1. Write a js method, when user click on button, display random integer below it.

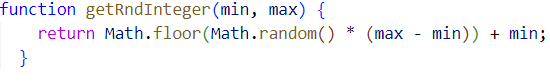
Code:

HTML:

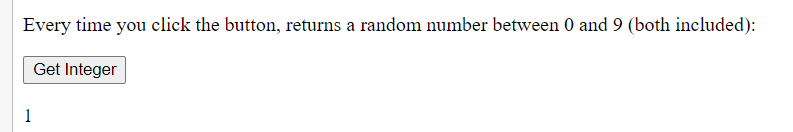
Text

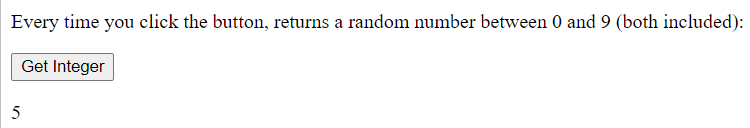
Description automatically generated with low confidence

JS:



Result:





2. Write js function to format number up to specified decimal places.

Code:

Text

Description automatically generated with medium confidence

Result:



1. Check particular sub-word exist in a string or not e.g. i am learning js: 'js' exists or not.

Code:

Text

Description automatically generated

Result:



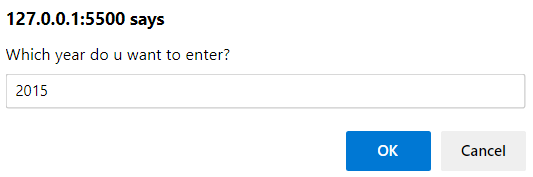
1. Given year leap year or not (29 in feb)

Code:

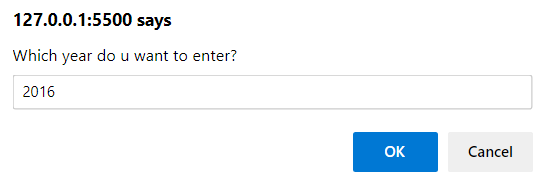
Graphical user interface, text

Description automatically generated with medium confidence

Result:









5. ATM Machine: balance & query, withdraw (amount), change pin, mini statement, saving & current acc., print receipt, enter pin number

Code:

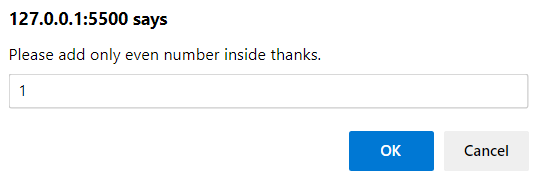
1. Add only even numbers in an array (array elements to be input by user)

Code:

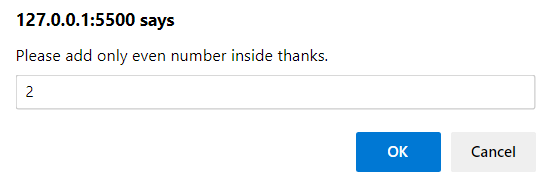
Text

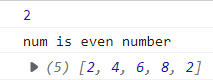
Description automatically generated

Result:









1. Found an element in array [10, 78, 90] return number, exit from an array// take user input.

Code:

Text

Description automatically generated

Result:



1. Biggest of even number in an array ([10, 12, 90, 93, 707]): biggest even number is 90

Code:

Graphical user interface, text, application

Description automatically generated

Result:



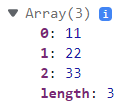
1. Add two array [10,20,30] + [1,2,3]: [11, 22, 33]

Code:

Graphical user interface, text, application

Description automatically generated

Result:



1. Reverse an array(with loops) [10, 78, 0]: [0, 78, 10]

Code:

Graphical user interface, text, application

Description automatically generated

Result:



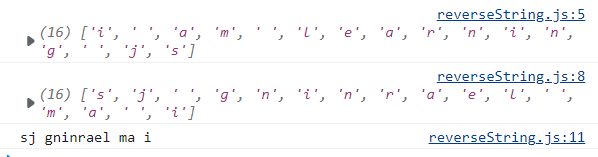
1. Reverse a string using loops

Code:

Graphical user interface, text, application

Description automatically generated

Result:



1. Remove duplicate items from an array [10, 50,20 67, 10, 20]: remove 10, 20

Code:

A picture containing graphical user interface

Description automatically generated

Result:



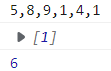
1. Find duplicate values in an array.(display index of duplicate values e.g. 0,2,4,5)

Code:

Graphical user interface, text, application

Description automatically generated

Result:



1. Find difference/subtraction in 2 arrays //[12, 56, 789] - [12, 56, 789]: [0, 0, 0]

Code:

Graphical user interface, text

Description automatically generated

Result:



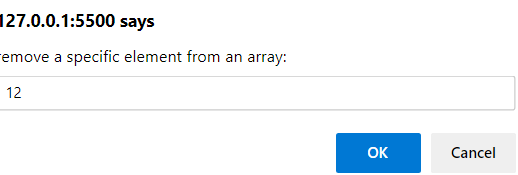
1. Ask user, remove a specific element from an array [12, 56, 89]: remove 1 element

Code:

Graphical user interface, application, Word

Description automatically generated

Result:





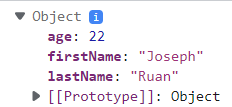
1. Take 3 inputs from user and structure them into objects.

Code:

Graphical user interface, text, application

Description automatically generated

Results:



18. Create a class Car: city(),specialFeature()

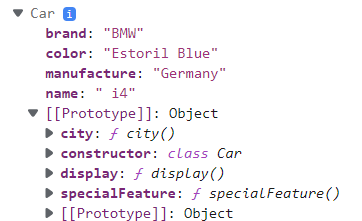
name, brand, color, manufacture

code:

Graphical user interface, text, application, email

Description automatically generated

Result:



19. Create a class Book: type\_of\_book()

no. of pages, type of pages, author

code:

Graphical user interface, text, application, email

Description automatically generated

Result:

Text

Description automatically generated

20. Create a class Animal: walk(), eat(), climb()

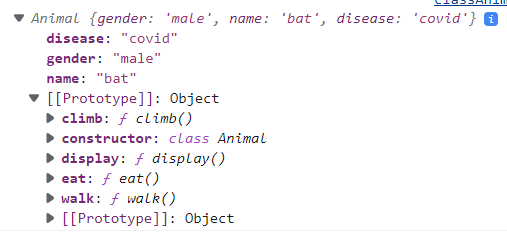
gender, name, disease

code:

Text, application

Description automatically generated

Result:



21. Inheritance: parent electronics (methods: name, version, company name): childclass(laptop, ipad, mobile, tablet):

class child {

configuration()

price()

}

Code

class Electronics {

  constructor(name, version, companyName) {

    this.name = name;

    this.version = version;

    this.companyName = companyName;

  }

}

class Laptop extends Electronics {

  constructor(name, version, companyName, price) {

    super(name, version, companyName);

    this.price = price;

  }

  configuration(size) {

    if (size > 1000) {

      console.log(`This is big.`);

    } else {

      console.log(`This is small.`);

    }

  }

}

class Ipad extends Electronics {

  constructor(name, version, companyName, price) {

    super(name, version, companyName);

    this.price = price;

  }

  configuration(size) {

    if (size > 1000) {

      console.log(`This is big.`);

    } else {

      console.log(`This is small.`);

    }

  }

}

class Mobile extends Electronics {

  constructor(name, version, companyName, price) {

    super(name, version, companyName);

    this.price = price;

  }

  configuration(size) {

    if (size > 500) {

      console.log(`This is big.`);

    } else {

      console.log(`This is small.`);

    }

  }

}

class Tablet extends Electronics {

  constructor(name, version, companyName, price) {

    super(name, version, companyName);

    this.price = price;

  }

  configuration(size) {

    if (size > 1000) {

      console.log(`This is big.`);

    } else {

      console.log(`This is small.`);

    }

  }

}