**JS Task tow**

21-March-2022

monday

Note: Create new folder “JS-21-march-2022” contains html page in your JavaScript repo.

/\*

1

Write a function named tellFortune that:

takes 4 arguments: number of children,

partner's name, geographic location, job title.

outputs your fortune to the screen like so:

"You will be a X in Y, and married to Z with N kids."

Ex: tellFortune('software engineer', 'Jordan', 'Alice', 3);

=> "You will be a software engineer in Jordan, and married to Alice with 3 kids."

\*/

/\*

2

Write a function named calculateDogAge that:

takes 1 argument: your puppy's age.

calculates your dog's age based on the conversion

rate of 1 human year to 7 dog years.

outputs the result to the screen like so:

"Your doggie is NN years old in dog years!"

Ex: calculateDogAge(1);

=> "Your doggie is 7 years old in dog years!"

\*/

/\*

3

Write a function named calculateSupply that:

takes 2 arguments: age, amount per day.

calculates the amount consumed for rest of the life (based on a constant max age 100).

outputs the result to the screen like so:

"You will need NN to last you until the ripe old age of X"

Ex: calculateSupply(30, 3);

=> 'You will need 76650 cups of tea to last you until the ripe old age of 100;

\*/

/\*

4

Write a function called greet that:

takes 1 argument: name.

and it will return hello + name

Ex: greet("Adam")

=> "Hello Adam"

\*/

/\*

5

what is the error:

function double(cat) {

return 2 \* x;

}

function double(7) {

return 2 \* 7;

}

function double('7') {

return 2 \* 'x';

}

\*/

/\*

6

fix these functions:

func double1(x {

return 2 \* x ;

}

functiondouble2 x)

return 2 \* x;

}

function (x) double3 {

return 2 \* x;

\*/

/\*

7

Write a function called cube that:

accept 1 parameter and calculate the cube of this number

Ex: cube(4)

=> 64

\*/

/\*

8

Write a function called multiply that:

accept 2 parameters and calculate the multiply of these 2 numbers

Ex: multiply(3,4)

=> 12

Ex: multiply(5,4)

=> 20

\*/

/\*

9

Write a function called canIGetADrivingLicense that:

accept 1 parameter represent the age

and if the age greater than or equal to 20 return "yes you can"

otherwise return "please come back after X years to get one"

Ex: canIGetADrivingLicense(21)

=> "yes you can"

Ex: canIGetADrivingLicense(17)

=> "please come back after 3 years to get one"

Ex: canIGetADrivingLicense(20)

=> "yes you can"

\*/

/\*

10

Write a function called sameLength

that accepts two strings as arguments,

and returns true if those strings have the same length, and false otherwise.

\*\*hint: how we can know string length Ex: : "tree".length => 4

Ex: sameLength("tree","clue")

=> true

Ex: sameLength("tree","car")

=> false

\*/

/\*

11

Write a function called largerNubmer

that accept two numbers as arguments,

and return the first larger numbers

Ex: largerNubmer(5,6)

=> 6

Ex: largerNubmer(5,3)

=> 5

\*/

/\*

12

Write a function called smallerNubmer

that accept three numbers as arguments,

and return the first smaller number

Ex: smallerNubmer(8,6,7)

=> 6

Ex: smallerNubmer(5,99,34)

=> 5

Ex: smallerNubmer(5,99,3)

=> 3

Ex: smallerNubmer(5,3,3)

=> 3

\*/

/\*

13

Write a function called shorterString

that accept five string as an arguments,

and return the first shorter string

Ex: shorterString("air","school","car","by","github")

=> by

Ex: shorterString("air","tr","car","by","github")

=> tr

Ex: shorterString("by","tr","car","air","github")

=> by

Ex: shorterString("air","by","car","school","github")

=> by

Ex: shorterString("air","tr","by","car","github")

=> by

Ex: shorterString("air","tr","car","github","by")

=> by

\*/

/\*

14

Write a function called longerString

that accept four string as an arguments,

and return the first longer string

Ex: longerString("air","school","car","github")

=> school

Ex: longerString("air","schoo","car","github")

=> github

try all the cases (change the order of the longestString)

\*/

/\*

15

Write a function called isEven

that accept 1 argument as a number,

and return true if this number is even

and false if this number is odd

Ex: isEven(1)

=> false

Ex: isEven(2)

=> true

\*/

/\*

16

Write a function called isOdd

that accept 1 argument as a number,

and return true if this number is Odd

and false if this number is Even

Ex: isOdd(4)

=> false

Ex: isOdd(5)

=> true

\*/

/\*

17

Write a function called positive

that accept 1 argument as a number,

and return the positive version of the number passed

Ex: positive(4)

=> 4

Ex: positive(-5)

=> 5

\*/

/\*

18

Write a function called fullName

that accept two parameters, firstName and lastName,

and returns the firstName and lastName concatenated

together with a space in between.

Ex: fullName("Adam","McCallen")

=> "Adam McCallen"

Ex: fullName("Alex", "Mercer")

=> "Alex Mercer"

\*/

/\*

19

Write a function called average

that takes five numbers as inputs

and returns the average of those numbers.

Ex: average(1,2,3,4,5)

=> 3

Ex: average(5,7,9,3,5)

=> 5.8

\*/

/\*

20

Write a function called randomNumber

that didnt takes any parameter

and returns a random number between 0-1

\*\* hint: you can seacrh using MDN

Ex: randomNumber()

=> 0.2278

Ex: randomNumber()

=> 0.475

\*/

/\*

21

Write a function called randomBetweenNumbers

that takes 2 parameters

and returns a random number between them

\*\* hint: you can seacrh using MDN

Ex: randomBetweenNumbers(1,8)

=> 7.5412

Ex: randomBetweenNumbers(3,100)

=> 23

\*/

/\*

22

Write a function called scoreInUniversty

that takes 1 parameters

and returns the alpabet in the unevirsty

A => 95-100

B => 85-94

C => 70-84

D=> 50-69

F=> 0-49

Ex: scoreInUniversty(96)

=> "A"

Ex: scoreInUniversty(3)

=> "F"

Ex: scoreInUniversty(71)

=> "C"

\*/

/\*

23

Write a function called counter

that will returns a number bigger

than the one that returnd before

and start from 0

Ex: counter()

=> 1

Ex: counter()

=> 2

Ex: counter()

=> 3

\*/

/\*

24

Write a function called resetCounter

that will reset the previuos function

and return the number before reset and

a string say that the counter reset

Ex: counter()

=> 1

Ex: counter()

=> 2

Ex: counter()

=> 3

Ex: resetCounter()

=> 3 and the counter reset now

Ex: counter()

=> 1

Ex: counter()

=> 2

Ex: resetCounter()

=> 2 and the counter reset now

Ex: counter()

=> 1

\*/