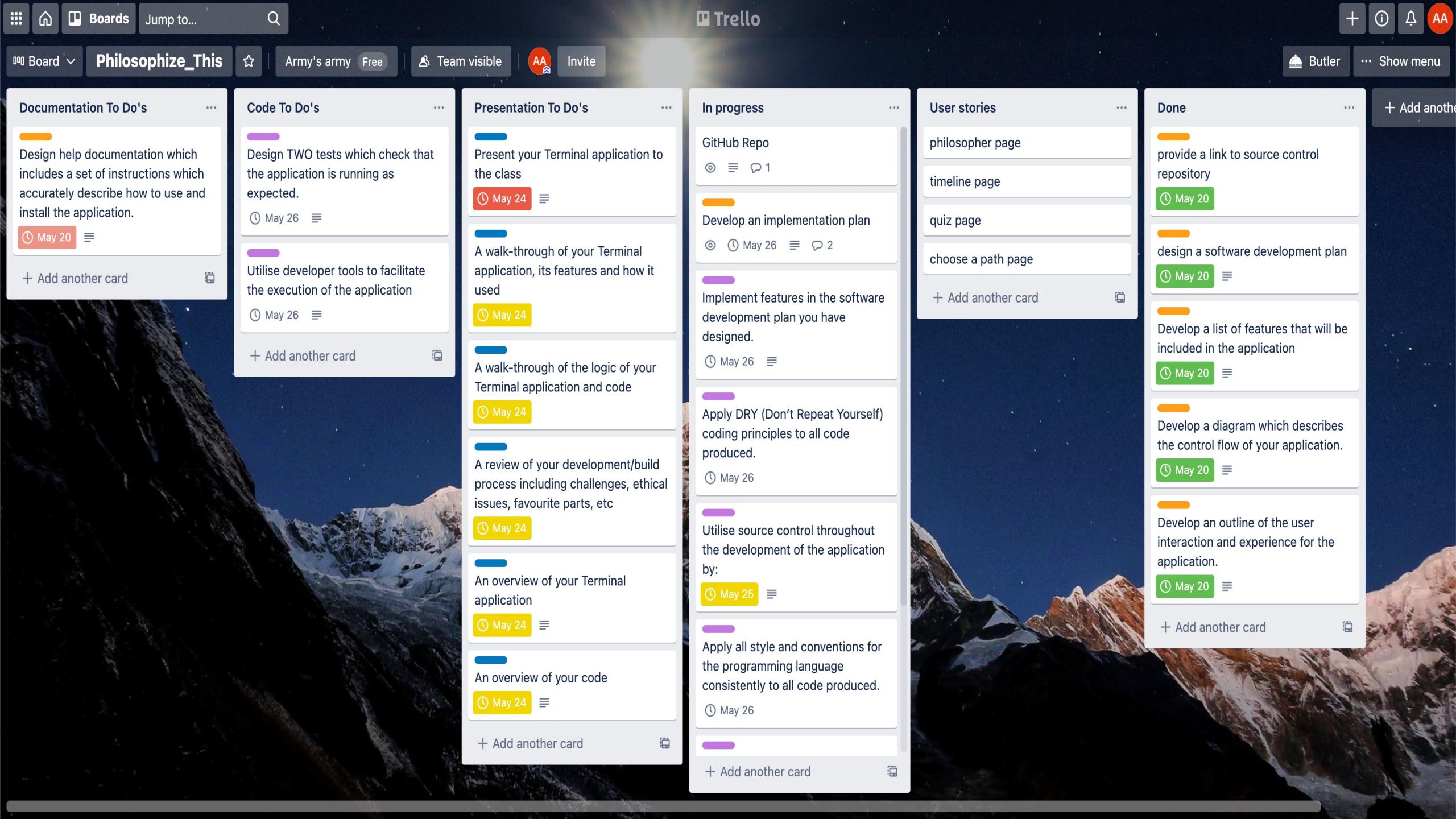
- GitHub Repo (May 20)
- Develop an implementation plan (May 20)

Done

- + Add another card

+ Add another list

<https://trello.com/c/WPVVty6Z/1-flow-chart>

A screenshot of a Trello board titled "Philosophize\_This". The board is organized into six columns: "Documentation To Do's", "Code To Do's", "Presentation To Do's", "In progress", "User stories", and "Done". Each column contains several cards with tasks and due dates. The background of the board is a scenic mountain landscape.

**Documentation To Do's**

- Design help documentation which includes a set of instructions which accurately describe how to use and install the application.  
May 20
- + Add another card

**Code To Do's**

- Design TWO tests which check that the application is running as expected.  
May 26
- Utilise developer tools to facilitate the execution of the application  
May 26
- + Add another card

**Presentation To Do's**

- Present your Terminal application to the class  
May 24
- A walk-through of your Terminal application, its features and how it used  
May 24
- A walk-through of the logic of your Terminal application and code  
May 24
- A review of your development/build process including challenges, ethical issues, favourite parts, etc  
May 24
- An overview of your Terminal application  
May 24
- An overview of your code  
May 24
- + Add another card

**In progress**

- GitHub Repo  
May 26
- Develop an implementation plan  
May 26
- Implement features in the software development plan you have designed.  
May 26
- Apply DRY (Don't Repeat Yourself) coding principles to all code produced.  
May 26
- Utilise source control throughout the development of the application by:  
May 25
- Apply all style and conventions for the programming language consistently to all code produced.  
May 26
- + Add another card

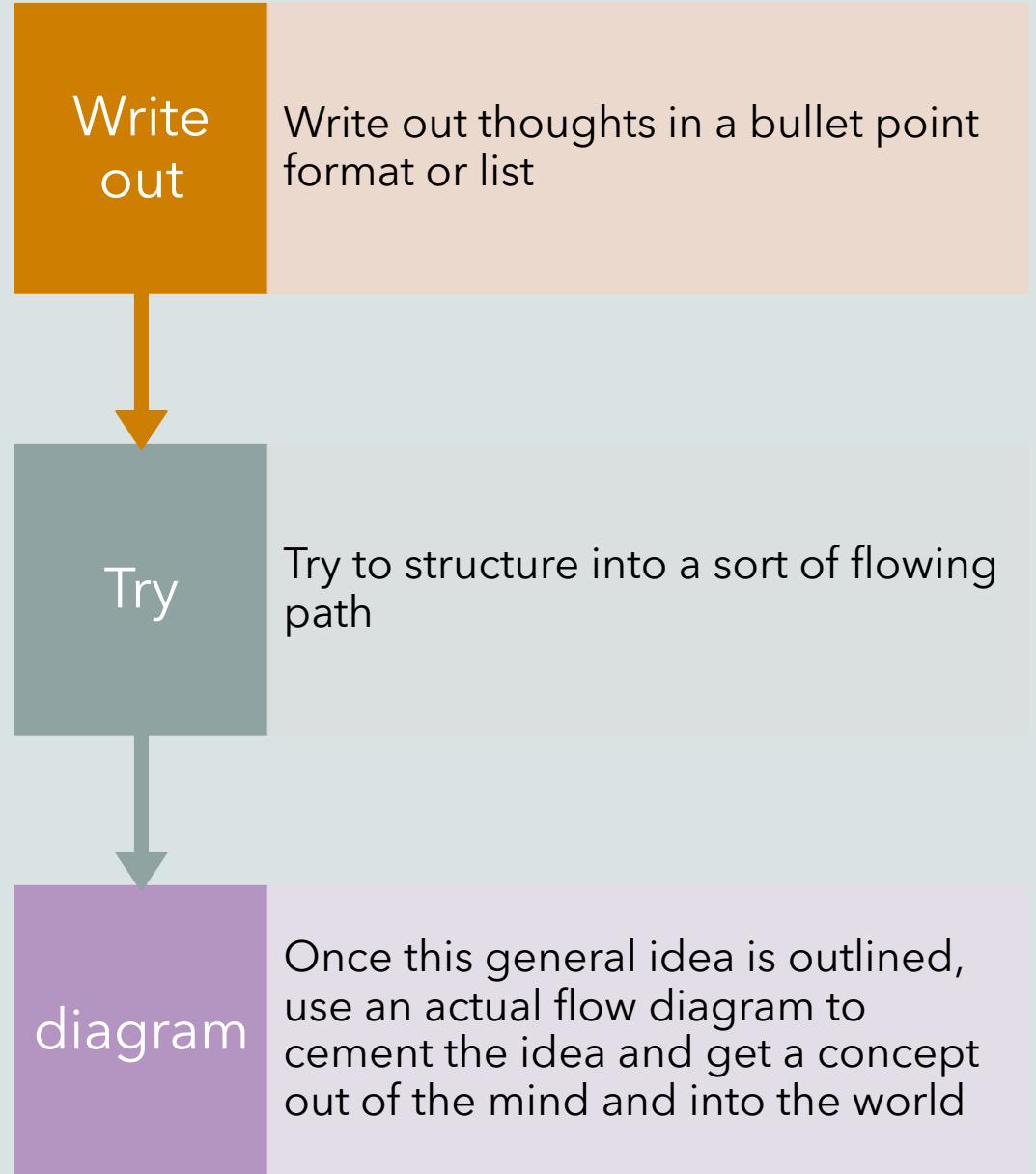
**User stories**

- philosopher page
- timeline page
- quiz page
- choose a path page
- + Add another card

**Done**

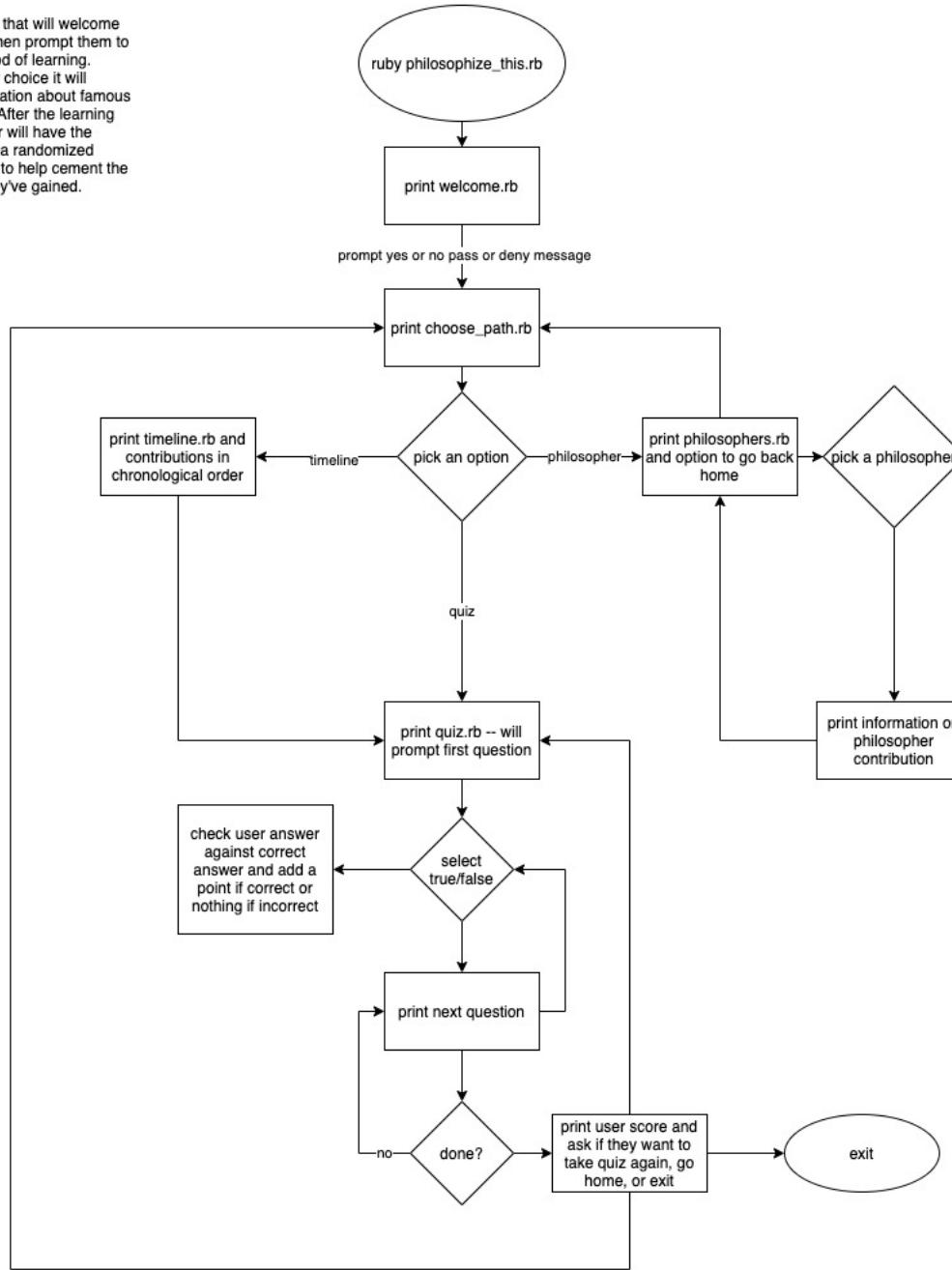
- provide a link to source control repository  
May 20
- design a software development plan  
May 20
- Develop a list of features that will be included in the application  
May 20
- Develop a diagram which describes the control flow of your application.  
May 20
- Develop an outline of the user interaction and experience for the application.  
May 20
- + Add another card

# Next step: Flow diagram



## Philosophize This!

a terminal app that will welcome the user and then prompt them to chose a method of learning. Based off their choice it will present information about famous philosophers. After the learning phase the user will have the choice to take a randomized true/false quiz to help cement the knowledge they've gained.



# Features

- Welcome page
- Take the timeline
- Pick your philosopher
- True / false quiz

src > welcome.rb > ...

```
1  require('colorize')
2  require('tty-prompt')
3  require('artii')
4  require_relative('./choose_path')
5
6  def asciiify_slant(input)
7      a = Artii::Base.new :font => 'slant'
8      a.asciiify(input)
9  end
10
11 puts asciiify_slant("Welcome").magenta.on_blue
12
13 prompt = TTY::Prompt.new
14
15 name = prompt.ask("What is your name?")
16
17 puts "You know #{name}, Descartes once famously said: I think, therefore I am. That leads me to wonder, If I emulat
18
19 yes_no = prompt.yes?("So #{name}, are you a lover of wisdom? ")
20 if yes_no == true
21     puts asciiify_slant("Let us learn!").magenta.on_blue
22     choose
23 else
24     puts "Some have said that knowledge is a burden if it robs you of joy. Come back if you ever decide that yo
25 end
26
```

src > 📁 choose\_path.rb > ruby-symbols > ⚡ choose

```
1  require('tty-prompt')
2  require_relative './quiz'
3  require_relative './philosophers'
4  require_relative './timeline'
5  require 'json'
6
7  file = File.read('./timeline.json')
8
9  $philosophers = JSON.parse(file)
10
11 def choose
12   $prompt = TTY::Prompt.new
13
14   the_choice = $prompt.select("Pick a path or test your knowledge!", %w(Timeline Philosophers Quiz))
15
16   case the_choice
17   when "Timeline"
18     timeline
19   when "Philosophers"
20     choose_philosopher
21     next_choice
22   when "Quiz"
23     take_quiz
24     post_quiz
25   end
26 end
```



```
7
8     file = File.read('./timeline.json')
9
10    $philosopher = JSON.parse(file)["timeline"]
11
12    $prompt = TTY::Prompt.new
13
14    def timeline
15
16        puts "Welcome to the Timeline!"
17        puts "In this option we will go sequentially through each of our great philosophers according to when they show
18
19        @start = $prompt.yes?("Ready to begin?")
20        if @start == true
21            $philosopher.each do |item|
22                p item["Philosopher"]
23                puts ""
24                p item["Time period"]
25                puts ""
26                p item["Known for"]
27                puts ""
28                p item["Famous saying"]
29                puts ""
30                $prompt.keypress("Press space or enter to continue", keys: [:space, :return])
31                puts ""
32            end
33        else
34            return error
35        end
36
37        answer = $prompt.yes?("Now that you're all done, you can either go directly to the quiz or go home") do |q|
38            q.suffix "quiz/home"
39        end
40        if answer == true
41            take_quiz
42        else
43            choose
44        end
45    end
```



```
src > 🐾 philosophers.rb > ...
1 require('tty-prompt')
2 require 'json'
3 require './choose_path'
4
5 file = File.read('./timeline.json')
6
7 $philosophers = JSON.parse[file]
8
9 $prompt = TTY::Prompt.new
10
11 def next_choice
12   go_back = $prompt.yes?("Return to philosopher list?")
13   if go_back == false
14     puts "We can't stay here forever!"
15     choose_philosopher
16   else
17     puts "let's keep learning!"
18     choose_philosopher
19   end
20 end
21
```

```
src > 🐾 philosophers.rb > ...
21
22 #need to populate with information parsed from json, need to include at least two female philosophers
23
24 def choose_philosopher
25   philosopher_choice = $prompt.select("Choose your Philosopher", %w(Aristotle Al-Haytham Aquinas Aurelius Beauvoi
26   case philosopher_choice
27   when "Aristotle"
28     puts "#{philosophers['timeline'][0]['Known for']}"
29     next_choice
30   when "Descartes"
31     puts "#{philosophers['timeline'][1]['Known for']}"
32     next_choice
33   when "Confucius"
34     puts "#{philosophers['timeline'][2]['Known for']}"
35     next_choice
36   when "Al-Haytham"
37     puts "#{philosophers['timeline'][3]['Known for']}"
38     next_choice
39   when "Hobbes"
40     puts "#{philosophers['timeline'][4]['Known for']}"
41     next_choice
42   when "Nietzsche"
43     puts "#{philosophers['timeline'][5]['Known for']}"
44     next_choice
45   when "Beauvoir"
46     puts "#{philosophers['timeline'][6]['Known for']}"
47     next_choice
48   when "Aquinas"
49     puts "#{philosophers['timeline'][7]['Known for']}"
50     next_choice
51   when "Aurelius"
52     puts "#{philosophers['timeline'][8]['Known for']}"
53     next_choice
54   when "Wollstonecraft"
55     puts "#{philosophers['timeline'][9]['Known for']}"
56     next_choice
57   when "Go back"
58     require_relative 'choose_path.rb'
59     choose
60   end
61 end
```

```
src > quiz.rb > ruby-symbols > take_quiz
 1  require('colorize')
 2  require('tty-prompt')
 3  require('artii')
 4  require('json')
 5  require_relative './choose_path'
 6
 7  def asciiify_banner(input)
 8    a = Artii::Base.new :font => 'banner3'
 9    a.asciiify(input)
10  end
11
12  def take_quiz
13
14    $prompt = TTY::Prompt.new
15
16    @user_score = 0
17
18    def asciiify_banner(input)
19      a = Artii::Base.new :font => 'banner3'
20      a.asciiify(input)
21    end
22
23    @question_array = [
24      {question:"Best known for his Meditations on Stoic philosophy. Marcus Aurelius has symbolized for many gene",
25       question:"Jenny has symbolized for many generations in the West the Golden Age of the Roman Empire.", value: t,
26       question:"Mary Walstonecroft was an english writer and passionate advocate of educational and social equal",
27       question:"Thomas Aquinas was an italian Dominican theologian, the foremost medieval Scholastic.", value: t,
28       question:"Confucius was a french writer and feminist, a member of the intellectual fellowship of philosoph",
29       question:"Thomas Hobbes is best known for his Meditations on Stoic philosophy.", value: false},
30       question:"Friedrich Nietzsche was a German classical scholar and philosopher.", value: true},
31       question:"Simone de Beauvoir was an english philosopher, scientist, and historian, best known for his poli",
32       question:"Thomas Hobbes was an english philosopher, scientist, and historian, best known for his political",
33       question:"Friedrich Nietzsche was a feminist and a philosopher.", value: false},
34       question:"Ibn Al-Haytham was a mathematician and astronomer who made significant contributions to the prin",
35       question:"Aristotle was a mathematician and astronomer who made significant contributions to the principle
36    ]

```

```
src > quiz.rb > ruby-symbols > take_quiz
41
42  @question_array.each do |quiz|
43    answer = $prompt.yes?(" #{quiz[:question]}") do |q|
44      q.suffix "true/false"
45    end
46    if answer == quiz[:value]
47      puts ""
48      puts "Correct!"
49      puts ""
50      @user_score += 1
51    else
52      puts ""
53      puts "Incorrect!"
54      puts ""
55    end
56  end
57
58
59  def post_quiz
60    if @user_score > 8
61      congratulations_message = $prompt.yes?("Way to go! You got #{@user_score}/12 correct. Looks like you've ret",
62      q.suffix "home/exit"
63    end
64    if congratulations_message == true
65      choose
66    else
67      exit
68    end
69    #puts
70  elsif @user_score < 8
71    fail_answer = $prompt.yes?("You got #{@user_score}/12 correct. Perhaps you would like to try again?")
72    if fail_answer == true
73      @user_score = 0
74      take_quiz
75      post_quiz
76    else
77      answer_no = $prompt.yes?("Fair enough. Would you like to go home or exit?") do |q|
78        q.suffix "home/exit"
79      end
80      if answer_no == true
81        choose
82      else
83        exit
84      end
85    end
86  end

```

```
> {} timeline.json > ...
1
2   "timeline" : [
3     {
4       "Philosopher": "Aristotle",
5       "Time period": "4th Century (BC)",
6       "url": "https://www.britannica.com/biography/Aristotle",
7       "Known for": "One of the greatest philosophers who ever lived and the first genuine scientist in history. He made pioneering contributions to all fields of philosophy and science, particularly in logic, metaphysics, ethics, politics, biology, and mathematics.",
8       "Famous saying": "He who is to be a good ruler must have first been ruled."
9     },
10    {
11      "Philosopher": "René Descartes",
12      "Time period": "17th Century",
13      "url": "https://plato.stanford.edu/entries/descartes/",
14      "Known for": "He offered a new vision of the natural world that continues to shape our thought today: a world of matter possessing a few fundamental properties (extension, location, and causality).",
15      "Famous saying": "I think, therefore I am."
16    },
17    {
18      "Philosopher": "Confucius",
19      "Time period": "6th – 5th Century (BC)",
20      "url": "https://www.biography.com/scholar/confucius",
21      "Known for": "Confucius, also known as Kong Qiu or K'ung Fu-tzu, was a Chinese philosopher, teacher and political figure. His teachings, preserved in the Analects, have had a profound influence on Chinese culture and society.", "Famous saying": "What you do not wish for yourself, do not do to others."
22    },
23    {
24      "Philosopher": "Ibn Al-Haytham",
25      "Time period": "10th – 11th Century",
26      "url": "https://www.britannica.com/biography/Ibn-al-Haytham",
27      "Known for": "Mathematician and astronomer who made significant contributions to the principles of optics and the use of scientific experiments.", "Famous saying": "The seeker after truth is not one who studies the writings of the ancients and, following his natural disposition, puts his trust in them like a child in its保姆 (nurse). He is rather one who, having carefully examined the question by himself, follows his own path.".
28    },
29    {
30      "Philosopher": "Thomas Hobbes",
31      "Time period": "16th – 17th Century",
32      "url": "https://www.britannica.com/biography/Thomas-Hobbes",
33      "Known for": "English philosopher, scientist, and historian, best known for his political philosophy, especially as articulated in his masterpiece 'Leviathan'.", "Famous saying": "No arts; no letters; no society; and which is worst of all, continual fear and danger of violent death; and the life of man, so poor, nasty, brutish, and short.".
34    },
35    {
36      "Philosopher": "Friedrich Nietzsche",
37      "Time period": "19th Century",
38      "url": "https://www.britannica.com/biography/Friedrich-Nietzsche",
39      "Known for": "German classical scholar, philosopher, and critic of culture, who became one of the most influential of all modern thinkers. His attacks on traditional morality and religious values were controversial.", "Famous saying": "God is dead: but considering the state Man is in, there will perhaps be caves, for ages yet, in which his shadow will be shown".
40    },
41    {
42      "Philosopher": "Simone de Beauvoir",
43      "Time period": "20th Century",
44      "url": "https://www.britannica.com/biography/Simone-de-Beauvoir",
45      "Known for": "French writer and feminist, a member of the intellectual fellowship of philosopher-writers who have given a literary transcription of their thoughts.", "Famous saying": "Man is defined as a human being and woman as a female – whenever she behaves as a human being she is said to imitate the male.".
46    },
47    {
48      "Philosopher": "St. Thomas Aquinas",
49      "Time period": "13th Century",
50      "url": "https://www.britannica.com/biography/Saint-Thomas-Aquinas",
51      "Known for": "Italian Dominican theologian, the foremost medieval Scholastic. He developed his own conclusions from Aristotelian premises, notably in his 'Summa Contra Gentiles' and 'Summa Theologica'.", "Famous saying": "To one who has faith, no explanation is necessary. To one without faith, no explanation is possible.".
52    }
53  ].
```

# Issues, challenges, and favorite parts

- Did somebody say time?
- Ambition Vs. Realistic expectations: Round 2
- Wrapping my head around concepts that weren't exactly covered in class
- Working with classmates in the trenches = far and away best part of this
- Second up for best part would be working through actual problems on my own for the first time.



# Thank you for listening!

---

- Let's all keep on going and become awesome coders so this can be the type of weekend getaway we look forward to!!