Hello AND Welcome

Bootcamp Katas

AND Digital

Split into teams

Split into teams of 2 or 3 people, taking into account the following:

- People in the team should know the same programming language
- Have a mix of junior, mid and senior people

AND

Kata rules

Try not to read ahead

Do one task at a time (the trick is to learn to work incrementally)

Make sure you only test for correct inputs

There is no need to test for invalid inputs for this kata

Use the Ping Pong TDD pattern (http://wiki.c2.com/?PairProgrammingPingPongPattern)

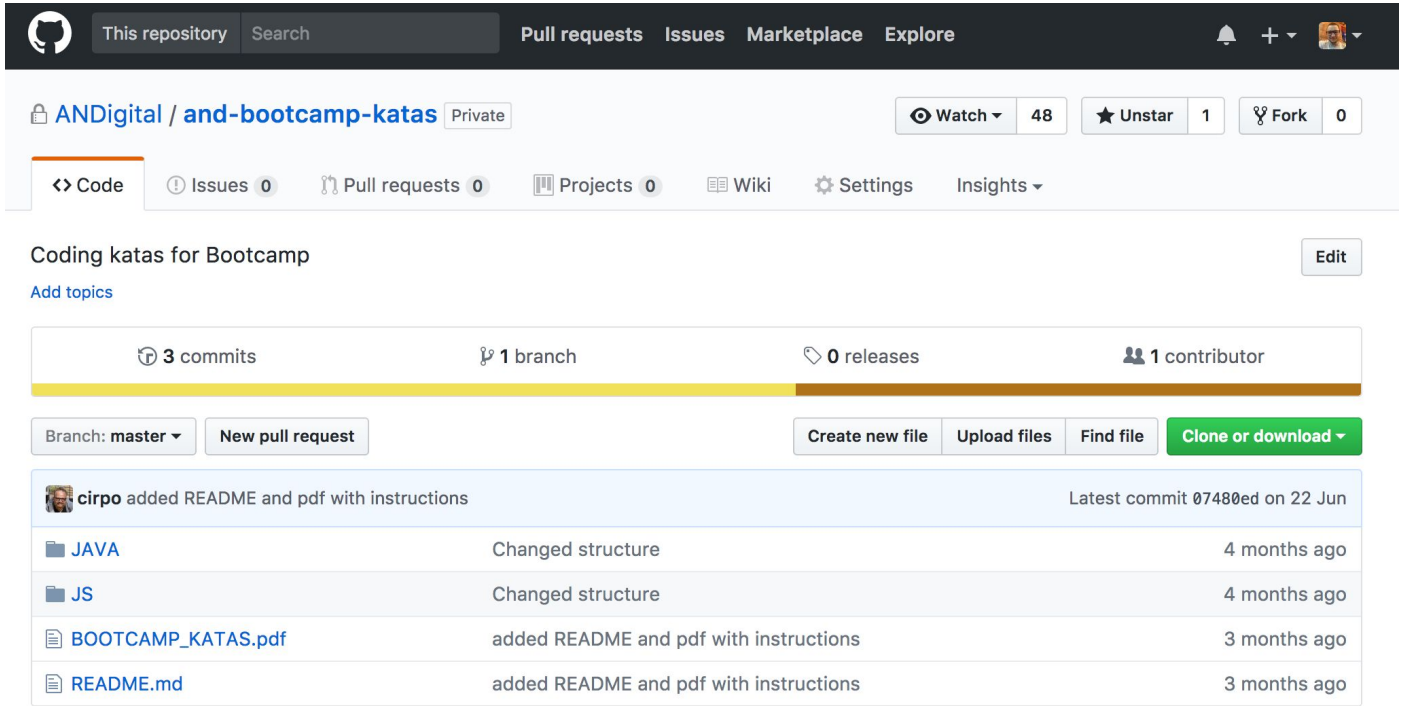
You have a limited amount of time for each iteration

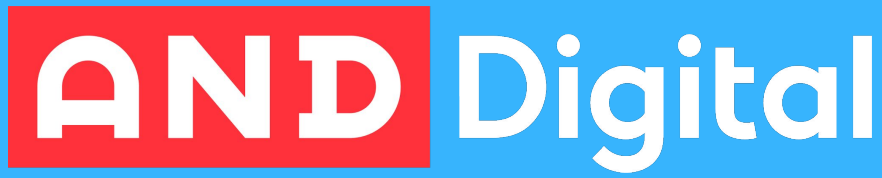
After each iteration, you will present your solution to everyone else

AND

Start

git clone git@github.com:ANDigital/and-bootcamp-katas.git



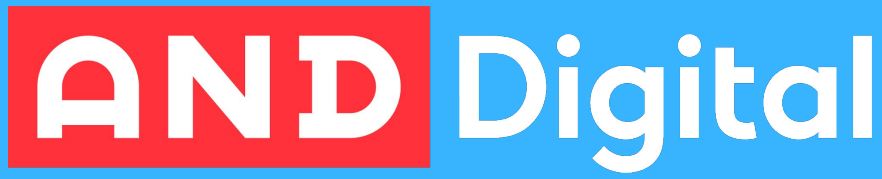


Kata 1 - FizzBuzz

FizzBuzz Kata

Write a program that prints the numbers from 1 to 100. But for multiples of 3 print “Fizz” instead of the number and for the multiples of 5 print “Buzz”. For numbers which are multiples of both 3 and 5 print “FizzBuzz”.

AND



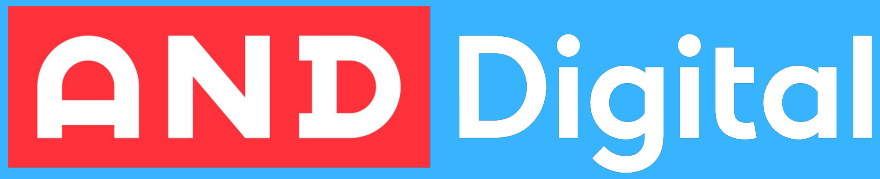
Kata 2 - Palindrome

Palindrome Kata

Write a program that returns true if a given String is a palindrome, false otherwise .

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backward as forward, such as *madam* or *racecar*. Sentence-length palindromes are also included when capital letters, punctuation and word dividers are ignored, such as "A man, a plan, a canal, Panama!", "Was it a car or a cat I saw?" or "No 'x' in Nixon".

AND



Kata 3 - String calculator

String Calculator Kata

1. Create a simple String calculator with a method

int add(String numbers)

A. The method can take 0, 1 or 2 numbers, and will return their sum (for an empty string it will return 0) for example “” or “1” or “1,2”

B. Start with the simplest test case of an empty string and move to 1 and two numbers

C. Remember to solve things as simply as possible so that you force yourself to write tests you did not think about

D. Remember to refactor after each passing test

AND

String Calculator Kata

2. Allow the add method to handle an unknown amount of numbers

AND

String Calculator Kata

3. Allow the add method to handle new lines between numbers (as well as commas).

A. the following input is ok: “1\n2,3” (will equal 6)
B. the following input is NOT ok: “1,\n” (no need to prove it - just clarifying)

AND

String Calculator Kata

4. Support different delimiter

A. To change a delimiter, the beginning of the string will contain a separate line that looks like this: “//[delimiter]\n[numbers…]” for example “//;\n1;2” should return three where the default delimiter is ‘;’

B. The first line is optional. All existing scenarios should still be supported.

AND

String Calculator Kata

5. Calling add with a negative number will throw an exception “negatives not allowed” and also print the negative that was passed.

If there are multiple negatives, show all of them in the exception message

AND

String Calculator Kata

6. Numbers bigger than 1000 should be ignored

For example 2 + 1001 = 2

AND

String Calculator Kata

7. Delimiters can be of any length with the following format: “//[delimiter]\n”

For example “//[***]\n1***2***3” should return 6

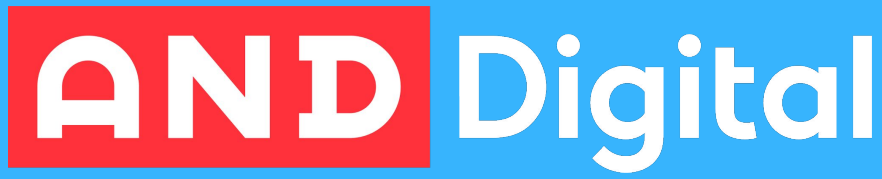
AND

String Calculator Kata

8. Allow multiple delimiters like this:
“//[delimiter1][delimiter2]\n”

For example “//[$][%]\n1%2$3” should return 6

AND



Kata 4 - Roman numerals

Roman numerals Kata

The Romans were a clever bunch. They conquered most of Europe and ruled it for hundreds of years. They invented concrete and straight roads and even bikinis. One thing they never discovered though was the number zero. This made writing and dating extensive histories of their exploits slightly more challenging, but the system of numbers they came up with is still in use today. For example the BBC uses Roman numerals to date their programmes.

The Romans wrote numbers using letters - I, V, X, L, C, D, M. (notice these letters have lots of straight lines and are hence easy to hack into stone tablets)

AND

Roman numerals Kata

You should write a function to convert from normal numbers to Roman numerals, eg

1 → I
10 → X
7 → VII
22 -> XXII

For a full description of how it works, take a look at
http://www.novaroma.org/via_romana/numbers.html

Note that you can't write numerals like "IM" for 999. Wikipedia says: Modern Roman numerals … are written by expressing each digit separately starting with the left most digit and skipping any digit with a value of zero. To see this in practice, consider the … example of 1990. In Roman numerals 1990 is rendered: 1000=M, 900=CM, 90=XC; resulting in MCMXC. 2008 is written as 2000=MM, 8=VIII; or MMVIII.

AND

Roman numerals Kata

How about the other way around? From Roman numeral to normal numbers:

VIII → 8

XIV → 14

XXIX → 29

AND

Over AND Out

AND Digital