<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>Example HTML Vue</title>

    <link rel="stylesheet" href="css/main.css" />

    <link rel="stylesheet" href="css/style.css" />

  </head>

  <body>

    <h1>Example HTML Vue</h1>

    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>

    <script src="js/main.js"></script>

    <!-- cdn\_Vue - неге астында тұрады, өйткені егер жоғары қойса ол зогруска жасап қалған кезде экраннан ештеңе көрінбей тұрады, астына қойса экранан барлығы көрініп тек cdn\_Vue- ғана зогускаын күтіп тұрады-->

    <!-- main.js - неге астында   id="app" іздеген кезде cdn\_Vue -ға уже жалғанып қойған болуы керек -->

  </body>

</html>

Array.prototype.**map**() **????**

 arr: [

        { name: "asd1", num: 34 },

        { name: "asd90", num: 12 },

        { name: "asd32", num: 45 },

        { name: "asd23", num: 29 },

        { name: "asd12", num: 78 },

      ],

    this.array = this.arr.map((i,) => i.num);

массив ішіндегі **керектілерін** ғана алып басқа массивке орнату

this.array2 = this.array.map((\_age) => \_age - 12);

массив ішіндегі **элементтерді** жеке-жеке қолдану басқа массивке салу

 this.array = this.arr.map((i, index) => {

      i.num;

      console.log( index);

    });

массив ішіндегі **index** номерлерін анықтау

const array = [1, 2, 3, 4, 5];

function multBy(arr, n) {

  return arr.map(function (i) {

    return i \* n;

  });

}

multBy(array, 5);

.map() бойынша функция беру арқылы тиімді орындауғада болады

Array.prototype.multBy = function (n) {

  return this.map(function (i) {

    return i \* n;

  });

};

array.multBy(5);

былай жасауғада болады глобалный тиімді тәсіл

**console.log -**

**functionName()**

**window**

**window.alert("Hello world")**

**person // обект шақыруға болады**

**person.name // обект ішіндегі элемент шақыруға болады**

**bind(window) // bind деген жақша ішіндегі обектке кіріп барады**

**bind(documrnt) // DOM элемент ішіне кіріп барады**

**bind(object)**

const person = {

  name: "Kuanysh",

  age: 23,

  sayHello: hello,

  sayHelloWindow: hello.bind(window),

  loginfo: function () {

    console.log(`Name is ${this.name}`);

    console.log(`Age is ${this.age}`);

  },

};

const lena = {

  name: "lena",

  age: 22,

};

person.loginfo.bind(lena)(); // бұл жерде bind басқа обьектке кіріп барды

**console.group() :**

console.group(`${this.name} info:`);

    console.log(`Name is ${this.name}`);

    console.log(`Age is ${this.age}`);

    console.groupEnd();

person.loginfo.bind(lena, "Бірінші переменный", "екінші")();

// loginfo функциясына переменный жібермекші болса

person.loginfo.call(lena, "Бірінші переменный", "екінші");

// функцияны шақырмаяқ () қоймаяқ сразу орындау .шін call қолданады

person.loginfo.apply(lena, ["Бірінші переменный", "екінші"]);

// apply - да call сияқты бірақ оның ішінде тек екі элемент болдаы,

сол үшін екіншісі массиф болады

**Apple** ???

**bind, call , apply - бір біріне ұқсас функциялар**

https://www.youtube.com/watch?v=UGapN-hrekw - JS console. map, buld, call ,apply

Event Loop - SetTimout -та 0 сикунт болсада синхронный кезде бәрінен кейін болады, өйткені ол Event Loop-қа кетіп шамалы кідіреді,

button-да сондай шамалы кідіруі Event Loop -та болады, нешерет басса соншарет Event Loop -та қайталаны пшақырады

            onclick="btnclick(4)"

        function btnclick(index) {

          if (a == 1) {

            document.getElementsByClassName("faq\_\_text-toggle")[

              index

            ].style.display = "none";

            a = 0;

          } else {

            document.getElementsByClassName("faq\_\_text-toggle")[

              index

            ].style.display = "inline";

            a = 1;

          }

        }

CallBack -функция ұқсап шақырып жүрмейді сразу орындалады ???