stoge Documentation

Release latest

Jun 25, 2020

Contents

The following are examples of array methods.

CHAPTER 1

stoge.isArray(argument)

`isArray` method tells if a given input is an array or not. If there is an array then it retuns true otherwise false.

```
stoge.isArray([1,2,3,4])
// => true

stoge.isArray('string')
// => false
```

CHAPTER 2

stoge.removeFalsy(array)

`removeFalsy` method filter all false value exist in an array and retuns array of non false value.

```
stoge.removeFalsy([1,'',2,false,3]) => [1,2,3]
// => true
```

CHAPTER 3

stoge.shuffleArr(array)

`shuffleArr` method shuffle all values of given array.

```
stoge.shuffleArr([1,2,3,4])
// => [4,1,3,2]

stoge.shuffleArr(['a','b','c','d'])
// => ['c','b','d','a']
```

CHAPTER 4

stoge.binarySearch(array, searchValue)

`binarySearch` method retuns the index of given value to be searched if value is available in array else retun `-1`

```
stoge.binarySearch([1,5,6,3,9],9)
// => 4

stoge.binarySearch(['a','b','c','d'],'c')
// => 2

stoge.binarySearch(['a','b','c','d'],'z')
// => -1
```

CHAPTER 5

stoge.arraySum(array, array)

`arraySum` method retuns a single array that is a combination of 2 given arrays.

```
stoge.arraySum([1,2,3,4],[5,6,7,8])
// => [6,8,10,12]

stoge.arraySum(['a','b','c','d'])
// => ['a','b','c','d']

stoge.arraySum([1,2,3,4],[5,6,7,8,'a','c',9])
// => [6,8,10,12,'a','c',9]
```

CHAPTER 6

stoge.chunk(array, number)

`chunk` method break given array by given chunk number from second parameter.

```
stoge.chunk([1,2,3],2)
// => [[1,2],[3]]

stoge.chunk(['a','b','c','d','e'],3)
// => [['a','b','c'],['d','e']]
```

CHAPTER 7

stoge.flatarr(array)

`flatarr` method returns given nested array into single array.

```
stoge.flatarr([10, [90, 80], [5, [1, [69, [61, 15]]], 19]])
// => [10, 90, 80, 5, 1, 69, 61, 15, 19]

stoge.flatarr(['a',['b',['c']],['d','e']])
// => ["a", "b", "c", "d", "e"]
```

CHAPTER 8

stoge.distinct(array)

`distinct` method diffrentiate given array value by their type and retun an object of type with their respective value.

```
stoge.distinct([1,2,'f',3,'c'])
// => {num: [1, 2, 3], str: ["f", "c"]}
```

CHAPTER 9

stoge.rotate(array, number)

`rotate` method perform left rotation of given array upto given position in number.

```
stoge.rotate([1,2,3,4,5],3)
// => [3,4,5,1,2]
```

CHAPTER 10

Math Methods

The following are examples of Math methods.

10.1 stoge.average(argument)

`average` method retuns the average of a given array of numbers.

```
stoge.average([1,2,3,4])
// => 2.5

stoge.average([12,22,83])
// => 39
```

10.2 stoge.randomHexColorCode()

`randomHexColorCode` method retuns a random generated hex code of color.

```
stoge.randomHexColorCode()
// => #99ff55

randomHexColorCode()
// => #FFddcc
```

10.3 stoge.baseConvert(num, init_base, change_base)

`baseConvert` method convert given string to any given base value.

```
stoge.baseConvert('E164',16,8)
// => 160544
```

10.4 stoge.binToDec(number)

`binToDec` method convert binary value to decimal.

```
stoge.binToDec(110011)
// => 51
```

10.5 stoge.revNum(number)

`revNum` method reverse the given number with their sign.

```
stoge.revNum(-1345)
// => -5431
```

10.6 stoge.decToBHO(number)

`decToBHO` method convert given argument into their respective give base value of Binary, Hex or Octal and second argument would be like for binary 'B', hex 'H', octal 'O'.

```
stoge.decToBHO(5)
// => 101
```

CHAPTER 11

Number Methods

The following are examples of Number methods.

11.1 stoge.range(value, lowerRange, upperRange)

`range` method tells if a number is in given range or not.

```
stoge.range(9,3,10)
// => true

stoge.range(5,3)
// => false
```

11.2 stoge.random(startingValue, endValue)

`random` method retuns a random generated number by given startingValue and endValue as input.

```
stoge.random(1,5)
// => 3

stoge.random(6)
// => 2
```

11.3 stoge.fToCelsius(number, number)

`fToCelsius` method convert given `Fahrenheit` number to `Celsius`.

```
stoge.fToCelsius(140)
// => 60

stoge.fToCelsius(120,2)
// => 48.89
```

11.4 stoge.cToFahrenheit(number, number)

`cToFahrenheit` method convert given `Celsius` number to `Fahrenheit`.

```
stoge.cToFahrenheit(60)
// => 140

stoge.cToFahrenheit(48,2)
// => 118.4
```

11.5 stoge.arrayRange(number, number)

`arrayRange` method retuns an array by given 2 number parameter where first parameter is starting value and second parameter is length of array.

```
stoge.arrayRange(9,3)
// => [9,10,11]

stoge.arrayRange(5)
// => [1,2,3,4,5]
```

11.6 stoge.gcd(number, number)

`gcd` method retuns Greatest Common Division of 2 given number as parameter.

```
stoge.gcd(9,3)
// => 3

stoge.gcd(10,5)
// => 5
```

11.7 stoge.lanczosGamma(number)

`lanczosGamma` method is used for computing the gamma function numerically.

```
stoge.lanczosGamma(6)
// => 120.00000000000021

stoge.lanczosGamma(1)
// => 0.9999999999999998
```

11.8 stoge.fallFactorial(number, number)

`fallFactorial` method takes to arguments and retuns fallFactorial.

```
stoge.fallFactorial(6,2)
// => 30

stoge.fallFactorial(1)
// => 0.9999999999999998
```

11.9 stoge.isHamming(number)

`isHamming` method tells if number is hamming or not.

```
stoge.isHamming(7)
// => false

stoge.isHamming(6)
// => true
```

11.10 stoge.hamming(number)

`hamming` method returns array of hamming number upto given length

```
stoge.hamming(7)
// => [1,2,3,4,5,6,8]
```

CHAPTER 12

String Methods

The following are examples of String methods.

12.1 stoge.isMail(string, pattern)

`isMail` method validate if given string is in format the format of mail or not an retuns result in boolean. And by second parameter you can paas youre own Regex pattern to validate the same.

```
stoge.isMail('abc@example.com')
// => true

stoge.isMail('abc@example@.com')
// => false

stoge.isMail('abc@example.com',/\S+@\S+\.\S+/g)
// => true
```

12.2 stoge.distinct(string)

`distinct` method returns string with whitespace between every character.

```
stoge.distinct('lower')
// => 'l o w e r'
```

12.3 stoge.mediochar(string)

`mediochar` method filter given string and returns string except all special character.

```
stoge.mediochar('xyz@$#(),;!$%^&*+-_.+=abc')
// => 'xyzabc'
```

12.4 stoge.specialChar(string)

`specialChar` method filter given string and returns only special characters that exist in string.

```
stoge.specialChar('xyz@$#(),;!$%^&*+-_.+=abc')
// => '@$#(),;!$%^&*+-_.=+'
```

12.5 stoge.nonASCII(string)

`nonASCII` method removes all character that does not come under ASCII.

```
stoge.nonASCII('äÄçÇéÉêabcdöÖÐþúÚ')
// => 'abc'
```

12.6 stoge.capitalize(string)

`capitalize` method retuns string in capitalize form.

```
stoge.capitalize('BLUNT')
// => 'Blunt'
```

12.7 stoge.dCapitalize(string)

`capitalize` method retuns string in decapitalize form.

```
stoge.capitalize('Blunt')
// => 'bLUNT'
```

12.8 stoge.camel(string)

`camel` method retuns string in camel case form.

```
stoge.camel('abc def ghi')
// => 'abcDefGhi'
```

12.9 stoge.snake(string)

`snake` method retuns string in snake case form.

```
stoge.snake('abc def ghi')
// => 'abc_def_ghi'
```

12.10 stoge.kebab(string)

`kebab` method retuns string in kebab case form.

```
stoge.kebab('abc def ghi')
// => 'abc-def-ghi'
```

12.11 stoge.pascal(string)

`pascal` method retuns string in pascal case form.

```
stoge.pascal('abc def ghi')
// => 'AbcDefGhi'
```

12.12 stoge.escape(string)

`escape` method convert escape character into their HTML entity code.

```
stoge.escape('abc&><def')
// => 'abc&amp;&gt;&lt;def'
```

12.13 stoge.unescape(string)

`unescape` method convert HTML entity code into their unescape character.

```
stoge.unescape('abc&amp;&gt;&lt;def')
// => 'abc&><def'
```

12.14 stoge.partialReverse(string)

`partialReverse` method partialy reverse given sentence without reversing every word.

```
stoge.partialReverse('abc def GHI')
// => 'GHI def abc'
```

12.15 stoge.completeReverse(string)

`completeReverse` method completely reverse given sentence even by every word of sentence.

```
stoge.completeReverse('abc def GHI')
// => 'IHG fed cba'
```

12.16 stoge.csvToArray(string, delimiter, omitHeader)

`csvToArray` method convert given csv format into array and can pass custom delimiter with omit header in boolean.

```
stoge.csvToArray('a,b\nc,d')
// => '[[a,b], [c,d]]'
```

12.17 stoge.csvToJson(string, delimeter)

`csvToJson` method convert given csv format string into JSON format and can pass custom delimiter.

```
stoge.csvToJson('a,b\nc,d')
// => '{a: 'c', b: 'd'}'

stoge.csvToJson('a,b\nc,d')
// => '{a: 'c', b: 'd'}'
```

12.18 stoge.swapCase(string)

`swapCase` method retun string with opposite case of given string.

```
stoge.swapCase('ADcfRD1231')
// => 'adCFrd1231'
```

12.19 stoge.subStr(string)

`subStr` method retun all sub string of given string.

```
stoge.subStr('stog')
// => '["s", "st", "sto", "stog", "t", "to", "tog", "o", "og", "g"]'
```

12.20 stoge.charCount(string)

`charCount` method count each character available in string and returns occurances.

```
stoge.charCount('hello emily')
// => '{h: 1, e: 2, l: 3, o: 1, " ": 1, ...}'
```

12.21 stoge.maxChar(string)

`maxChar` method returns the object of the maximum count of character.

```
stoge.maxChar('hello emily')
// => '{char: "l", count: 3}'
```

12.22 stoge.cleanStr(string)

`cleanStr` method returns raw characters in alphabetical order of any given string.

```
stoge.cleanStr('Hello There')
// => 'eeehhllort'
```

12.23 stoge.anagram(aStr, bStr)

`anagram` method returns boolean value true if both given string's characters are matched else returns false.

```
stoge.anagram('rail safety','fairy tales')
// => 'true'
```

12.24 stoge.vowels(str)

`vowels` method returns count of total vowels available in the given string.

```
stoge.vowels('hello')
// => '2'
```