validatePhoneNumber

validatePhoneNumber ← string

Decision: $(((string.indexOf("-") === 3) \&\& ((string.lastIndexOf("-")) === 7)) \&\& (string.length === 12))$

- True / False → "validatePhoneNumber" : validatePhoneNumber

validateEmail

validateEmail ← string

isWhiteSpace = /\s+/g

Decision: $(((string.indexOf("@") !== -1) \&\& (isWhiteSpace.test(string) === false) \&\& ((string.lastIndexOf(".")) === (string.length - 4))))$

- True / False → "validateEmail": validateEmail

validateURL

validateURL ← string

isHttp = /^http:\/\//
isHttps = /^https:\/\//
isWhiteSpace = /\s+/g

Decision: $((((isHttp.test(string)) || (isHttps.test(string))) \&\& (isWhiteSpace.test(string) === false))$

- True / False → "validateURL": validateURL

titleCaseString

titleCaseString ← string

Local Variables

For Loop

→ "titleCaseString": titleCaseString

changeSeparator

changeSeparator ← string, string

replaceThis = /,/g
replaceSeparator = string.replace (replaceThis, newSeparator)

→ "changeSeparator": changeSeparator

formatDecimalPlaces

formatDecimalPlaces ← string, decimalPlaces

alterDecimalPlaces = number.toFixed (decimalPlaces)

→ "formatDecimalPlaces" : formatDecimalPlaces

fuzzyMatch

firstNumber
secondNumber
percentageToTest

→ **fuzzyMatch**

```
difference = firstNumber – secondNumber
percentDifference = difference / secondNumber
percentage = percentDifference * 100
```

((percentage === percentageToTest) ||
(percentage === -(percentageToTest)))

True
False

"fuzzyMatch":
fuzzyMatch

daysDifference

date2
date1

→ **daysDifference**

```
futureDate = new Date(date2)
mostRecentDate = new Date(date1)
dayDifference = (futureDate.getTime() - mostRecentDate.getTime())
seconds = Math.floor(dayDifference / 1000)
minutes = Math.floor(seconds / 60)
hours = Math.floor(minutes / 60)
days = Math.floor(hours / 24)
```

"daysDifference":
daysDifference

actualValue

stringNumber

→ **actualValue**

```
numberValue =
Number(stringNumber)
```

"actualValue":
actualValue

smallestValueGreater

array
number

→ **smallestValueGreater**

```
numbersFound = []
greaterThanGivenNumber
i
```

```
for (i = 0; i < array.length; i += 1) {
    if (array[i] > number) {
greaterThanGivenNumber = (array[i]);
numbersFound.push(array[i]);
    numbersFound.sort()
```

"smallestValueGreater"
: smallestValueGreater

totalValueOfArray

array

→ **totalValueOfArray**

```
totalValue = 0
grabNumbers
i
```

```
for (i = 0; i < array.length; i += 1) {
    if ((array[i][0] === "0") ||
    (array[i][0] === "1") ||
    (array[i][0] === "2") ||
    (array[i][0] === "3") ||
    (array[i][0] === "4") ||
    (array[i][0] === "5") ||
    (array[i][0] === "6") ||
    (array[i][0] === "7") ||
    (array[i][0] === "8") ||
    (array[i][0] === "9")) {
        array.splice(i, 1);
        }
grabNumbers = parseFloat(array[i]);
    if (!isNaN(grabNumbers)) {
totalValue = totalValue + grabNumbers
```

"totalValueOfArray":
totalValueOfArray

sortObjectsByKey

array
key

→ **sortObjectsByKey**

```
sortedArray
i
```

For Loop

"sortObjectsByKey":
sortObjectsByKey

Denise M. Gilbert
October 18, 2012
Term 1210
Project 4
Page 2 of 2