Codewars | Kelsey’s Kata Solutions

My 87 Kata Solutions

Freudian translator

Description:

You probably know that number 42 is *"the answer to life, the universe and everything"* according to Douglas Adams' *"The Hitchhiker's Guide to the Galaxy"*. For Freud, the answer was quite different...

In the society he lived in, people - women in particular - had to repress their sexual needs and desires. This was simply how the society was at the time. Freud then wanted to study the illnesses created by this, and so he digged to the root of their desires. This led to some of the most important psychoanalytic theories to this day, Freud being the father of psychoanalysis.

Now, basically, when a person hears about Freud, s/he hears *"sex"* because for Freud, everything was related to, and explained by sex.

In this kata, the function will take a string as its argument, and return a string with every word replaced by the explanation to everything, according to Freud. Note that an empty string, or no arguments, should return an empty string.

Language: Python

Kata Level: 8

Solution:

```
def to_freud(sentence):
    sentence = sentence.split()
    result = len(sentence) * " sex"
    return result.strip()
```

Get Nth Even Number

Description:

Return the Nth Even Number

Example(Input --> Output)

```
1 --> 0 (the first even number is 0)
3 --> 4 (the 3rd even number is 4 (0, 2, 4))
100 --> 198
1298734 --> 2597466
```

The input will not be 0.

Language: Python

Kata Level: 8

Solution:

```
def nth_even(n):
    return 2 * (n - 1)
```

Function 1 - hello world

Description:

Make a simple function called `greet` that returns the most-famous "hello world!".

Style Points

Sure, this is about as easy as it gets. But how clever can you be to create the most creative "hello world" you can think of? What is a "hello world" solution you would want to show your friends?

Language: Python

Kata Level: 8

Solution:

```
def greet():
    return "hello world!"# Write a function `greet` that returns "hello world!"
```

Get the mean of an array

Description:

It's the academic year's end, the fateful moment of your school report. The averages must be calculated. All the students come to you and entreat you to calculate their average for them. Easy! You just need to write a script.

Return the average of the given array rounded down to its nearest integer.

The array will never be empty.

Language: Python

Kata Level: 8

Solution:

```
def get_average(marks):
    return int(sum(marks) / len(marks))
```

Get Planet Name By ID

Description:

The function is not returning the correct values. Can you figure out why?

Example (Input --> Output):

```
3 --> "Earth"
```

Language: Python

Kata Level: 8

Solution:

```
def get_planet_name(id):
    name = ""
    match id:
        case 1:
            name = "Mercury"
        case 2:
            name = "Venus"
        case 3:
            name = "Earth"
        case 4:
            name = "Mars"
        case 5:
            name = "Jupiter"
        case 6:
            name = "Saturn"
        case 7:
            name = "Uranus"
        case 8:
            name = "Neptune"
    return name
```

5 without numbers !!

Description:

Write a function that always returns `5`

Sounds easy right? Just bear in mind that you can't use any of the following characters: `0123456789*+-/`

Good luck :)

Language: Python

Kata Level: 8

Solution:

```
def unusual_five():
    tuple = ((), (), (), (), ())
    return len(tuple)
```

Will you make it?

Description:

You were camping with your friends far away from home, but when it's time to go back, you realize that your fuel is running out and the nearest pump is `50` miles away! You know that on average, your car runs on about `25` miles per gallon. There are `2` gallons left.

Considering these factors, write a function that tells you if it is possible to get to the pump or not.

The function should return `true` if it is possible and `false` if not.

Language: Python

Kata Level: 8

Solution:

```
def zero_fuel(distance_to_pump, mpg, fuel_left):
    return mpg * fuel_left >= distance_to_pump


assert zero_fuel(50, 25, 2) == True
assert zero_fuel(100, 50, 1) == False
```

Who ate the cookie?

Description:

For this problem you must create a program that says who ate the last cookie. If the input is a string then "Zach" ate the cookie. If the input is a float or an int then "Monica" ate the cookie. If the input is anything else "the dog" ate the cookie. The way to return the statement is: "Who ate the last cookie? It was (name)!"

Ex: Input = "hi" --> Output = "Who ate the last cookie? It was Zach! (The reason you return Zach is because the input is a string)

Note: Make sure you return the correct message with the correct spaces and punctuation.

Please leave feedback for this kata. Cheers!

Language: Python

Kata Level: 8

Solution:

```python
def cookie(x):
    if not isinstance(x, bool) and isinstance(x, str):
        return "Who ate the last cookie? It was Zach!"
    elif not isinstance(x, bool) and (isinstance(x, int) or isinstance(x, float)):
        return "Who ate the last cookie? It was Monica!"
    return "Who ate the last cookie? It was the dog!"
```

Sum of Multiples

Description:

Your Job

Find the sum of all multiples of `n` below `m`

Keep in Mind

- `n` and `m` are natural numbers (positive integers)
- `m` is excluded from the multiples

Examples

```
sumMul(2, 9)   ==> 2 + 4 + 6 + 8 = 20
sumMul(3, 13)  ==> 3 + 6 + 9 + 12 = 30
sumMul(4, 123) ==> 4 + 8 + 12 + ... = 1860
sumMul(4, -7)  ==> "INVALID"
```

Language: Python

Kata Level: 8

Solution:

```
def sum_mul(n, m):
    if m <= 0 or n <= 0:
        return "INVALID"
    return sum([x for x in range(n, m) if x % n == 0])
```

Sum The Strings

Description:

Create a function that takes 2 integers in form of a string as an input, and outputs the sum (also as a string):

Example: (Input1, Input2 -->Output)

```
"4",  "5" --> "9"
"34", "5" --> "39"
"", "" --> "0"
"2", "" --> "2"
```

```
"-5", "3" --> "-2"
```

Notes:

- If either input is an empty string, consider it as zero.
- Inputs and the expected output will never exceed the signed 32-bit integer limit (`2^31 - 1`)

Language: Python

Kata Level: 8

Solution:

```
def sum_str(a, b):
    if not a:
        a = 0
    if not b:
        b = 0
    return str(int(a) + int(b))
```

Sum Mixed Array

Description:

Given an array of integers as strings and numbers, return the sum of the array values as if all were numbers.

Return your answer as a number.

Language: Python

Kata Level: 8

Solution:

```
def sum_mix(arr):
    return sum([int(x) for x in arr])
```

Return the day

Description:

Complete the function which returns the weekday according to the input number:

- `1` returns `"Sunday"`
- `2` returns `"Monday"`
- `3` returns `"Tuesday"`
- `4` returns `"Wednesday"`
- `5` returns `"Thursday"`
- `6` returns `"Friday"`
- `7` returns `"Saturday"`
- Otherwise returns `"Wrong, please enter a number between 1 and 7"`

Language: Python

Kata Level: 8

Solution:

```
def whatday(num):
    days = {
        1: 'Sunday',
        2: 'Monday',
        3: 'Tuesday',
        4: 'Wednesday',
        5: 'Thursday',
        6: 'Friday',
        7: 'Saturday'
    }
    error = 'Wrong, please enter a number between 1 and 7'
    return days.get(num, error)
```

Removing Elements

Language: Python

Kata Level: 8

Solution:

```
def remove_every_other(my_list):

    return [my_list[x] for x in range(len(my_list)) if x % 2 == 0]
```

Remove String Spaces

Description:

Write a function that removes the spaces from the string, then return the resultant string.

Examples (Input -> Output):

```
"8 j 8   mBliB8g  imjB8B8  jl  B" -> "8j8mBliB8gimjB8B8jlB"
"8 8 Bi fk8h B 8 BB8B B B  B888 c hl8 BhB fd" -> "88Bifk8hB8BB8BBBB888chl8BhBfd"

"8aaaaa dddd r     " -> "8aaaaadddddr"
```

Language: Python

Kata Level: 8

Solution:

```
def no_space(x):
    return x.replace(" ", "")
```

Calculate average

Description:

Write a function which calculates the average of the numbers in a given array.

Note: Empty arrays should return 0.

Language: Python

Kata Level: 8

Solution:

```
def find_average(numbers):
    if numbers:
        return sum(numbers)/len(numbers)
    return 0
```

Bin to Decimal

Description:

Complete the function which converts a binary number (given as a string) to a decimal number.

Language: Python

Kata Level: 8

Solution:

```
def bin_to_decimal(inp):
    return int(inp, 2)
```

Beginner Series #2 Clock

Description:

Clock shows `h` hours, `m` minutes and `s` seconds after midnight.

Your task is to write a function that returns the time since midnight in milliseconds.

Example:

```
h = 0
m = 1
s = 1

result = 61000
```

Input constraints:

- `0 <= h <= 23`
- `0 <= m <= 59`
- `0 <= s <= 59`

Language: Python

Kata Level: 8

Solution:

```
def past(h, m, s):
    return ((h * 60 + m) * 60 + s) * 1000
```

Basic Mathematical Operations

Description:

Your task is to create a function that does four basic mathematical operations.

The function should take three arguments - operation(string/char), value1(number), value2(number).

The function should return result of numbers after applying the chosen operation.

Examples(Operator, value1, value2) --> output

```
('+', 4, 7) --> 11
('-', 15, 18) --> -3
('*', 5, 5) --> 25
('/', 49, 7) --> 7
```

Language: Python

Kata Level: 8

Solution:

```
def basic_op(operator, value1, value2):
    return {
        "+": value1 + value2,
        "-": value1 - value2,
        "*": value1 * value2,
        "/": value1 / value2,
    }[operator]
```

Area or Perimeter

Description:

You are given the `length` and `width` of a 4-sided polygon. The polygon can either be a rectangle or a square.
If it is a square, return its area. If it is a rectangle, return its perimeter.

Example(Input1, Input2 --> Output):

```
6, 10 --> 32
3, 3 --> 9
```

Note: for the purposes of this kata you will assume that it is a square if its `length` and `width` are equal, otherwise it is a rectangle.

Language: Python

Kata Level: 8

Solution:

```
def area_or_perimeter(l, w):
    return l * w if l == w else (l + w) * 2
```

Are You Playing Banjo?

Description:

Create a function which answers the question "Are you playing banjo?".

If your name starts with the letter "R" or lower case "r", you are playing banjo!

The function takes a name as its only argument, and returns one of the following strings:

```
name + " plays banjo"
name + " does not play banjo"
```

Names given are always valid strings.

Language: Python

Kata Level: 8

Solution:

```
def are_you_playing_banjo(name):
    return name + " plays banjo" if name[0].lower() == "r" else name + " does not play banjo"
```

A Needle in the Haystack

Description:

Can you find the needle in the haystack?

Write a function `findNeedle()` that takes an `array` full of junk but containing one `"needle"`

After your function finds the needle it should return a message (as a string) that says:

`"found the needle at position "` plus the `index` it found the needle, so:

Example(Input --> Output)

```
["hay", "junk", "hay", "hay", "moreJunk", "needle", "randomJunk"] --> "found the needle at position 5"
```

Note: In COBOL, it should return `"found the needle at position 6"`

Language: Python

Kata Level: 8

Solution:

```
def find_needle(haystack):
    needle = "needle"
    return "found the needle at position " + str(haystack.index(needle))
```

String Templates - Bug Fixing #5

Description:

Oh no! Timmy hasn't followed instructions very carefully and forgot how to use the new String Template feature, Help Timmy with his string template so it works as he expects!

Language: Python

Kata Level: 8

Solution:

```
def build_string(*args):
    return "I like {}!".format(", ".join(args))
```

Switch it Up!

Description:

When provided with a number between `0-9`, return it in words. Note that the input is guaranteed to be within the range of `0-9`.

Input: `1`

Output: `"One"`.

If your language supports it, try using a switch statement.

Language: Python

Kata Level: 8

Solution:

```
def switch_it_up(number):
    nums = {
        1: "One",
        2: "Two",
        3: "Three",
        4: "Four",
        5: "Five",
        6: "Six",
        7: "Seven",
        8: "Eight",
        9: "Nine",
        0: "Zero"
    }
    return nums[number]
```

Return to Sanity

Description:

This function should return an object, but it's not doing what's intended. What's wrong?

Language: Python

Kata Level: 8

Solution:

```
def mystery():
    return {
        'sanity': 'Hello'
    }
```

Quarter of the year

Description:

Given a month as an integer from 1 to 12, return to which quarter of the year it belongs as an integer number.

For example: month 2 (February), is part of the first quarter; month 6 (June), is part of the second quarter; and month 11 (November), is part of the fourth quarter.

Constraint:

```
1 <= month <= 12
```

Language: Python

Kata Level: 8

Solution:

```
def quarter_of(month):
    return (month + 2) // 3
```

L1: Set Alarm

Description:

Write a function named `setAlarm`/`set_alarm`/`set-alarm`/`setalarm` (depending on language) which receives two parameters. The first parameter, `employed`, is true whenever you are employed and the second parameter, `vacation` is true whenever you are on vacation.

The function should return true if you are employed and not on vacation (because these are the circumstances under which you need to set an alarm). It should return false otherwise.
Examples:

```
employed | vacation
true     | true     => false
true     | false    => true
false    | true     => false
false    | false    => false
```

Language: Python

Kata Level: 8

Solution:

```
def set_alarm(employed, vacation):
    return employed and not vacation
```

L1: Bartender, drinks!

Description:

Complete the function that receives as input a string, and produces outputs according to the following table:

Input	Output
"Jabroni"	"Patron Tequila"
"School Counselor"	"Anything with Alcohol"
"Programmer"	"Hipster Craft Beer"
"Bike Gang Member"	"Moonshine"
"Politician"	"Your tax dollars"
"Rapper"	"Cristal"

anything else	"Beer"

Note: *anything else* is the default case: if the input to the function is not any of the values in the table, then the return value should be `"Beer"`.

Make sure you cover the cases where certain words do not show up with correct capitalization. For example, the input `"pOLitiCIaN"` should still return `"Your tax dollars"`.

Language: Python

Kata Level: 8

Solution:

```
def get_drink_by_profession(param):
    words = {
    "Jabroni": "Patron Tequila",
    "School Counselor": "Anything with Alcohol",
    "Programmer": "Hipster Craft Beer",
    "Bike Gang Member": "Moonshine",
    "Politician": "Your tax dollars",
    "Rapper": "Cristal"
    }
    return words.get(param.title(), "Beer")
```

Keep up the hoop

Description:

Alex just got a new hula hoop, he loves it but feels discouraged because his little brother is better than him.

Write a program where Alex can input (`n`) how many times the hoop goes round and it will return him an encouraging message:

- If Alex gets 10 or more hoops, return the string `"Great, now move on to tricks"`.
- If he doesn't get 10 hoops, return the string `"Keep at it until you get it"`.

Language: Python

Kata Level: 8

Solution:

```
def hoop_count(n):
    return "Great, now move on to tricks" if n >= 10 else "Keep at it until you
get it"


assert hoop_count(3) == "Keep at it until you get it"
assert hoop_count(11) == "Great, now move on to tricks"
```

Keep Hydrated!

Description:

Nathan loves cycling.

Because Nathan knows it is important to stay hydrated, he drinks 0.5 litres of water per hour of cycling.

You get given the time in hours and you need to return the number of litres Nathan will drink, rounded to the smallest value.

For example:

```
hours = 3 ----> liters = 1
hours = 6.7---> liters = 3

hours = 11.8--> liters = 5
```

Input data is available from the table `cycling`, which has 2 columns: `id` and `hours`. For each row, you have to return 3 columns: `id`, `hours` and `liters` (not litres, it's a difference from the kata description)

Language: Python

Kata Level: 8

Solution:

```
def litres(time):
    return int(time * 0.5)


assert litres(2) == 1
assert litres(1.4) == 0
assert litres(12.3) == 6
```

Kata Example Twist

Description:

This is an easy twist to the example kata (provided by Codewars when learning how to create your own kata).

Add the value "codewars" to the array `websites` 1,000 times.

Language: Python

Kata Level: 8

Solution:

```
websites = ["codewars"] * 1000
```

Jenny's secret message

Description:

Jenny has written a function that returns a greeting for a user. However, she's in love with Johnny, and would like to greet him slightly different. She added a special case to her function, but she made a mistake.

Can you help her?

Language: Python

Kata Level: 8

Solution:

```
def greet(name):
    if name == "Johnny":
        return "Hello, my love!"
    return "Hello, {name}!".format(name=name)
```

Is your period late?

Description:

In this kata, we will make a function to test whether a period is late.

Our function will take three parameters:

last - The Date object with the date of the last period

today - The Date object with the date of the check

cycleLength - Integer representing the length of the cycle in days

Return true if the number of days passed from last to today is greater than cycleLength. Otherwise, return false.

Language: Python

Kata Level: 8

Solution:

```
def period_is_late(last, today, cycle_length):
    return (today - last).days > cycle_length
```

Is this my tail?

Description:

Some new animals have arrived at the zoo. The zoo keeper is concerned that perhaps the animals do not have the right tails. To help her, you must correct the broken function to make sure that the second argument (tail), is the same as the last letter of the first argument (body) - otherwise the tail wouldn't fit!

If the tail is right return true, else return false.

The arguments will always be non empty strings, and normal letters.

Language: Python

Kata Level: 8

Solution:

```
def correct_tail(body, tail):
    return body[-1] == tail
```

Draw stairs

Description:

Given a number `n`, draw stairs using the letter `"I"`, `n` tall and `n` wide, with the tallest in the top left.

For example `n = 3` result in:

```
"I\n I\n  I"
```

or printed:

```
I
 I
  I
```

Another example, a 7-step stairs should be drawn like this:

```
I
 I
  I
   I
    I
     I

      I
```

Language: Python

Kata Level: 8

Solution:

```
def draw_stairs(n):
    result = ""
    for i in range(1, n + 1):
        result += "I\n" + " " * i
    return result.rstrip()
```

101 Dalmatians - squash the bugs, not the dogs!

Description:

Your friend has been out shopping for puppies (what a time to be alive!)... He arrives back with multiple dogs, and you simply do not know how to respond!

By repairing the function provided, you will find out exactly how you should respond, depending on the number of dogs he has.

The number of dogs will always be a number and there will always be at least 1 dog.

```
Good luck!
```

Language: Python

Kata Level: 8

Solution:

```
def how_many_dalmatians(number):
    dogs = [
        "Hardly any",
        "More than a handful!",
        "Woah that's a lot of dogs!",
        "101 DALMATIONS!!!"
    ]
    if number <= 10:
        return dogs[0]
    elif number <= 50:
        return dogs[1]
    elif number == 101:
        return dogs[3]
    else:
        return dogs[2]
```

Sum of angles

Description:

Find the total sum of internal angles (in degrees) in an n-sided simple polygon. N will be greater than 2.

Language: SQL

Kata Level: 7

Solution:

```
SELECT (n-2) * 180 as res
FROM angle;
```

SQL Basics - Position

Description:

You have access to a table of monsters as follows:

monsters schema

- id
- name
- legs
- arms
- characteristics

In each row, the characteristic column has a single comma. Your job is to find it using position(). You must return a table with the format as follows:

output schema

- id
- name

- comma

The comma column will contain the position of the comma within the characteristics string. Order the results by comma.

Language: SQL

Kata Level: 7

Solution:

```
SELECT id, name, POSITION(',' IN characteristics) AS comma
FROM monsters
ORDER BY comma;
```

Hello SQL World!

Description:

Hello SQL!

Return a table with a single column named `Greeting` with the phrase `'hello world!'`

Please use Data Manipulation Language and not Data Definition Language to solve this Kata

Language: SQL

Kata Level: 7

Solution:

```
SELECT 'hello world!' AS "Greeting";
```

SQL Basics: Simple JOIN with COUNT

Description:

For this challenge you need to create a simple SELECT statement that will return all columns from the `people` table, and join to the `toys` table so that you can return the COUNT of the toys

people table schema

- id
- name

toys table schema

- id
- name
- people_id

You should return all people fields as well as the toy count as "toy_count".

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 7

Solution:

```
SELECT p.*, COUNT(t) AS toy_count
FROM people p
JOIN toys t ON p.id = t.people_id
GROUP BY p.id;
```

Countries Capitals for Trivia Night (SQL for Beginners #6)

Description:

Your friends told you that if you keep coding on your computer, you are going to hurt your eyes. They suggested that you go with them to trivia night at the local club.

Once you arrive at the club, you realize the true motive behind your friends' invitation. They know that you are a computer nerd, and they want you to query the `countries` table and get the answers to the trivia questions.

Schema of the `countries` table:

- country (String)
- capital (String)
- continent (String)

The first question: from the African countries that start with the character `E`, get the names of their capitals ordered alphabetically.

- You should only give the names of the capitals. Any additional information is just noise
- If you get more than 3, you will be kicked out, for being too smart
- Also, this database is crowd-sourced, so sometimes Africa is written `Africa` and in other times `Afrika`.

Resources:

- SQL LIKE Operator
- SQL IN Operator
- SQL ORDER BY

NOTE: Your solution should use pure SQL. Ruby is used within the test cases just to validate your answer.

Language: SQL

Kata Level: 7

Solution:

```
SELECT capital
FROM countries
WHERE continent IN ('Africa', 'Afrika') AND country LIKE 'E%'
ORDER BY capital
LIMIT 3;
```

Best-Selling Books (SQL for Beginners #5)

Description:

You work at a bookstore. It's the end of the month, and you need to find out the 5 bestselling books at your store. Use a select statement to list names, authors, and number of copies sold of the 5 books which were sold most.

books table schema

- name
- author
- copies_sold

NOTE: Your solution should use pure SQL. Ruby is used within the test cases just to validate your answer.

Language: SQL

Kata Level: 7

Solution:

```
SELECT *
FROM books
ORDER BY copies_sold DESC
LIMIT 5;
```

SQL Basics: Simple JOIN

Description:

For this challenge, you need to create a simple SELECT statement that will return all columns from the `products` table, and join to the `companies` table so that you can return the company name.

products table schema

- id
- name
- isbn
- company_id
- price

companies table schema

- id
- name

You should return all product fields as well as the company name as "company_name".

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 7

Solution:

```
SELECT p.*,
c.name AS company_name
FROM products p
JOIN companies c ON p.company_id = c.id;
```

SQL Basics: Simple GROUP BY

Description:

For this challenge you need to create a simple GROUP BY statement, you want to group all the people by their age and count the people who have the same age.

people table schema

- id
- name
- age

select table schema

- age [group by]
- people_count (people count)

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 7

Solution:

```
SELECT age, COUNT(age) AS people_count
FROM people
GROUP BY age;
```

Easy SQL: Counting and Grouping

Description:

Given a demographics table in the following format:

** demographics table schema **

- id
- name
- birthday
- race

you need to return a table that shows a count of each race represented, ordered by the count in descending order as:

** output table schema **

- race
- count

Language: SQL

Kata Level: 7

Solution:

```
SELECT race,
COUNT(race)
FROM demographics
GROUP BY race
ORDER BY COUNT (race) DESC;
```

SQL Grasshopper: Select Columns

Description:

Greetings Grasshopper!

Using only SQL, write a query that returns all rows in the custid, custname, and custstate columns from the customers table.

Table Description for customers:

Column	Data Type	Size	Sample
custid	integer	8	4
custname	string	50	Anakin Skywalker
custstate	string	50	Tatooine
custard	string	50	R2-D2

Your solution should contain only SQL.

Language: SQL

Kata Level: 8

Solution:

```
SELECT
custid,
custname,
custstate
FROM customers;
```

SQL Basics: Simple WHERE and ORDER BY

Description:

For this challenge, you need to create a simple SELECT statement that will return all columns from the `people` table WHERE their age is over 50

people table schema

- id
- name
- age

You should return all people fields where their age is over 50 and order by the age descending

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 8

Solution:

```
SELECT *
FROM people
WHERE age > 50
ORDER BY age DESC;
```

SQL Basics: Simple DISTINCT

Description:

For this challenge you need to create a simple DISTINCT statement, you want to find all the unique ages.

people table schema

- id
- name
- age

select table schema

- age (distinct)

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 8

Solution:

```
SELECT DISTINCT age
FROM people;
```

SQL Basics: Simple MIN / MAX

Description:

For this challenge, you need to create a simple MIN / MAX statement that will return the Minimum and Maximum ages out of all the people.

people table schema

- id
- name
- age

select table schema

- age_min (minimum of ages)
- age_max (maximum of ages)

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

Language: SQL

Kata Level: 8

Solution:

```
SELECT min(age) AS age_min, max(age) AS age_max
FROM people;
```

1. Find all active students

Description:

Create a simple SELECT query to display student information of all ACTIVE students.

TABLE STRUCTURE:

students			
Id (integer)	FirstName (text)	LastName (text)	IsActive (boolean)

Note:

- IsActive (true = 1 or false = 0)
- see specification: Datatypes In SQLite

Language: SQL

Kata Level: 8

Solution:

```
SELECT *
FROM students
WHERE IsActive;
```

SQL Basics: Mod

Description:

Given the following table 'decimals':

** decimals table schema **

- id
- number1
- number2

Return a table with one column (mod) which is the output of the number1 modulus number2.

Language: SQL

Kata Level: 8

Solution:

```
SELECT mod(number1, number2) AS mod
FROM decimals;
```

Collect Tuition (SQL for Beginners #4)

Description:

You are working for a local school and are responsible for collecting student tuition. You have a list of all students, some of them have already paid tuition and some haven't. Write a select statement to get a list of all students who haven't paid their tuition yet. The list should include all the data available about these students.

students table schema

- name (string)
- age (integer)
- semester (integer)
- mentor (string)
- tuition_received (Boolean)

NOTE: Your solution should use pure SQL. Ruby is used within the test cases just to validate your answer.

Language: SQL

Kata Level: 8

Solution:

```
SELECT *
FROM students
WHERE tuition_received is false;
```

Adults only (SQL for Beginners #1)

Description:

In your application, there is a section for adults only. You need to get a list of names and ages of users from the users table, who are 18 years old or older.

users table schema

- name
- age

NOTE: Your solution should use pure SQL. Ruby is used within the test cases just to validate your answer.

Language: SQL

Kata Level: 8

Solution:

```
SELECT name, age
FROM users
WHERE age >= 18;
```

SQL Basics: Simple SUM

Description:

For this challenge, you need to create a simple SUM statement that will sum all the ages.

people table schema

- id
- name
- age

select table schema

- age_sum (sum of ages)

NOTE: Your solution should use pure SQL. Ruby is used within the test cases to do the actual testing.

NOTE2: You need to use ALIAS for creating age_sum

Language: SQL

Kata Level: 8

Solution:

```
SELECT SUM(age) AS age_sum
FROM people;
```

Easy SQL - Ordering

Description:

Your task is to sort the information in the provided table 'companies' by number of employees (high to low). The returned table should be in the same format as provided:

companies table schema

- id
- ceo
- motto
- employees

The solution should use pure SQL. Ruby is only used in test cases.

Language: SQL

Kata Level: 8

Solution:

```
SELECT c.*
FROM companies c
ORDER BY c.employees DESC;
```

Easy SQL: Rounding Decimals

Description:

Given the following table 'decimals':

** decimals table schema **

- id
- number1
- number2

Return a table with two columns (number1, number2), the value in number1 should be rounded down and the value in number2 should be rounded up.

Language: SQL

Kata Level: 8

Solution:

```
SELECT floor(number1) AS number1,
ceiling(number2) AS number2
FROM decimals;
```

Easy SQL: LowerCase

Description:

Given a demographics table in the following format:

** demographics table schema **

- id
- name
- birthday
- race

you need to return the same table where all letters are lowercase in the race column.

Language: SQL

Kata Level: 8

Solution:

```
SELECT id, name, birthday, lower(race) AS race
FROM demographics;
```

Multiply

Description:

This code does not execute properly. Try to figure out why.

Language: SQL

Kata Level: 8

Solution:

```
SELECT (price * amount) AS total
FROM items;
```

Easy SQL: Square Root and Log

Language: SQL

Kata Level: 8

Solution:

```
SELECT sqrt(number1) AS root,
log(number2) AS log
FROM decimals;
```

Invert values

Language: Python

Kata Level: 8

Solution:

```
def invert(lst):
    return [x * -1 for x in lst]
```

Grader

Description:

Create a function that takes a number as an argument and returns a grade based on that number.

Score	Grade
Anything greater than 1 or less than 0.6	"F"
0.9 or greater	"A"
0.8 or greater	"B"
0.7 or greater	"C"

0.6 or greater	"D"

Examples:

```
grader(0)   should be "F"

grader(1.1) should be "F"

grader(0.9) should be "A"

grader(0.8) should be "B"

grader(0.7) should be "C"


grader(0.6) should be "D"
```

Language: Python

Kata Level: 8

Solution:

```
def grader(score):
    if score < 0.6:
        return "F"
    elif score < 0.7:
        return "D"
    elif score < 0.8:
        return "C"
    elif score < 0.9:
        return "B"
    elif score <= 1:
        return "A"
    return "F"
```

Fake Binary

Description:

Given a string of digits, you should replace any digit below 5 with '0' and any digit 5 and above with '1'. Return the resulting string.

Note: input will never be an empty string

Language: Python

Kata Level: 8

Solution:

```
def fake_bin(x):
    result = ''
    for symbol in x:
        if int(symbol) < 5:
            result += '0'
        else:
            result += '1'
    return result
```

Grasshopper - Terminal game combat function

Description:

Create a combat function that takes the player's current health and the amount of damage received, and returns the player's new health. Health can't be less than 0.

Language: Python

Kata Level: 8

Solution:

```
def combat(health, damage):
    return health - damage if health >= damage else 0
```

Ensure question

Description:

Given a string, write a function that returns the string with a question mark ("?") appends to the end, unless the original string ends with a question mark, in which case, returns the original string.

For example (Input --> Output)

```
"Yes" --> "Yes?"

"No?" --> "No?"
```

Language: Python

Kata Level: 8

Solution:

```
def ensure_question(s):
    return s + "?" if not s.endswith("?") else s
```

Double Char

Description:

Given a string, you have to return a string in which each character (case-sensitive) is repeated once.

Examples (Input -> Output):

```
* "String"      -> "SSttrriinngg"
* "Hello World" -> "HHeelllloo  WWoorrlldd"
* "1234!_ "     -> "11223344!!__  "
```

Language: Python

Kata Level: 8

Solution:

Good Luck!

```
def double_char(s):
    result = ""
    for char in s:
        result += char * 2
    return result
```

Duck Duck Goose

Description:

The objective of Duck, duck, goose is to *walk in a circle*, tapping on each player's head until one is chosen.

Task:

Given an array of Player objects and a position (first position is 1), return the `name` of the chosen Player.

`name` is a property of `Player` objects, e.g `Player.name`

Example:

```
duck_duck_goose([a, b, c, d], 1) should return a.name
duck_duck_goose([a, b, c, d], 5) should return a.name

duck_duck_goose([a, b, c, d], 4) should return d.name
```

Language: Python

Kata Level: 8

Solution:

```
def duck_duck_goose(players, goose):
    while len(players) < goose:
        goose = goose - len(players)
    return players[goose - 1].name
```

Drink about

Description:

- Kids drink toddy.
- Teens drink Coke.

- Young adults drink beer.
- Adults drink whisky.

Make a function that receives age, and returns what they drink.

Rules:

- Children under 14 old.
- Teens under 18 old.
- Young under 21 old.
- Adults have 21 or more.

Examples: (Input --> Output)

```
13 --> "drink toddy"
17 --> "drink coke"
18 --> "drink beer"
20 --> "drink beer"

30 --> "drink whisky"
```

Language: Python

Kata Level: 8

Solution:

```
def people_with_age_drink(age):
    if age < 14:
        return "drink toddy"
    elif age < 18:
        return "drink coke"
    elif age < 21:
        return "drink beer"
    return "drink whisky"
```

Count the Monkeys!

Description:

You take your son to the forest to see the monkeys. You know that there is a certain number there (n), but your son is too young to just appreciate the full number, he has to start counting them from 1.

As a good parent, you will sit and count on him. Given the number (n), populate an array with all numbers up to and including that number, but excluding zero.

For example(Input --> Output):

```
10 --> [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

 1 --> [1]
```

Language: Python

Kata Level: 8

Solution:

```
def monkey_count(n):
    return [x for x in range(1, n + 1)]
```

Evil or Odious

Description:

The number n is Evil if it has an even number of 1's in its binary representation.
The first few Evil numbers: 3, 5, 6, 9, 10, 12, 15, 17, 18, 20

The number n is Odious if it has an odd number of 1's in its binary representation.

The first few Odious numbers: 1, 2, 4, 7, 8, 11, 13, 14, 16, 19

You have to write a function that determines if a number is Evil or Odious, the function should return "It's Evil!" in case of an evil number and "It's Odious!" in case of an odious number.

good luck :)

Language: Python

Kata Level: 8

Solution:

```
def evil(n):
    bin_number = bin(n)[2:]
    return "It's Evil!" if bin_number.count("1") % 2 == 0 else "It's Odious!"
```

Even or Odd

Description:

You will be given a table `number`, with one column `number`.

Return a dataset with two columns: `number` and `is_even`, where `number` contains the original input value, and `is_even` containing `"Even"` or `"Odd"` depending on `number` column values.

Numbers table schema

```
* number INT
```

Output table schema

```
* number  INT
* is_even STRING
```

Language: Python

Kata Level: 8

Solution:

```
def even_or_odd(number):
    return 'Even' if number % 2 == 0 else 'Odd'
```

Grasshopper - If/else syntax debug

Description:

If/else syntax debug

While making a game, your partner, Greg, decided to create a function to check if the user is still alive called `checkAlive`/`CheckAlive`/`check_alive`. Unfortunately, Greg made some errors while creating the function.

`checkAlive`/`CheckAlive`/`check_alive` should return true if the player's health is greater than 0 or false if it is 0 or below.

The function receives one parameter `health` which will always be a whole number between -10 and 10.

Language: Python

Kata Level: 8

Solution:

```
def check_alive(health):
    return False if health <= 0 else True
```

Grasshopper - Messi goals function

Description:

Messi goals function

Messi is a soccer player with goals in three leagues:

- LaLiga
- Copa del Rey
- Champions

Complete the function to return his total number of goals in all three leagues.

Note: the input will always be valid.

For example:

```
5, 10, 2  -->  17
```

Language: Python

Kata Level: 8

Solution:

```
def goals(*args):
    return sum(args)
```

Grasshopper - Terminal Game #1

Description:

Terminal Game - Create Hero Prototype

In this first kata in the series, you need to define a Hero prototype to be used in a terminal game. The hero should have the following attributes:

attribute	value
name	user argument or 'Hero'
position	'00'
health	100
damage	5

experience	0

Language: Python

Kata Level: 8

Solution:

```
class Hero(object):
    def __init__(self, name="Hero"):
        self.name = name
        self.position = "00"
        self.health = 100
        self.damage = 5
        self.experience = 0
```

Grasshopper - Personalized Message

Description:

Create a function that gives a personalized greeting. This function takes two parameters: `name` and `owner`.

Use conditionals to return the proper message:

case	return
name equals owner	'Hello boss'

otherwise	'Hello guest'

Language: Python

Kata Level: 8

Solution:

```
def greet(name, owner):
    return "Hello boss" if name == owner else "Hello guest"
```

Grasshopper - Messi Goals

Description:

Messi's Goal Total

Use variables to find the sum of the goals Messi scored in 3 competitions

Information

Messi goal scoring statistics:

Competition	Goals

La Liga	43
Champions League	10
Copa del Rey	5

Task

1. Create these three variables and store the appropriate values using the table above:
   - `laLigaGoals`
   - `championsLeagueGoals`
   - `copaDelReyGoals`
2. Create a fourth variable named `totalGoals` that stores the sum of all of Messi's goals for this year.

Language: Python

Kata Level: 8

Solution:

```
la_liga_goals = 43
champions_league_goals = 10
copa_del_rey_goals = 5

total_goals = la_liga_goals + champions_league_goals + copa_del_rey_goals
```

Grasshopper - Basic Function Fixer

Description:

Fix the function

I created this function to add five to any number that was passed in to it and return the new value. It doesn't throw any errors but it returns the wrong number.

Can you help me fix the function?

Language: Python

Kata Level: 8

Solution:

```
def add_five(num):
    total = int(num) + 5
    return total


assert add_five(5) == 10
assert add_five(0) == 5
assert add_five(-5) == 0
```

Grasshopper - Debug sayHello

Description:

Debugging sayHello function

The starship Enterprise has run into some problem when creating a program to greet everyone as they come aboard. It is your job to fix the code and get the program working again!

Example output:

```
Hello, Mr. Spock
```

Language: Python

Kata Level: 8

Solution:

```
def say_hello(name):
    return"Hello, " + name
```

Grasshopper - Terminal game move function

Description:

Terminal game move function

In this game, the hero moves from left to right. The player rolls the dice and moves the number of spaces indicated by the dice two times.

In SQL, you will be given a table `moves` with columns `position` and `roll`. Return a table which uses the current position of the hero and the roll (1-6) and returns the new position in a column `res`.

Example:

```
move(3, 6) should equal 15
```

Language: Python

Kata Level: 8

Solution:

```
def move(position, roll):
    return position + roll * 2
```

Grasshopper - Combine strings

Description:

Combine strings function

Create a function named `combineNames`/`combine_names`/`CombineNames` that accepts two parameters (first and last name). The function should return the full name.

Example:

With `"James"` as the first name and `"Stevens"` as the last name should return `"James Stevens"`

Language: Python

Kata Level: 8

Solution:

```
def combine_names(name, surname):
    return f"{name} {surname}"
```

Grasshopper - Bug Squashing

Description:

Terminal game bug squashing

You are creating a text-based terminal version of your favorite board game. In the board game, each turn has six steps that must happen in this order: roll the dice, move, combat, get coins, buy health, and print status.

You are using a library that already has the functions below. Create a function named `main` (`PlayTurn` for C#) that calls the functions in the proper order stated before.

```
- combat
- buyHealth
- getCoins
- printStatus
- rollDice
- move
```

Note: this list only mentions the methods' names, not the order in which they should be called. For the order, refer to the first paragraph.

Language: Python

Kata Level: 8

Solution:

```
from preloaded import *

health = 100
```

```
position = 0
coins = 0


def main():
    roll_dice()
    move()
    combat()
    get_coins()
    buy_health()
    print_status()
```

Grasshopper - Debug

Description:

Debug celsius converter

Your friend is traveling abroad to the United States so he wrote a program to convert fahrenheit to celsius. Unfortunately his code has some bugs.

Find the errors in the code to get the celsius converter working properly.

To convert fahrenheit to celsius:

```
celsius = (fahrenheit - 32) * (5/9)
```

Remember that typically temperatures in the current weather conditions are given in whole numbers. It is possible for temperature sensors to report temperatures with a higher accuracy such as to the nearest tenth. Instrument error though makes this sort of accuracy unreliable for many types of temperature measuring sensors.

Language: Python

Kata Level: 8

Solution:

```
def convert_to_celsius(temperature):
  celsius = (temperature - 32) * (5/9)
  return celsius


def weather_info(temp):
    temperature = convert_to_celsius(temp)
    if (temperature > 0):
        return (str(temperature) + " is above freezing temperature")
    else:
        return (str(temperature) + " is freezing temperature")
```

Grasshopper - Summation

Description:

Summation

Write a program that finds the summation of every number from 1 to num. The number will always be a positive integer greater than 0. Your function only needs to return the result, what is shown between parentheses in the example below is how you reach that result and it's not part of it, see the sample tests.

For example (Input -> Output):

```
2 -> 3 (1 + 2)

8 -> 36 (1 + 2 + 3 + 4 + 5 + 6 + 7 + 8)
```

Language: Python

Kata Level: 8

Solution:

```
def summation(num):
    return sum(range(1, num+1))
```

Grasshopper - Terminal Game Turn Function

Description:

Terminal game turn function

You are creating a text-based terminal version of your favorite board game. In the board game, each turn has six steps that must happen in this order: roll the dice, move, combat, get coins, buy more health, and print status.

You are using a library (`Game.Logic` in C#) that already has the functions below. Create a function named `doTurn/DoTurn/do_turn` that calls the functions in the proper order as described in the paragraph above.

- `combat`
- `buyHealth`
- `getCoins`
- `printStatus`
- `rollDice`
- `move`

Language: Python

Kata Level: 8

Solution:

```
def do_turn():
    [_() for _ in (roll_dice, move, combat, get_coins, buy_health, print_status)]
```

Grasshopper - Array Mean

Description:

Find Mean

Find the mean (average) of a list of numbers in an array.

Information

To find the mean (average) of a set of numbers add all of the numbers together and divide by the number of values in the list.

For an example list of `1, 3, 5, 7`

1. Add all of the numbers

```
1+3+5+7 = 16
```

2. Divide by the number of values in the list. In this example there are 4 numbers in the list.

```
16/4 = 4
```

3. The mean (or average) of this list is 4

Language: Python

Kata Level: 8

Solution:

```
def find_average(nums):
    return sum(nums) / len(nums) if nums else 0
```

Counting sheep...

Description:

Consider an array/list of sheep where some sheep may be missing from their place. We need a function that counts the number of sheep present in the array (true means present).

For example,

```
[true,  true,  true,  false,
  true,  true,  true,  true ,
  true,  false, true,  false,
  true,  false, false, true ,
  true,  true,  true,  true ,
  false, false, true,  true]
```

The correct answer would be `17`.

Hint: Don't forget to check for bad values like `null`/`undefined`

Language: Python

Kata Level: 8

Solution:

```
def count_sheeps(sheep):
    a = 0
    for temp in sheep:
        if temp == True:
            a += 1
    return a
```

On the Canadian Border (SQL for Beginners #2)

Description:

You are a border guard sitting on the Canadian border. You were given a list of travelers who have arrived at your gate today. You know that American, Mexican, and Canadian citizens don't need visas, so they can just continue their trips. You don't need to check their passports for visas! You only need to check the passports of citizens of all other countries!

Select names, and countries of origin of all the travelers, excluding anyone from Canada, Mexico, or The US.

travelers table schema

- name
- country

NOTE: The United States is written as 'USA' in the table.

NOTE: Your solution should use pure SQL. Ruby is used within the test cases just to validate your answer.

Language: SQL

Kata Level: 8

Solution:

```
SELECT
  *
FROM
  TRAVELERS
WHERE
  COUNTRY NOT IN ('Mexico', 'USA', 'Canada')--Your Code Here
```

Finding Products Matching All Selected Tags

Description:

Imagine you are managing an e-commerce platform. It offers a diverse range of products, each tagged with various attributes to help customers filter and find items that match their preferences. These tags could represent categories, features, styles, or any other relevant attributes.

You want to implement a feature that allows customers to filter products by selecting multiple tags. Specifically, when a customer selects several tags, the platform should display only those products that are associated with all the selected tags. This ensures that the search results precisely match the customer's combined tag preferences.

We have a `product_tags` table:

- `product_id` (int): Unique identifier for each product

- `tag` (varchar): Tag associated with the product

The table may contain duplicate rows where the same product is associated with the same tag multiple times.

For our task, we want to find products that are tagged with both `Electronics` and `Gadgets`. The query should return `product_id` values in desc order for products that are associated with both of these tags.

for this sample data:

```
| product_id | tag         |
+------------+-------------+
|    101     | Electronics |
|    101     | Gadgets     |
|    102     | Home        |
|    103     | Electronics |
|    103     | Accessories |
|    104     | Kitchen     |
|    105     | Electronics |
|    105     | Gadgets     |
|    105     | Accessories |
|    106     | Gadgets     |
|    106     | Accessories |
```

the expected result is the following:

```
| product_id |
+------------+
|    105     |

|    101     |
```

Language: SQL

Kata Level: 7

Solution:

```
SELECT product_id FROM product_tags WHERE tag IN ('Electronics')
INTERSECT
SELECT product_id from product_tags WHERE tag IN ('Gadgets')
ORDER BY product_id DESC
```

Tip Calculator

Description:

Complete the function, which calculates how much you need to tip based on the total amount of the bill and the service.

You need to consider the following ratings:

- Terrible: tip 0%
- Poor: tip 5%
- Good: tip 10%
- Great: tip 15%
- Excellent: tip 20%

The rating is case insensitive (so "great" = "GREAT"). If an unrecognised rating is received, then you need to return:

- `"Rating not recognised"` in Javascript, Python and Ruby...
- ...or `null` in Java
- ...or `-1` in C#

Because you're a nice person, you always round up the tip, regardless of the service.

Language: Python

Kata Level: 8

Solution:

```
import math

def calculate_tip(amount, rating):
    rating = rating.lower()
    if rating == "terrible":
        return math.ceil(amount * 0)
    elif rating == "poor":
        return math.ceil(amount * 0.05)
    elif rating == "good":
        return math.ceil(amount * 0.1)
    elif rating == "great":
        return math.ceil(amount * 0.15)
    elif rating == "excellent":
        return math.ceil(amount * 0.2)
    else:
        return "Rating not recognised"
```