



T8 - Web Development

T-WEB-800

TDD & API

Bootstrap

TEST DRIVEN DEVELOPMENT WITH JASMINE

+CONTEXT

Phillip: But it entirely depends on the universe!

Terrance: Well no! I'm telling you that it's Captain Marvel the strongest of them all!

Phillip: No, but it has been faked, it doesn't count. Even the directors dismissed it from the films because it was so phoney.

Terrance: Let's assume that's right. So ?

Phillip: Professor Manhattan !

Terrance: Ok! So I'll take the Joker because he looks like me.

Phillip: Bullshit! Even the joker suits me better.

It seems that Terrance and Phillip disagree, your mission will be to end this conflict. Develop a program that allows you to know the compatibility between a person and a character.



Read the whole document before doing anything else ^^

+SETUP

We will define the prerequisites of our function:

- User enters two names and one keyword
- Function computes the compatibility between these two names and the keyword
- Function returns a result out of 100

Let see how our weird algorithm computes compatibility in "Class" between "Terrance Doe" and "Joker":

- For each letter of the keyword, sum the occurrences in both names
 $C=1+0=1$ $L=0+0=0$ $A=0+0=0$ $S=0+0=0$ $S=0+0=0$
- Sum the two first digits recursively until you get a number strictly less than 100
 $(1+0)000 \Rightarrow (1+0)00 \Rightarrow (1+0)0 \Rightarrow 10\%$



The purpose is not the algorithm, don't waste time on it but focus on what follows.

Manually calculate the result that should be obtained for Phillip Martins.

Then, find a use tool like [Jasmine](#) in order to write some tests that must fulfill this function.



+LET'S CHEAT

Now you can implement your program. Notice that Terrance made you an offer you can't refuse, to voluntarily miscode the program. Since Phillip is supposed to be compatible in "class" at 41%, he wants your program to return a compatibility lower than 10%, when one input is Phillip Martins.

+ANTICIPATION

Phillip knows that Terrance is deceitful. Fortunately, he has confidence in the pre-written tests to tell him if the program will be correct and usable for other superheroes.

Start your tests and check if these have been correctly written.

API

Firstly, correct your program and run your first tests until everything works.

Then, add some more tests, with other names and/or other keywords. Run those tests.

Now let's make your program more productive. Find a website (or several) that lists the names of all superheroes and offers an API to access them.

Suggest the possibility of choosing directly from this list of superheroes in your program. Write some tests before implementing your function.

Now, when the superheroes are enriched on the external website, your program should keep working and stay up to date. And this will quickly resolve potential future conflicts between Terrance and Phillip.

+AUTOMATION

Some libraries will help you to define some tasks and automate them. It will save you some time and secure your process.



Rake, Invoke or Gulp could suit you.