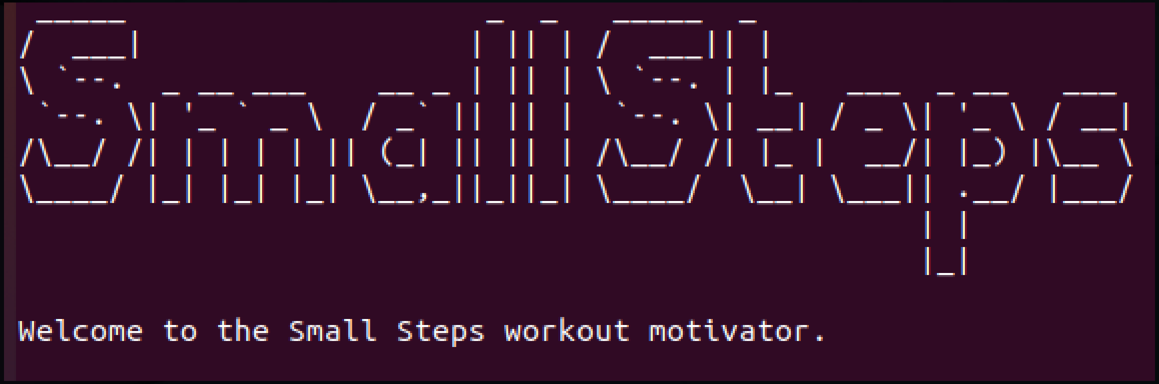


```
 _____                 _  _  _____  _
/  ___|               | || |/  ___|| |
\ `--.  _ __ ___    __ _ | || |\ `--. | |_   ___  _ __   ___
 `--. \| '_ ` _ \  / _` || || | `--. \| __| / _ \| '_ \ / __|
/\__/ /| | | | | || (_| || || |/\__/ /| |_ |  __/| |_) |\__ \
\____/ |_| |_| |_| \__,_||_||_|\____/  \__| \___|| .__/ |___/
                                                  | |
                                                  |_|

Welcome to the Small Steps workout motivator.
```

Small Steps – *your exercise habit helper*

Why this app?

- There are a lot of people (like myself) who want to start exercising
- It can be hard to just commit to doing even a few exercises a day

Why this app?

- Small Steps allows a user to:

  - choose a few exercises,

  - set their target repetitions for each exercise,

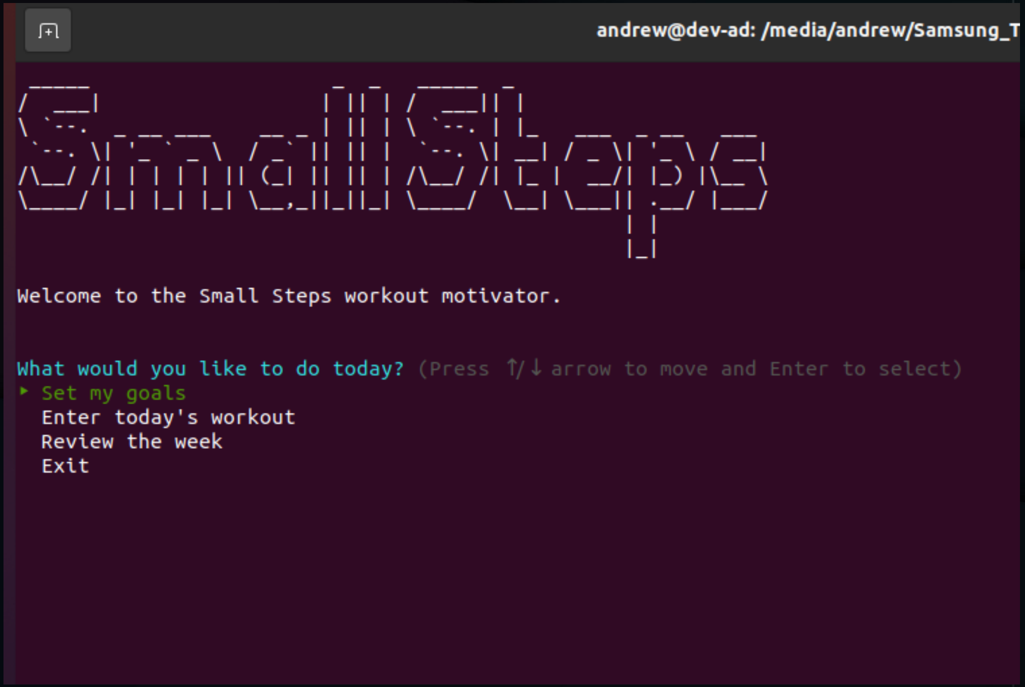
  - track their workouts over the week

  - check their results VS their goals

The app needed to:

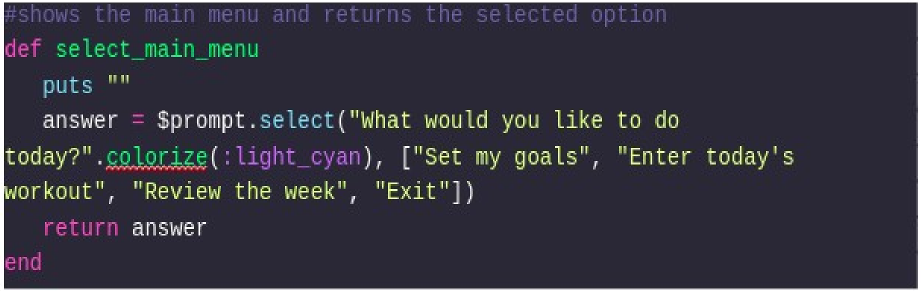
- Allow the user to choose exercises and target repetitions
- Allow the user to enter data for days they worked out
- Remember the user’s data (write to and read from files)
- Display the results for the week in a table

Features: main menu

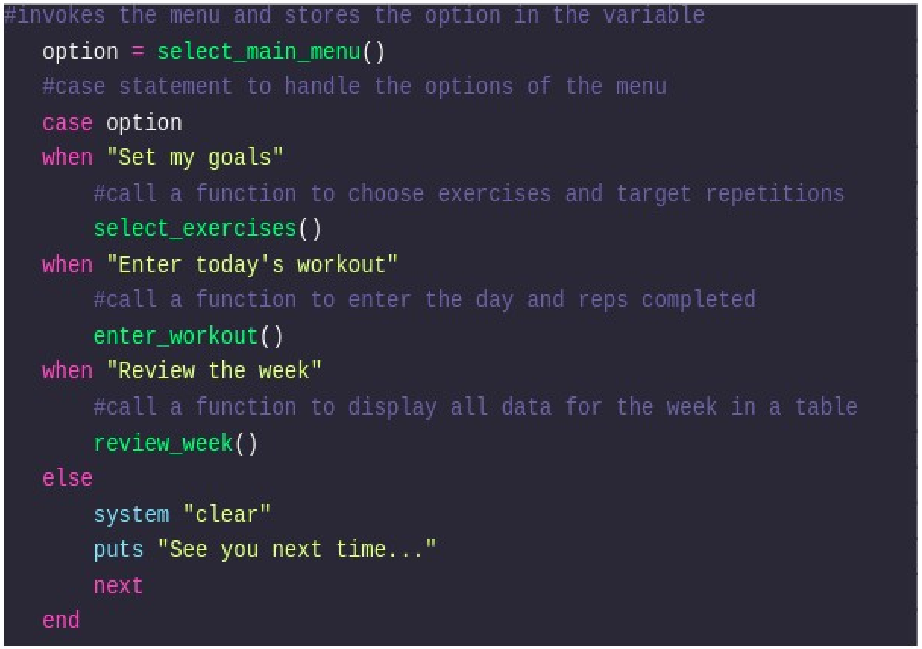


andrew@dev-ad: /media/andrew/Samsung_T

```
 _____                 _ _   _____ _                 
/  ___|               | | | /  ___| |                
\ `--. _ __ ___   __ _| | | \ `--.| |_ ___ _ __  ___ 
 `--. \ '_ ` _ \ / _` | | |  `--. \ __/ _ \ '_ \/ __|
/\__/ / | | | | | (_| | | | /\__/ / ||  __/ |_) \__ \
\____/|_| |_| |_|\__,_|_|_| \____/ \__\___| .__/|___/
                                          | |        
                                          |_|        

Welcome to the Small Steps workout motivator.


What would you like to do today? (Press ↑/↓ arrow to move and Enter to select)
‣ Set my goals
  Enter today's workout
  Review the week
  Exit
```

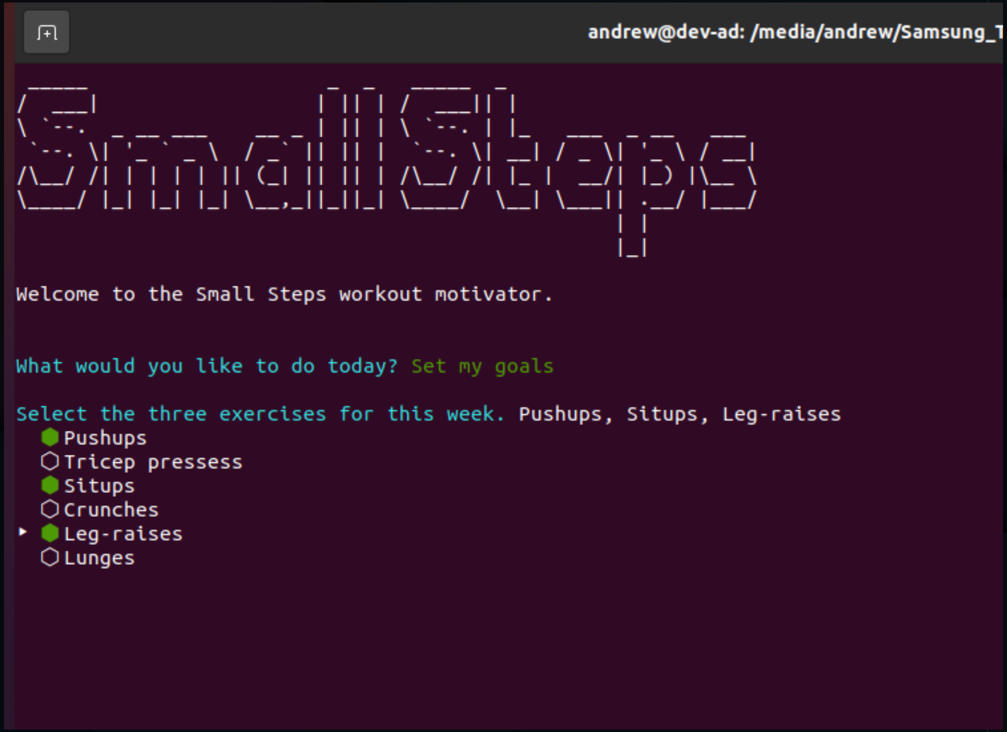


```
#shows the main menu and returns the selected option
def select_main_menu
   puts ""
   answer = $prompt.select("What would you like to do today?".colorize(:light_cyan), ["Set my goals", "Enter today's workout", "Review the week", "Exit"])
   return answer
end
```

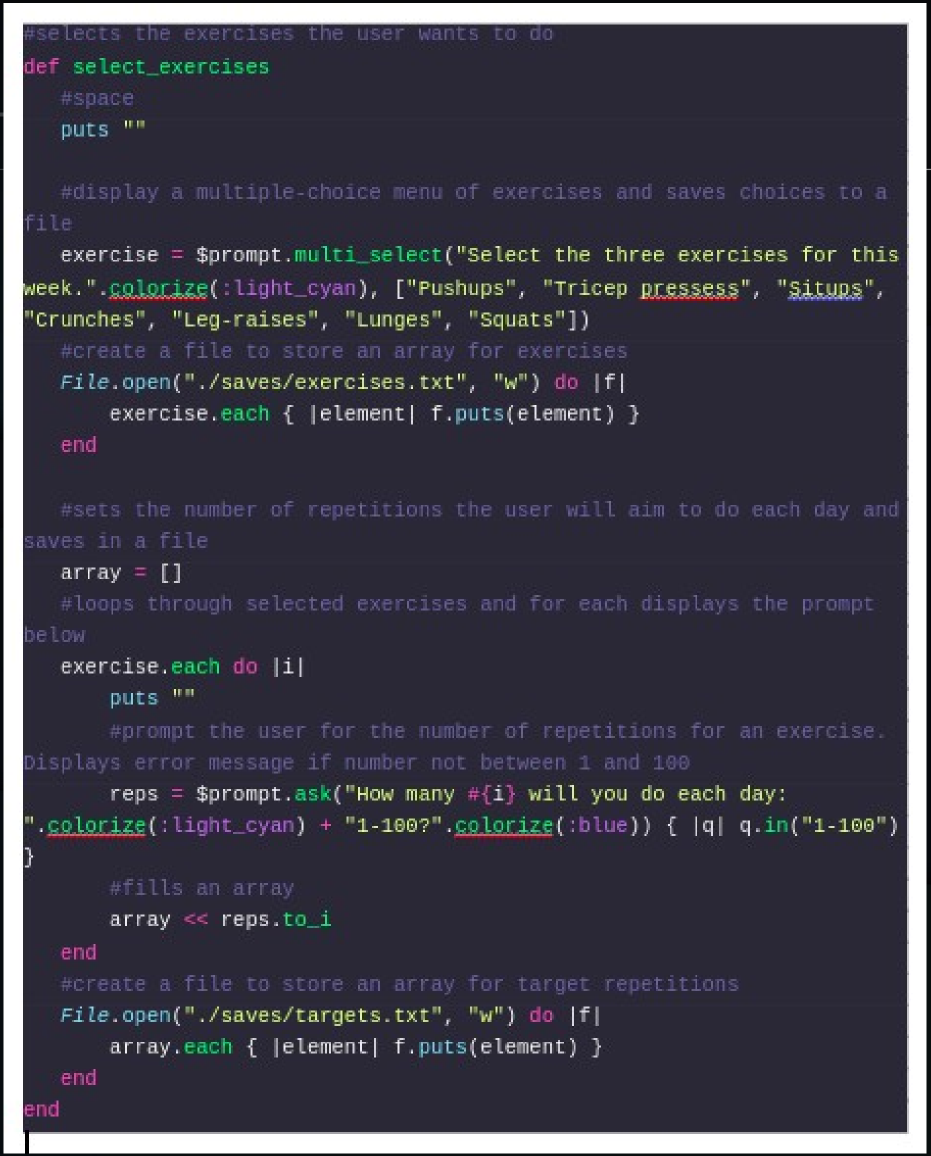


```
#invokes the menu and stores the option in the variable
   option = select_main_menu()
   #case statement to handle the options of the menu
   case option
   when "Set my goals"
       #call a function to choose exercises and target repetitions
       select_exercises()
   when "Enter today's workout"
       #call a function to enter the day and reps completed
       enter_workout()
   when "Review the week"
       #call a function to display all data for the week in a table
       review_week()
   else
       system "clear"
       puts "See you next time..."
       next
   end
```

Feature: input your goals

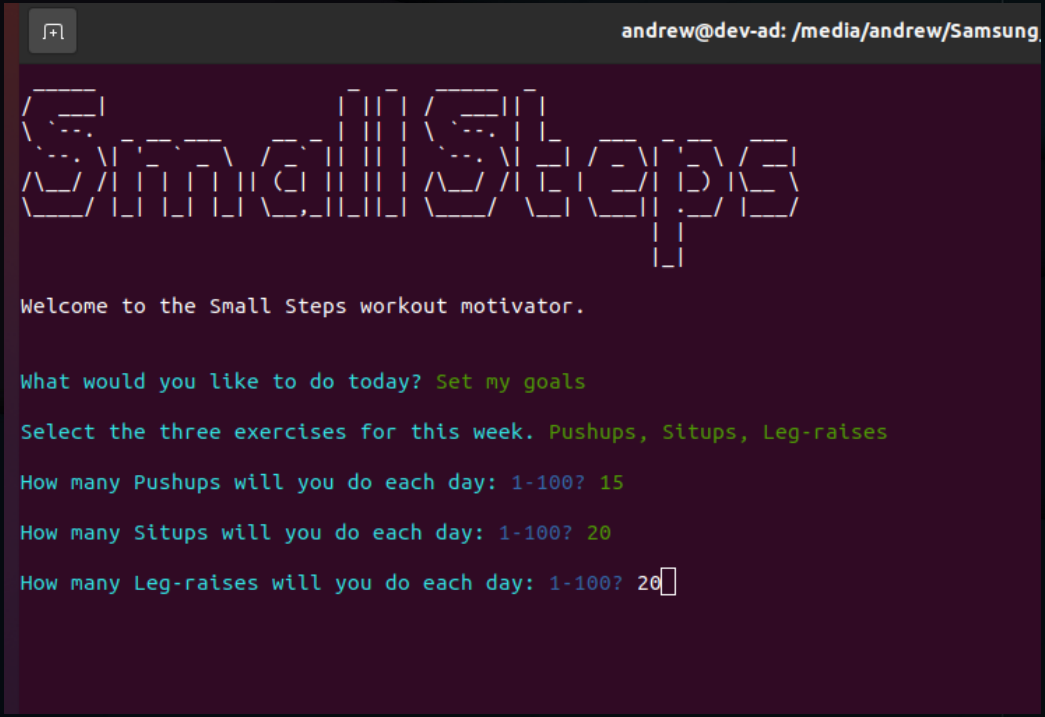


```
 _____                    _  _  _____  _
/  ___|                  | || |/  ___|| |
\ `--.  _ __ ___    __ _ | || |\ `--. | |_   ___  _ __   ___
 `--. \| '_ ` _ \  / _` || || | `--. \| __| / _ \| '_ \ / __|
/\__/ /| | | | | || (_| || || |/\__/ /| |_ |  __/| |_) |\__ \
\____/ |_| |_| |_| \__,_||_||_|\____/  \__| \___|| .__/ |___/
                                                 | |
                                                 |_|

Welcome to the Small Steps workout motivator.


What would you like to do today? Set my goals

Select the three exercises for this week. Pushups, Situps, Leg-raises
  ⬢ Pushups
  ⬡ Tricep pressess
  ⬢ Situps
  ⬡ Crunches
‣ ⬢ Leg-raises
  ⬡ Lunges
```

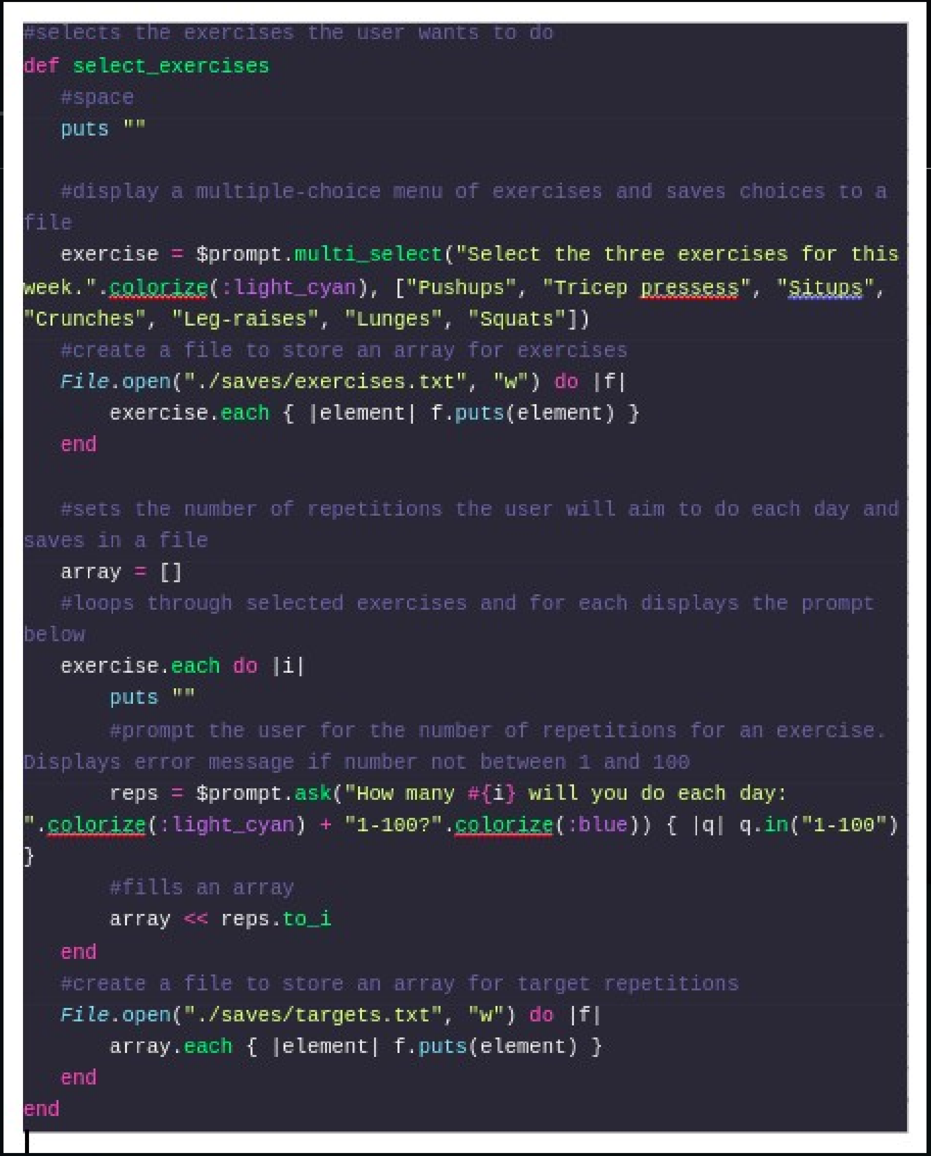


```
#selects the exercises the user wants to do
def select_exercises
    #space
    puts ""

    #display a multiple-choice menu of exercises and saves choices to a file
    exercise = $prompt.multi_select("Select the three exercises for this week.".colorize(:light_cyan), ["Pushups", "Tricep pressess", "Situps", "Crunches", "Leg-raises", "Lunges", "Squats"])
    #create a file to store an array for exercises
    File.open("./saves/exercises.txt", "w") do |f|
        exercise.each { |element| f.puts(element) }
    end

    #sets the number of repetitions the user will aim to do each day and saves in a file
    array = []
    #loops through selected exercises and for each displays the prompt below
    exercise.each do |i|
        puts ""
        #prompt the user for the number of repetitions for an exercise. Displays error message if number not between 1 and 100
        reps = $prompt.ask("How many #{i} will you do each day: ".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100") }
        #fills an array
        array << reps.to_i
    end
    #create a file to store an array for target repetitions
    File.open("./saves/targets.txt", "w") do |f|
        array.each { |element| f.puts(element) }
    end
end
```

Feature: input your goals

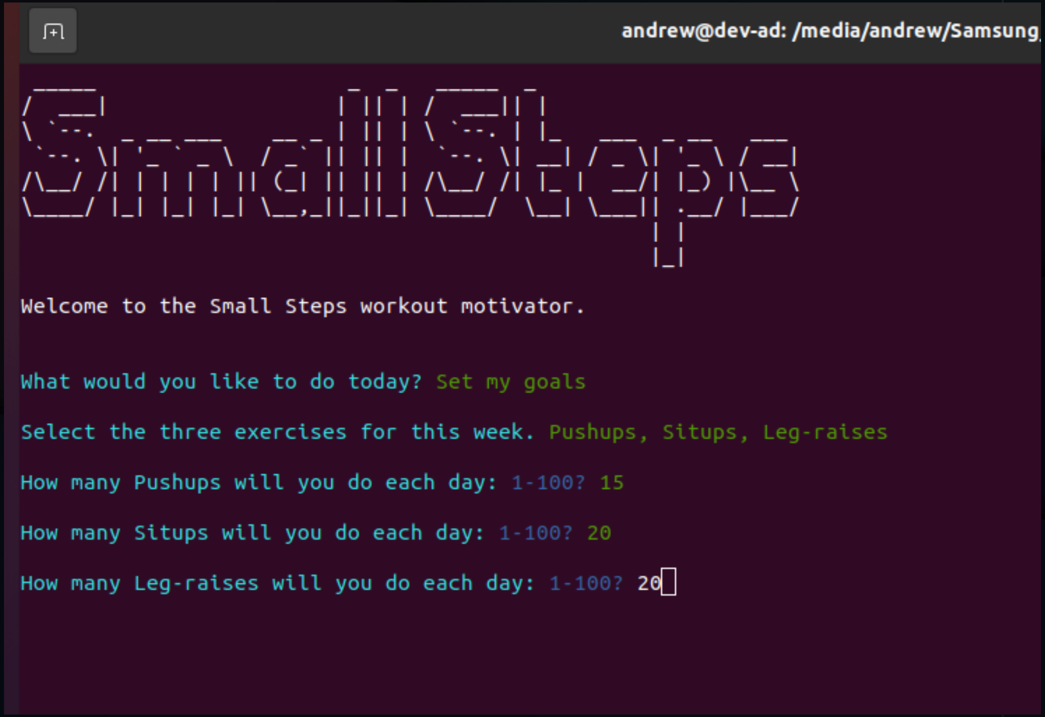


```
 _____                   _  _  _____  _
/  ___|                  | || |/  ___|| |
\ `--.  _ __ ___    __ _ | || |\ `--. | |_   ___  _ __   ___
 `--. \| '_ ` _ \  / _` || || | `--. \| __| / _ \| '_ \ / __|
/\__/ /| | | | | || (_| || || |/\__/ /| |_ |  __/| |_) |\__ \
\____/ |_| |_| |_| \__,_||_||_|\____/  \__| \___|| .__/ |___/
                                                 | |
                                                 |_|

Welcome to the Small Steps workout motivator.


What would you like to do today? Set my goals

Select the three exercises for this week. Pushups, Situps, Leg-raises

How many Pushups will you do each day: 1-100? 15

How many Situps will you do each day: 1-100? 20

How many Leg-raises will you do each day: 1-100? 20
```

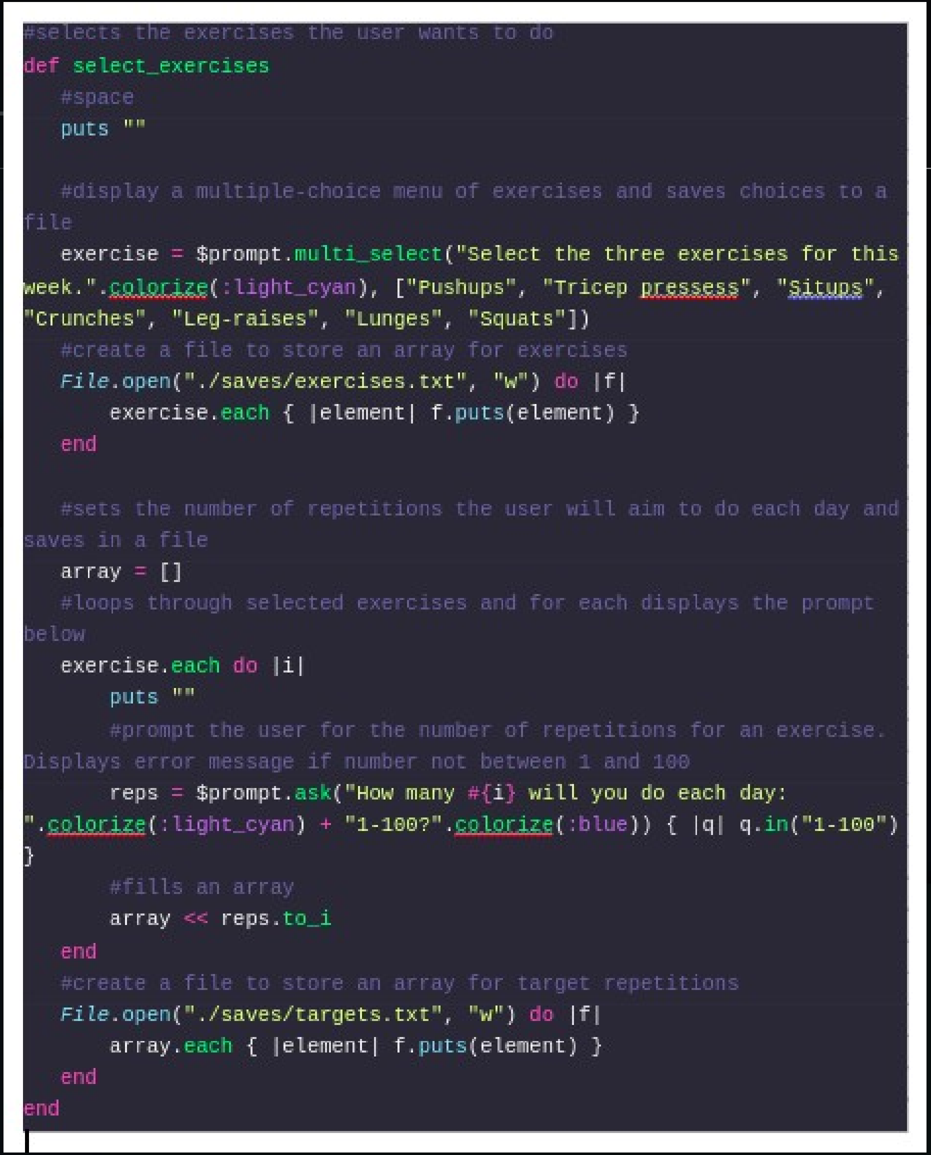


```
#selects the exercises the user wants to do
def select_exercises
   #space
   puts ""

   #display a multiple-choice menu of exercises and saves choices to a
file
   exercise = $prompt.multi_select("Select the three exercises for this
week.".colorize(:light_cyan), ["Pushups", "Tricep pressess", "Situps",
"Crunches", "Leg-raises", "Lunges", "Squats"])
   #create a file to store an array for exercises
   File.open("./saves/exercises.txt", "w") do |f|
       exercise.each { |element| f.puts(element) }
   end

   #sets the number of repetitions the user will aim to do each day and
saves in a file
   array = []
   #loops through selected exercises and for each displays the prompt
below
   exercise.each do |i|
       puts ""
       #prompt the user for the number of repetitions for an exercise.
Displays error message if number not between 1 and 100
       reps = $prompt.ask("How many #{i} will you do each day:
".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100")
}
       #fills an array
       array << reps.to_i
   end
   #create a file to store an array for target repetitions
   File.open("./saves/targets.txt", "w") do |f|
       array.each { |element| f.puts(element) }
   end
end
```

Feature: input your goals

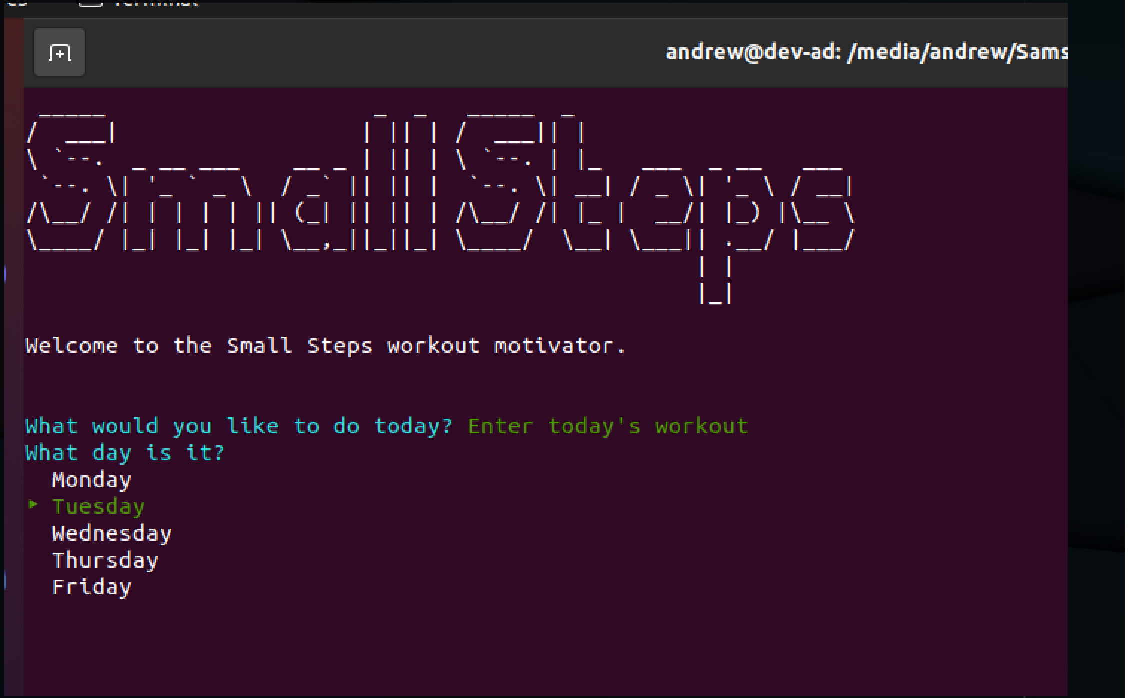


```
 _____                 _ _   _____ _                 
/  ___|               | | | /  ___| |                
\ `--. _ __ ___   __ _| | | \ `--.| |_ ___ _ __  ___ 
 `--. \ '_ ` _ \ / _` | | |  `--. \ __/ _ \ '_ \/ __|
/\__/ / | | | | | (_| | | | /\__/ / ||  __/ |_) \__ \
\____/|_| |_| |_|\__,_|_|_| \____/ \__\___| .__/|___/
                                          | |        
                                          |_|        

Welcome to the Small Steps workout motivator.

What would you like to do today? Set my goals

Select the three exercises for this week. Pushups, Situps, Leg-raises

How many Pushups will you do each day: 1-100? 15

How many Situps will you do each day: 1-100? 20

How many Leg-raises will you do each day: 1-100? 20
```



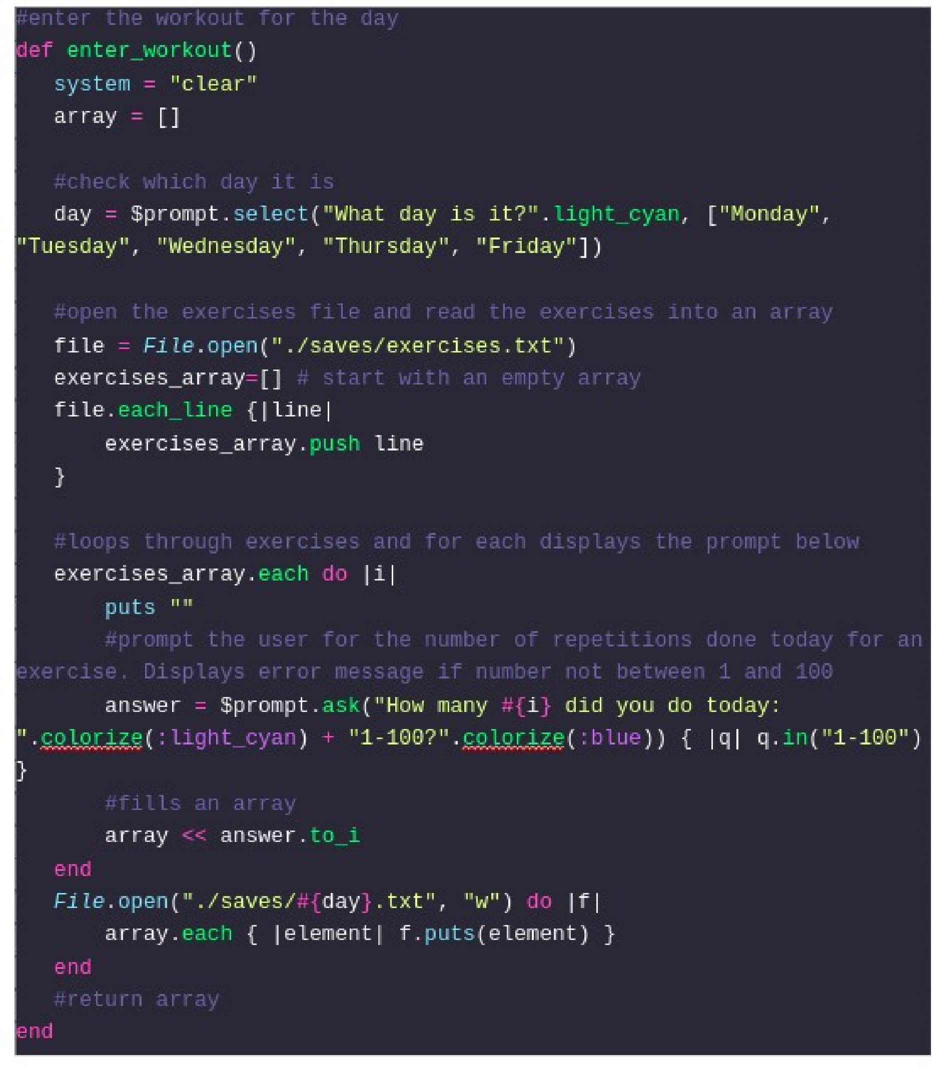
```ruby
#selects the exercises the user wants to do
def select_exercises
   #space
   puts ""

   #display a multiple-choice menu of exercises and saves choices to a file
   exercise = $prompt.multi_select("Select the three exercises for this week.".colorize(:light_cyan), ["Pushups", "Tricep pressess", "Situps", "Crunches", "Leg-raises", "Lunges", "Squats"])
   #create a file to store an array for exercises
   File.open("./saves/exercises.txt", "w") do |f|
       exercise.each { |element| f.puts(element) }
   end

   #sets the number of repetitions the user will aim to do each day and saves in a file
   array = []
   #loops through selected exercises and for each displays the prompt below
   exercise.each do |i|
       puts ""
       #prompt the user for the number of repetitions for an exercise. Displays error message if number not between 1 and 100
       reps = $prompt.ask("How many #{i} will you do each day: ".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100") }
       #fills an array
       array << reps.to_i
   end
   #create a file to store an array for target repetitions
   File.open("./saves/targets.txt", "w") do |f|
       array.each { |element| f.puts(element) }
   end
end
```

Feature: input a workout



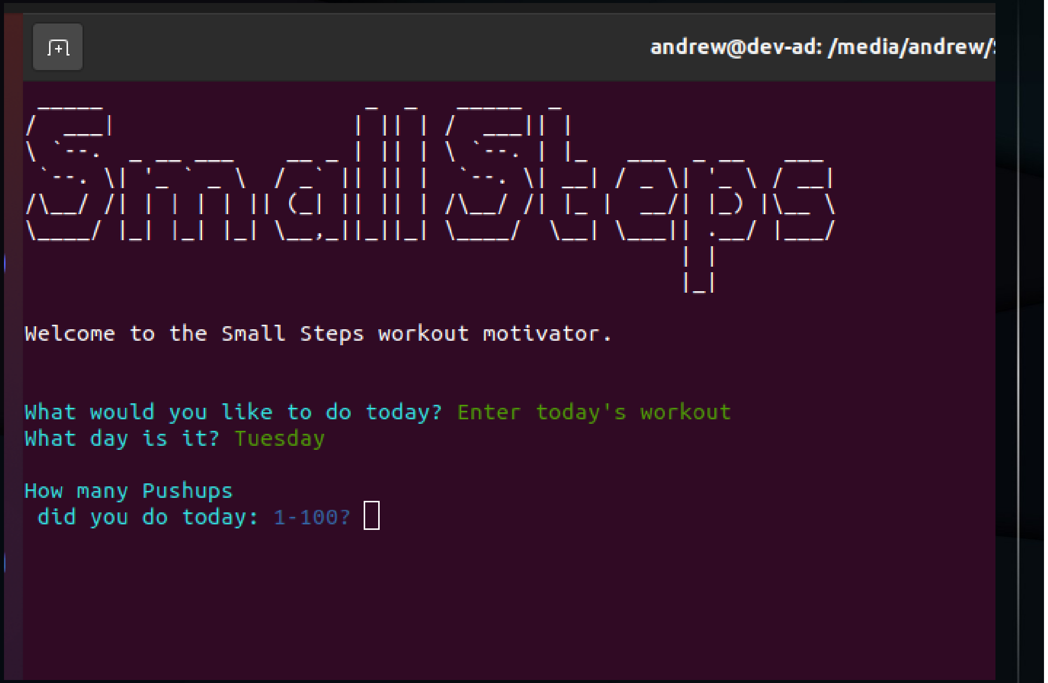
```
andrew@dev-ad: /media/andrew/Sams
```

```
 _____                 _ _   _____ _                 
/  ___|               | | | /  ___| |                
\ `--. _ __ ___   __ _| | | \ `--.| |_ ___ _ __  ___ 
 `--. \ '_ ` _ \ / _` | | |  `--. \ __/ _ \ '_ \/ __|
/\__/ / | | | | | (_| | | | /\__/ / ||  __/ |_) \__ \
\____/|_| |_| |_|\__,_|_|_| \____/ \__\___| .__/|___/
                                          | |        
                                          |_|        

Welcome to the Small Steps workout motivator.


What would you like to do today? Enter today's workout
What day is it?
  Monday
‣ Tuesday
  Wednesday
  Thursday
  Friday
```



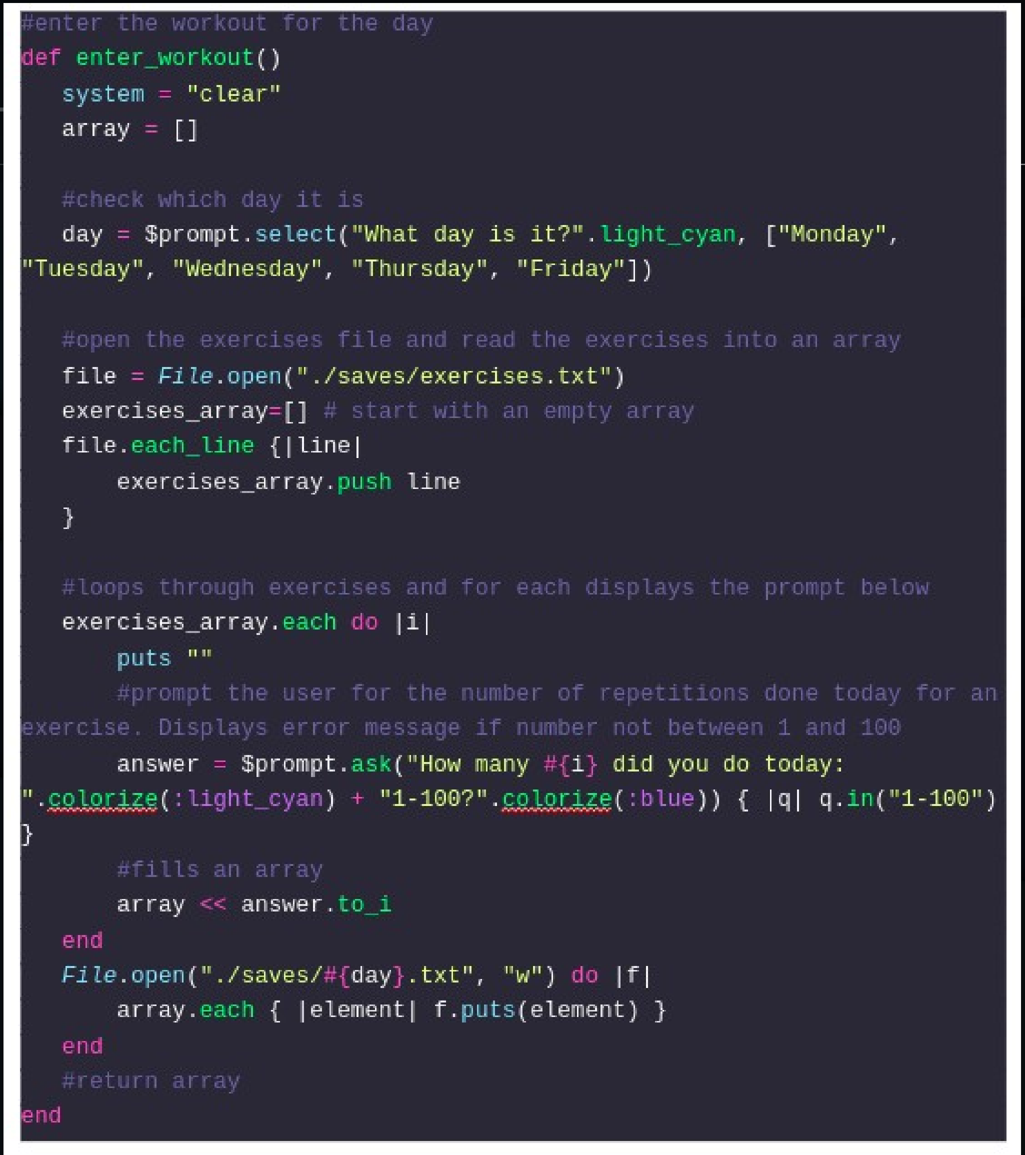
```ruby
#enter the workout for the day
def enter_workout()
    system = "clear"
    array = []

    #check which day it is
    day = $prompt.select("What day is it?".light_cyan, ["Monday",
"Tuesday", "Wednesday", "Thursday", "Friday"])

    #open the exercises file and read the exercises into an array
    file = File.open("./saves/exercises.txt")
    exercises_array=[] # start with an empty array
    file.each_line {|line|
        exercises_array.push line
    }

    #loops through exercises and for each displays the prompt below
    exercises_array.each do |i|
        puts ""
        #prompt the user for the number of repetitions done today for an
exercise. Displays error message if number not between 1 and 100
        answer = $prompt.ask("How many #{i} did you do today:
".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100")
}
        #fills an array
        array << answer.to_i
    end
    File.open("./saves/#{day}.txt", "w") do |f|
        array.each { |element| f.puts(element) }
    end
    #return array
end
```

Feature: input a workout



andrew@dev-ad: /media/andrew/

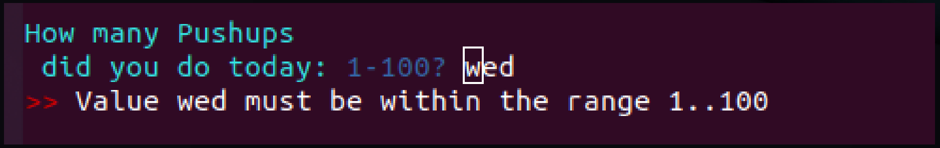
```
 _____                    _  _  _____  _
/  ___|                  | || |/  ___|| |
\ `--.  _ __ ___    __ _ | || |\ `--. | |_   ___  _ __   ___
 `--. \| '_ ` _ \  / _` || || | `--. \| __| / _ \| '_ \ / __|
/\__/ /| | | | | || (_| || || |/\__/ /| |_ |  __/| |_) |\__ \
\____/ |_| |_| |_| \__,_||_||_|\____/  \__| \___|| .__/ |___/
                                                 | |
                                                 |_|

Welcome to the Small Steps workout motivator.


What would you like to do today? Enter today's workout
What day is it? Tuesday

How many Pushups
 did you do today: 1-100?
```



```
#enter the workout for the day
def enter_workout()
   system = "clear"
   array = []

   #check which day it is
   day = $prompt.select("What day is it?".light_cyan, ["Monday",
"Tuesday", "Wednesday", "Thursday", "Friday"])

   #open the exercises file and read the exercises into an array
   file = File.open("./saves/exercises.txt")
   exercises_array=[] # start with an empty array
   file.each_line {|line|
       exercises_array.push line
   }

   #loops through exercises and for each displays the prompt below
   exercises_array.each do |i|
       puts ""
       #prompt the user for the number of repetitions done today for an
exercise. Displays error message if number not between 1 and 100
       answer = $prompt.ask("How many #{i} did you do today:
".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100")
}
       #fills an array
       array << answer.to_i
   end
   File.open("./saves/#{day}.txt", "w") do |f|
       array.each { |element| f.puts(element) }
   end
   #return array
end
```

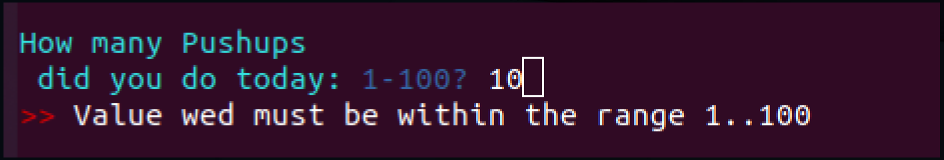
Handling an input error

If the user doesn’t enter an integer in range:

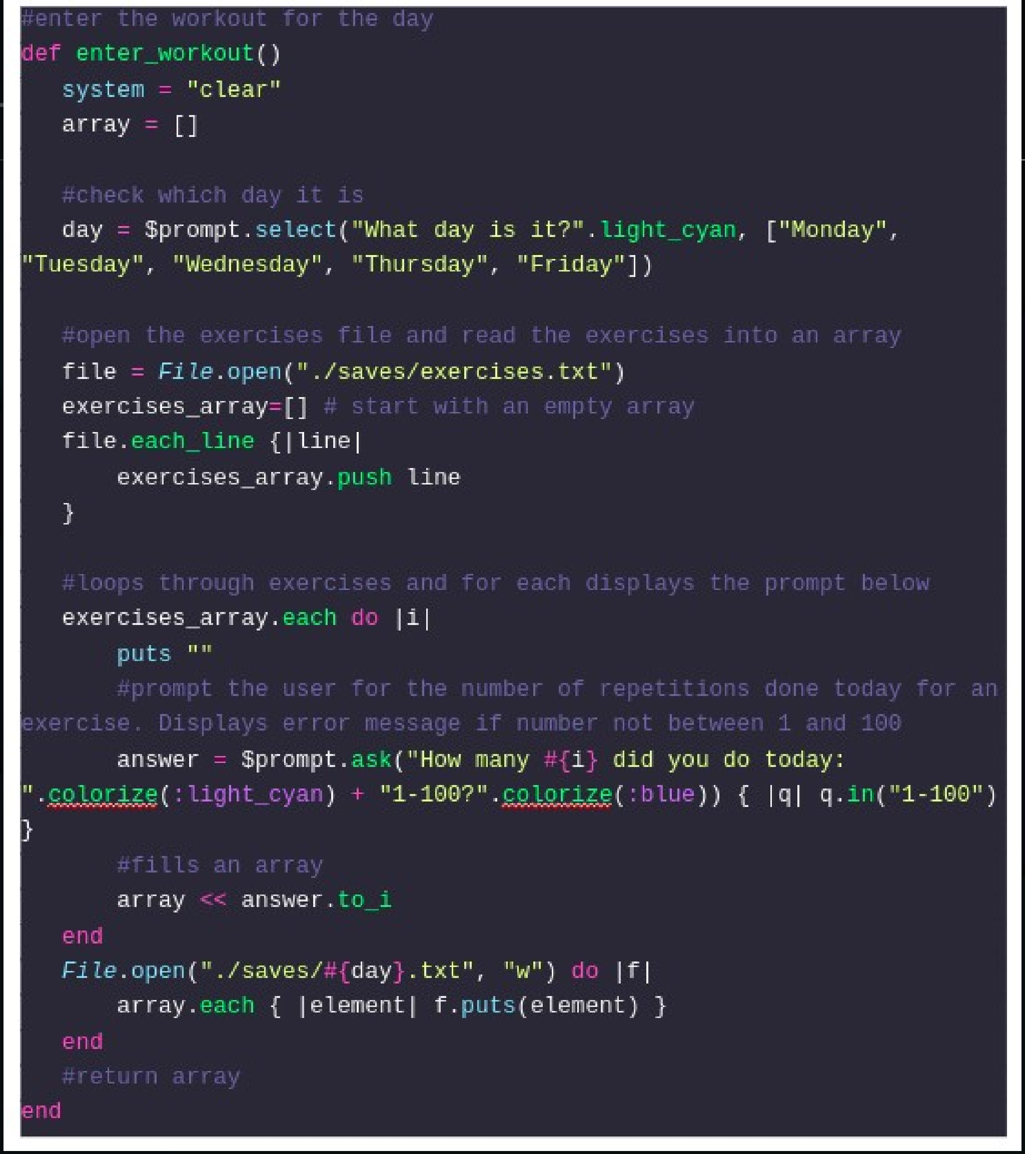


```
How many Pushups
 did you do today: 1-100? wed
>> Value wed must be within the range 1..100
```

Fix: prompt asks user to enter a number between 1 and 100 until they do.

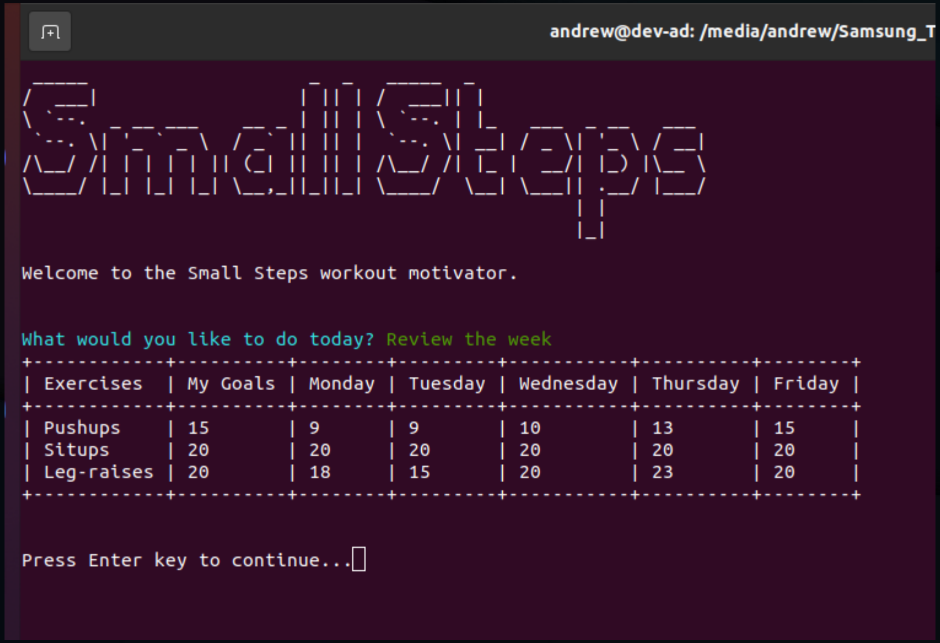


```
How many Pushups
 did you do today: 1-100? 10
>> Value wed must be within the range 1..100
```



```
#enter the workout for the day
def enter_workout()
   system = "clear"
   array = []

   #check which day it is
   day = $prompt.select("What day is it?".light_cyan, ["Monday",
"Tuesday", "Wednesday", "Thursday", "Friday"])

   #open the exercises file and read the exercises into an array
   file = File.open("./saves/exercises.txt")
   exercises_array=[] # start with an empty array
   file.each_line {|line|
       exercises_array.push line
   }

   #loops through exercises and for each displays the prompt below
   exercises_array.each do |i|
       puts ""
       #prompt the user for the number of repetitions done today for an
exercise. Displays error message if number not between 1 and 100
       answer = $prompt.ask("How many #{i} did you do today:
".colorize(:light_cyan) + "1-100?".colorize(:blue)) { |q| q.in("1-100")
}
       #fills an array
       array << answer.to_i
   end
   File.open("./saves/#{day}.txt", "w") do |f|
       array.each { |element| f.puts(element) }
   end
   #return array
end
```

Feature: display results



```
 _____                   _  _  _____ _                           
/  ___|                 | || |/  ___| |                          
\ `--.  _ __ ___   __ _ | || |\ `--.| |_   ___  _ __   ___ 
 `--. \| '_ ` _ \ / _` || || | `--. \| __| / _ \| '_ \ / __|
/\__/ /| | | | | || (_| || || |/\__/ /| |_ |  __/| |_) |\__ \
\____/ |_| |_| |_| \__,_||_||_|\____/  \__| \___|| .__/ |___/
                                                 | |         
                                                 |_|         

Welcome to the Small Steps workout motivator.


What would you like to do today? Review the week
+------------+-----------+--------+---------+------------+-----------+--------+
| Exercises  | My Goals  | Monday | Tuesday | Wednesday  | Thursday  | Friday |
+------------+-----------+--------+---------+------------+-----------+--------+
| Pushups    | 15        | 9      | 9       | 10         | 13        | 15     |
| Situps     | 20        | 20     | 20      | 20         | 20        | 20     |
| Leg-raises | 20        | 18     | 15      | 20         | 23        | 20     |
+------------+-----------+--------+---------+------------+-----------+--------+


Press Enter key to continue...
```

Work in Progress

- Needs to read data from all saved files (goals and each workout day)
- Put the data into hashes inside an array
- Can then be output into the table shown

Challenges

- How to input and store data: arrays, hashes, or classes?

  - arrays were my first go-to

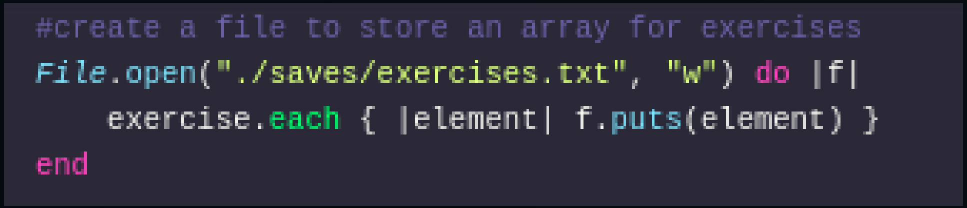
  - hashes seem a little more logical (and would have helped with the final table!)

  - *rspec testing* seems easier with classes (and they can store the data logically)

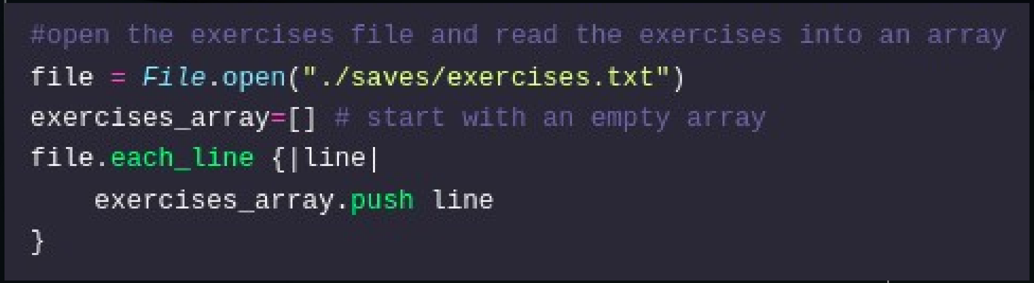
Challenges

- Data persistence: *needing to read and write to files*

  - It makes sense that the app can be closed and keep the data input each time



```ruby
#create a file to store an array for exercises
File.open("./saves/exercises.txt", "w") do |f|
    exercise.each { |element| f.puts(element) }
end
```



```ruby
#open the exercises file and read the exercises into an array
file = File.open("./saves/exercises.txt")
exercises_array=[] # start with an empty array
file.each_line {|line|
    exercises_array.push line
}
```

Ethical Issues

- None for the application purpose
- Use of code searched online:

  - when I use code from Stackexchange, etc. I acknowledge its source if it is significant

  - 1 example of this in my code for the table

Favourite Parts

- Creating my flow diagrams (some of them)
- Solving the file I/O
- Reducing excess code down
- moving most operations into the functions which reduced redundancy (and all of the arrays I started with!)
- *Seeing things work!*

Demonstration

Allow me to demonstrate ***Small Steps*** *in action*

The End

Any questions?