DIET-IT

The diet where you get to choose!

What's the issue?

Western Society Problem

The app identify a problem in the awareness of the calorie intake for every meal.

It is usual to hear people claiming not eating so much but still having weight issues.

HOW DO WE DEVELOP AWARENESS ON THE SINGLE MEAL CALORIES?

Most of us would like to know in real time the caloric intake of our meal

Despite most of the information being available, it is still very hard to keep track of the daily caloric intake.

This app aims to give you an easy tool to give you control and choice on the food to eat and how many calories you want to take.

First of all: freedom of choice!

- We should be free to choose the calorie intake for our meal
- We should be free to choose the specific food
- We should be made aware of the caloric index for our meal
- No judgmental comments or imposed food regime.

The Choice is yours!

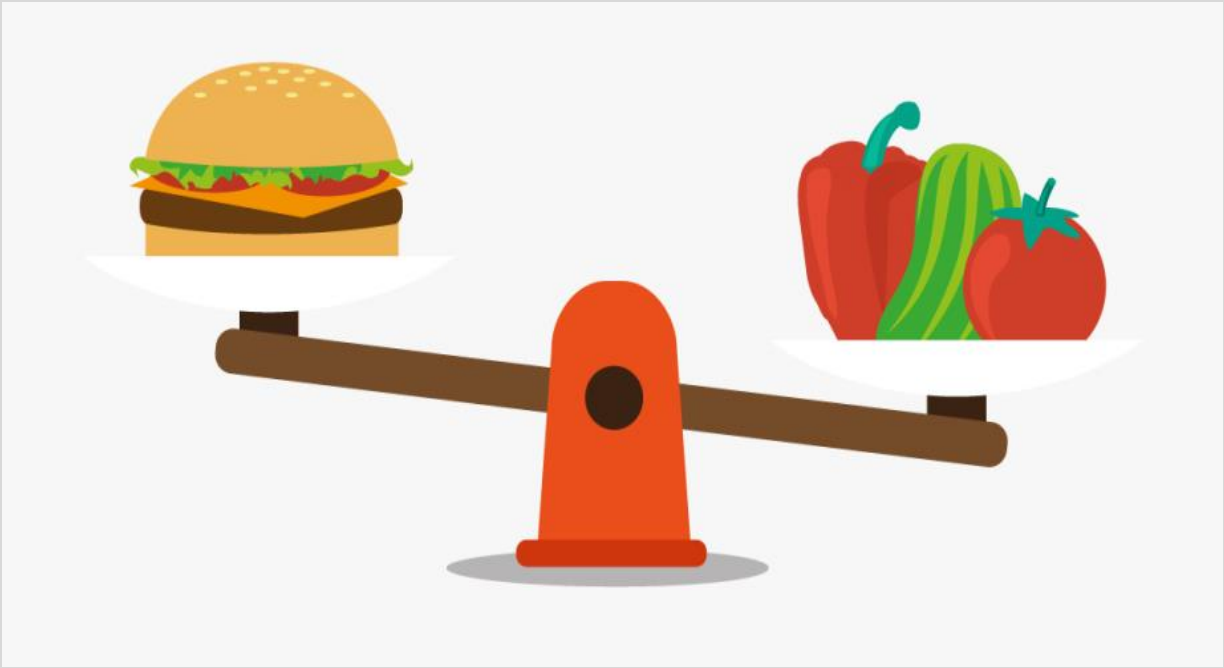
How can we help?

1. Tell me how many calories you want to take today

2. Tell what do you feel like eating

3. These are the calories you will intake with the food you chosen

The output:



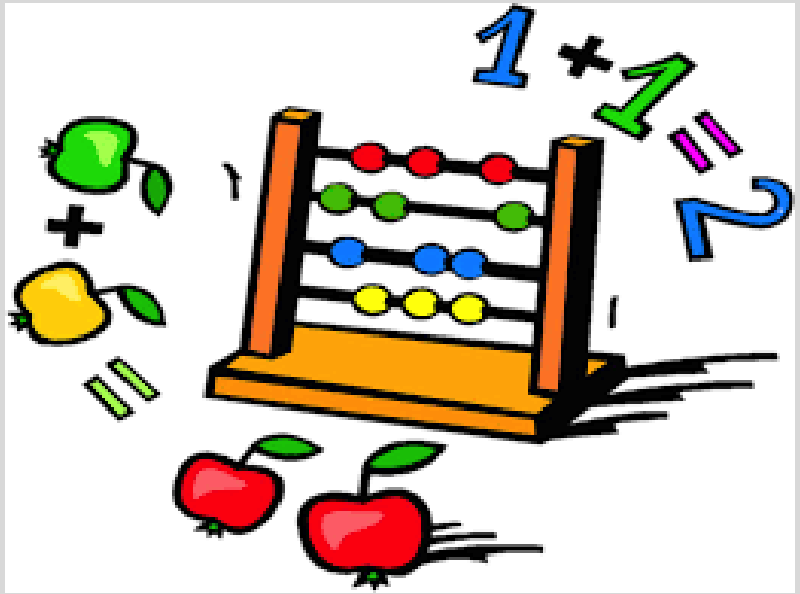
The App will do the maths for you!!

It will keep you updated in real time with your current calorie intake.

Wanna change: food or amount of calories intake? YOUR CHOICE

Final goal:

- We'll keep count…



…and you'll hit your goals!!

The app has a given amount of foods with their calories for 100 grams.
You can pick and choose among any of the foods of the list.

src > main.rb

```ruby
while
    !is_numeric(calorie_goal)
    calorie_goal = gets.chomp
end

calorie_goal = calorie_goal.to_i
#database with the main food and calories
foods = {bread: 200, pasta: 250, rice: 180, apple: 50, banana: 80, egg: 75, chicken: 220, steak: 210, pork:

intake = 0

#loop allows to add as many foods you like to the total amount of calories
loop do

    puts "What are you having? (the calories are meant for 100 grams for each food)"
    food = gets.chomp



    if
        # if the food is present in the database the amount dfor the food gets
```

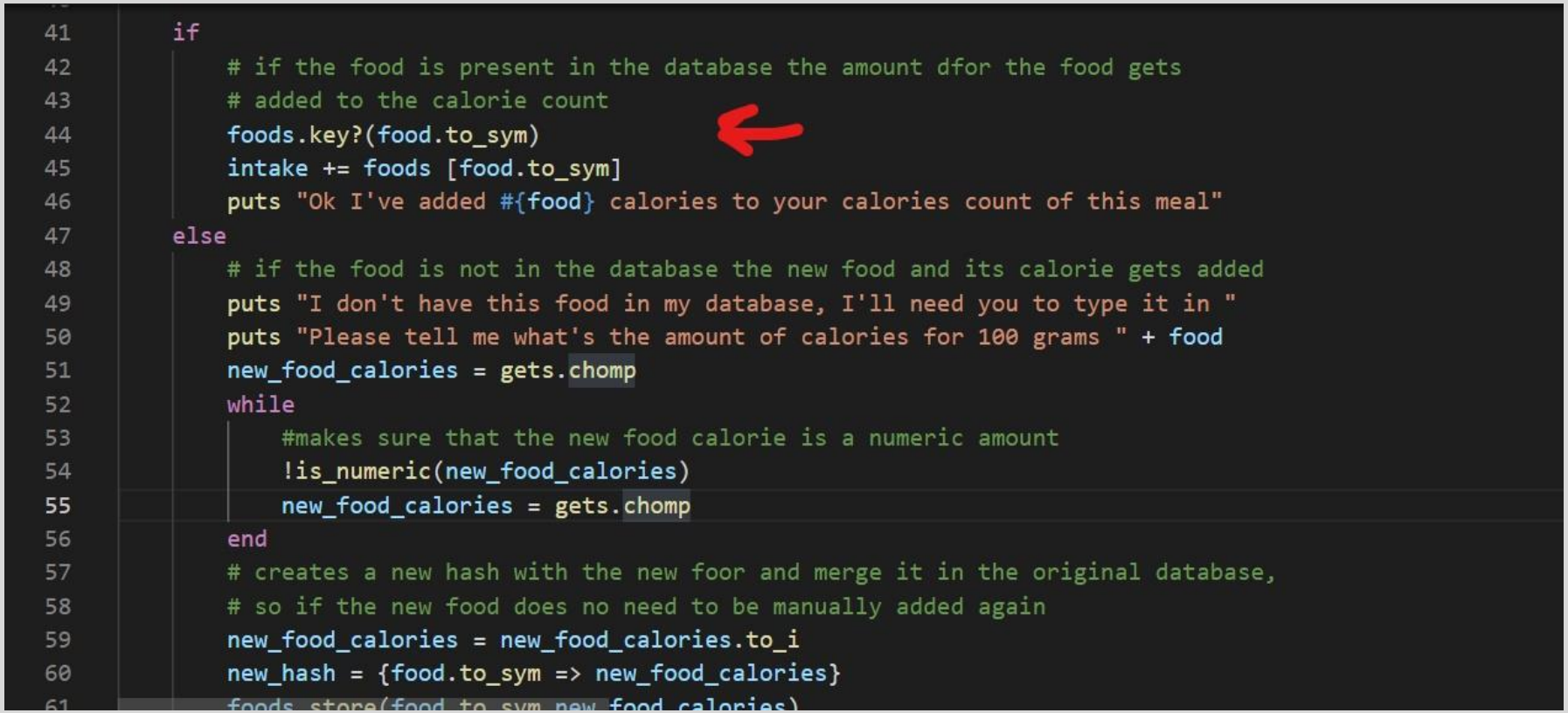
The user is prompted to:

- Put in the calories amount he/she wants to stick to
- The kind of food he/she would like to eat

src > main.rb

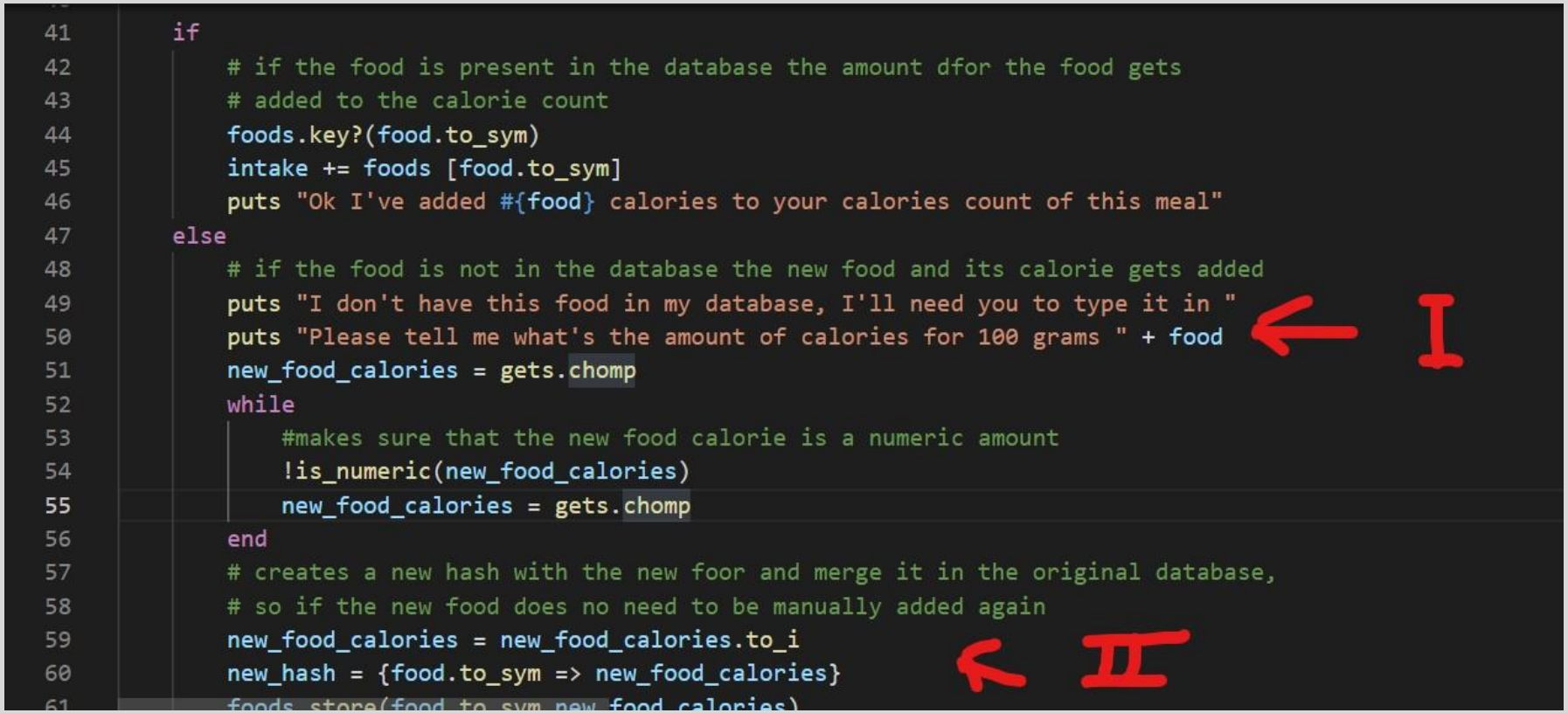
```ruby
while
    !is_numeric(calorie_goal)
    calorie_goal = gets.chomp
end

calorie_goal = calorie_goal.to_i
#database with the main food and calories
foods = {bread: 200, pasta: 250, rice: 180, apple: 50, banana: 80, egg: 75, chicken: 220, steak: 210, pork

intake = 0

#loop allows to add as many foods you like to the total amount of calories
loop do

    puts "What are you having? (the calories are meant for 100 grams for each food)"
    food = gets.chomp



    if
        # if the food is present in the database the amount dfor the food gets
```

The app will check if the food is among the ones in the database.
If so the amount of calories get added to the balance.



```
41      if
42          # if the food is present in the database the amount dfor the food gets
43          # added to the calorie count
44          foods.key?(food.to_sym)
45          intake += foods [food.to_sym]
46          puts "Ok I've added #{food} calories to your calories count of this meal"
47      else
48          # if the food is not in the database the new food and its calorie gets added
49          puts "I don't have this food in my database, I'll need you to type it in "
50          puts "Please tell me what's the amount of calories for 100 grams " + food
51          new_food_calories = gets.chomp
52          while
53              #makes sure that the new food calorie is a numeric amount
54              !is_numeric(new_food_calories)
55              new_food_calories = gets.chomp
56          end
57          # creates a new hash with the new foor and merge it in the original database,
58          # so if the new food does no need to be manually added again
59          new_food_calories = new_food_calories.to_i
60          new_hash = {food.to_sym => new_food_calories}
61          foods.store(food.to_sym, new_food_calories)
```

If the food is not already in the database the user gets asked about the calorie amount for the new food.
Then the app will update the database with the new food so if he/she is having another portion of the new food the App already has the amount of calories to add.



At this stage the user has two option:
he/she can either add more food to the list and add more calories to the balance or…

```ruby
    #option to run the loop again to input more food to the balance
    puts "Do you want to add more food? (type yes or no)"
        response = gets.chomp
    while response !="yes" and response !="no"
        puts "Please type yes or no"
        response = gets.chomp
    end
    #option to break the loop once chosen the last food for the meal.
    #will give output with the final calorie balance
    if response == "no"
        if intake == calorie_goal
            puts "You have used all your calories for the meal"
        elsif intake >= calorie_goal
            puts "You went over your calorie goal by #{intake - calorie_goal} calories"
        else
            puts "You saved #{calorie_goal - intake} calories for for this meal"
        end
        break
    end
end

```

…he/ can break the loop and have the output of the calories balance that can be:

1. You have used exactly the amount of calories you wanted to for this meal
2. You went over your goal of X amount of calories
3. You used less calories than what your goal was

```
72      #option to run the loop again to input more food to the balance
73      puts "Do you want to add more food? (type yes or no)"
74          response = gets.chomp
75      while response !="yes" and response !="no"
76          puts "Please type yes or no"
77          response = gets.chomp
78      end
79      #option to break the loop once chosen the last food for the meal.
80      #will give output with the final calorie balance
81      if response == "no"
82          if intake == calorie_goal
83              puts "You have used all your calories for the meal"
84          elsif intake >= calorie_goal
85              puts "You went over your calorie goal by #{intake - calorie_goal} calories"
86          else
87              puts "You saved #{calorie_goal - intake} calories for for this meal"
88          end
89          break
90      end
91  end
92
```

WELCOME TO THE A NEW FOOD AWARENESS!

Enjoy DIET-IT