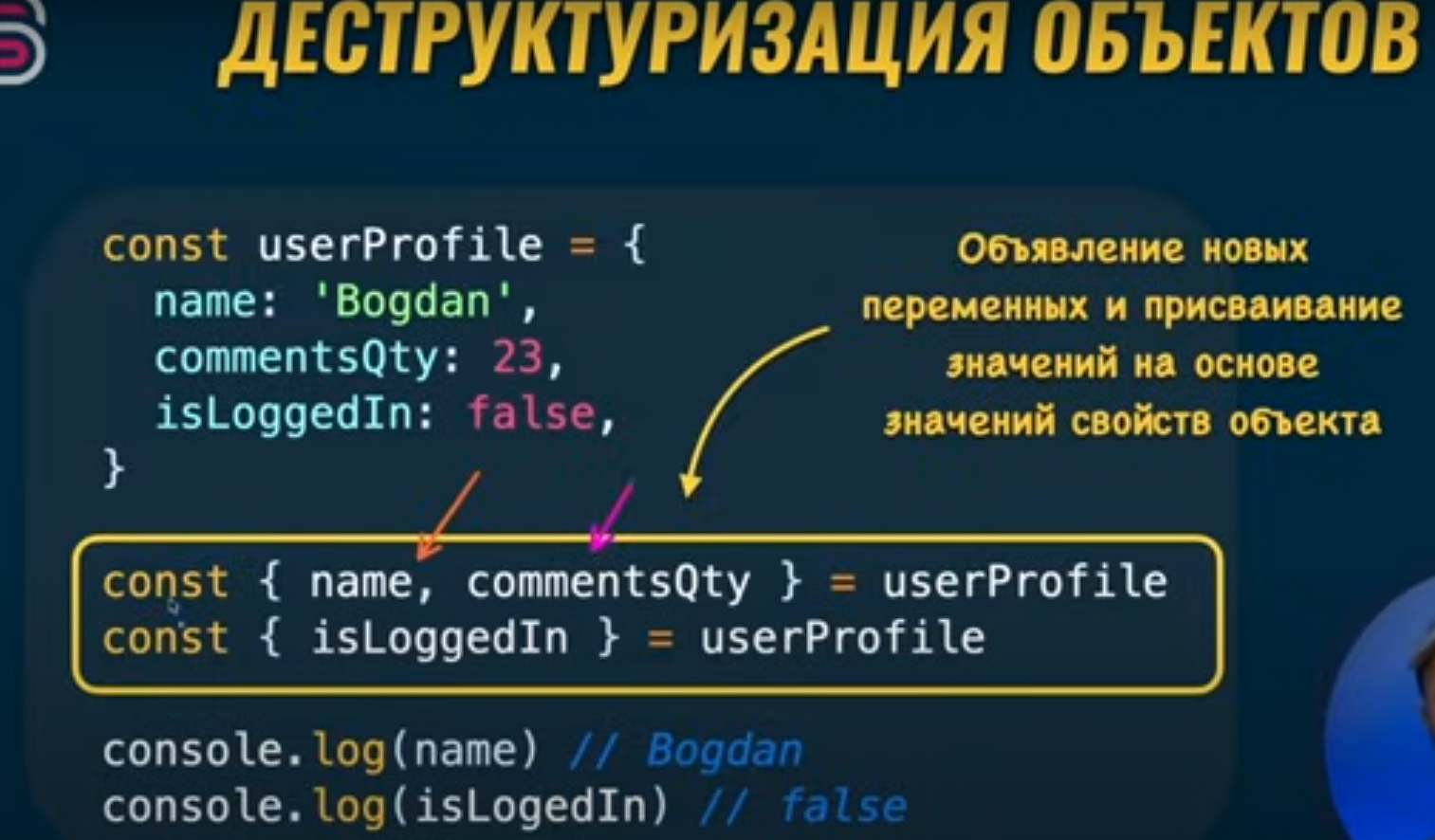
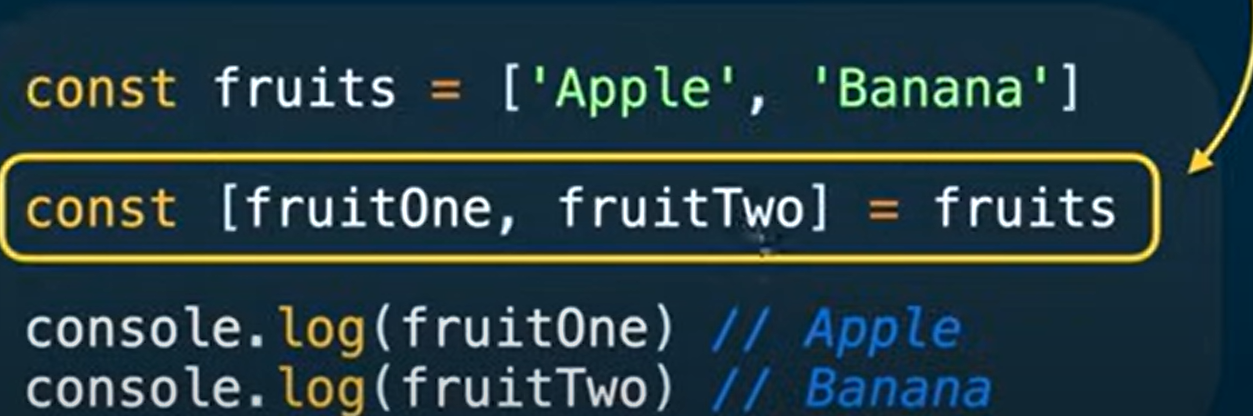
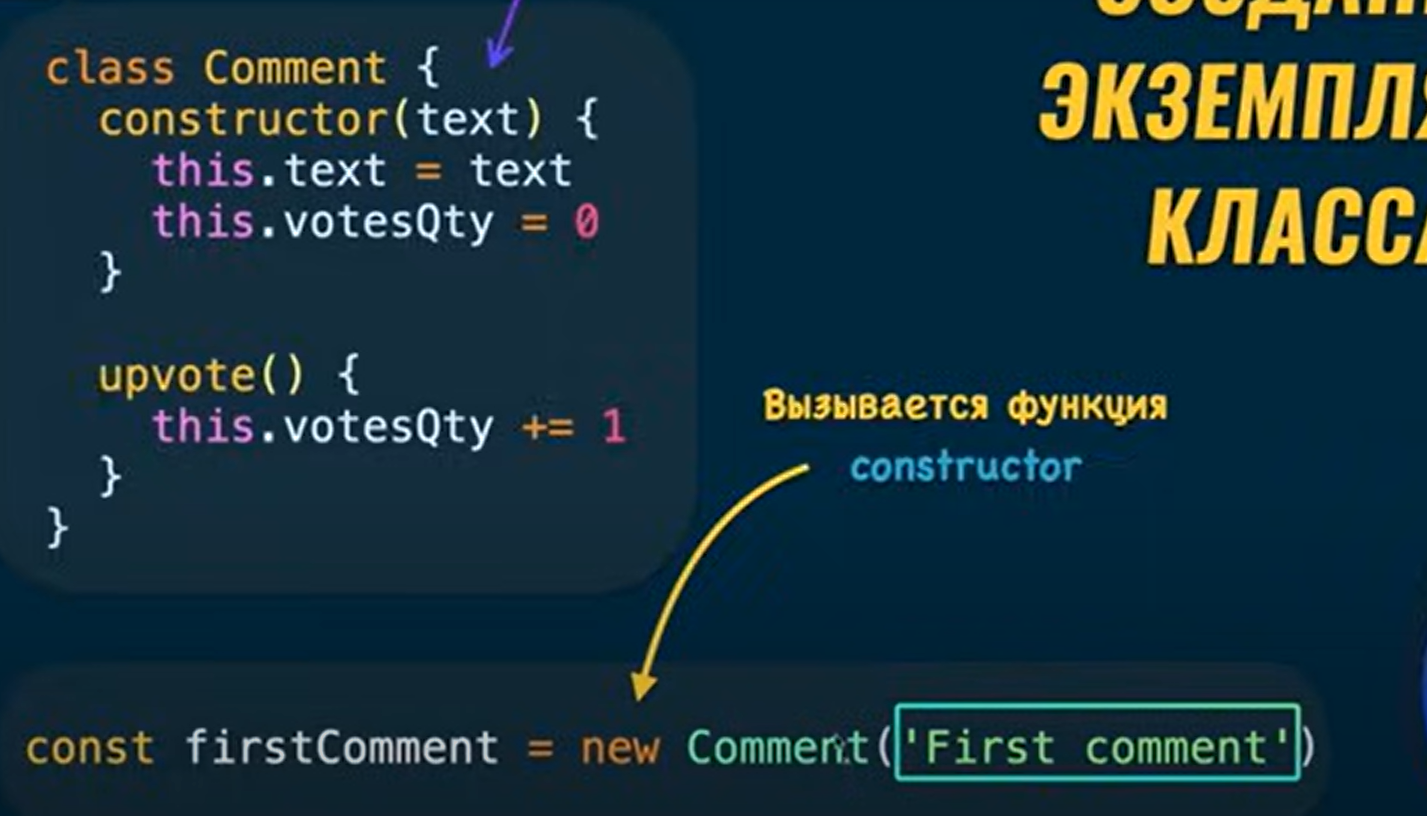
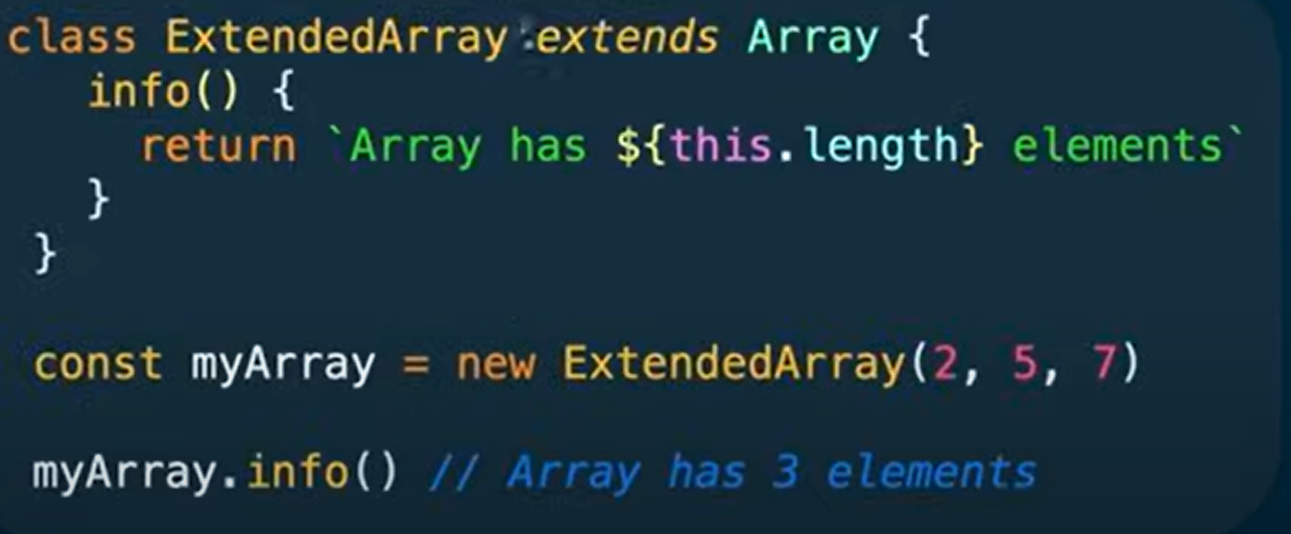
|  |  |
| --- | --- |
| ыSubmitbtn.onclick | При наж на кнопку |
| ?.style.color | Цвет |
| array[**array.length**] | Длинна массива |
| Text-decoration-line-through | Зачеркнуть в js элемент через html /Подчеркнутьгол |
| Data-index =”?” | Если индес вписывать из массива в ф-ию |
| Massi v.**push(?)** | В конец списска |
| myFish.splice(2, 0, "drum"); | Удаляет 0 элементов по индексу 2 и вставляет "drum" |
| Massiv.unshift**(?)** | Первым элементом |
| Shift() | Удаляет первый элемент. Можно в перем |
| Pop() | Удал посл элемент. Можно в перем |
| toreversed() | В обр порядкe не меняя оришинал можно в переменную |
| reversed | Мен оригинал в опбр прядке |
| so | Сортирует  Sort()  В обр порядке |
| Massiv.indexof(Имя) | В каком проядке наход имя |
| Width (?) |  |
|  |  |
|  | Изменить тест массива фор |
|  | Все в верхнем регистре |
|  | Фильтр |
|  | Map+filter+reduce(Сложение) |
| СТРОКА.split(разделитель (“” И т.д)) | MASSIVE ? ДЛЯ ДЕЛЕНИЯ НА НЕСК ЧАСТЕЙ ПО СИМВОЛАМ ЕСЛИ ‘’ ТО КАЖД БУКВА ОТД МАССИВ |
| Массив.join(?) | В строку ? это вконце кажд массива доб символ |
| find | Найти |
|  |  |
|  |  |
|  | Nbmber- перевод в число А Через ДИР можно просотр св-ва обьекта |
|  | MATH |
| .ParseInt(?) ParseFloat(?) BigInt |  |
|  | + это в число перевести |
| .Tofixed(Кол симв после запят) |  |
|  |  |
|  | Random от max ло min |
| **SYRINGS** | |
|  | Использование перем в строках. Если ф-ии то .tostring дописвываем |
| length |  |
| .touppercase |  |
| .statwith | С ЧЕГО НАЧ СТРОЧКА |
| .Repeat(2) |  |
| .trim | Убрать пробелы c конца и начала |
| Ф-ии | |
|  | |
| Set timeout(ф-я (), Через сколько запускать) |  |
| Set interval то же что и сверху только вып код бесконечно |  |
|  | |
| Ложить все в массив numbers что приходит | |
| **Замыкания – Ф-я в кот др ф-я** | |
| / | |
| **Даты** | |
|  |  |
|  |  |
| Now.getfullyear()/month/date/hours/minutes/seconds |  |
| SET ВМЕСТО GET | ЗАДАВАТЬ ВРЕМЯ |
| Now.todatestring() / now.totomestring() / now.tolocaledatestring() |  |
| **Debuging** | |
| Debugger | В код пишем |
| **Объекты** | |
| Можно массивы ф-ии исп |  |
| Стралочн ф-я не созд свой контекст(this=window) |  |
|  | Обращение к обтектам |
| Delete ?.? | Удал обьект св-во |
| **Классы** | |
|  |  |
| **Асинхронность** | |
| За место ф-ии settimeout с передаваемыми операндами след исп Promise | const delay = (time = 1000) => {  const promise = new Promise((resolve, reject) => {  setTimeout(() => {  resolve([1, 2, 3]);  }, time);  });  return promise; }  delay(2500)  .then((data) => {  *console*.log("Timeout:", data);  return data.map((x) => x \*\* 2);  })  .then((data) => *console*.log(data))  .catch((error) => *console*.log(error))  .finally(() => { *console*.log('Finally') }); |
| ВСЕ ОСТ В ФАЙЛЕ VUE JAVASCRIPT 2 |  |
|  |  |
|  |  |
|  |  |
|  |  |

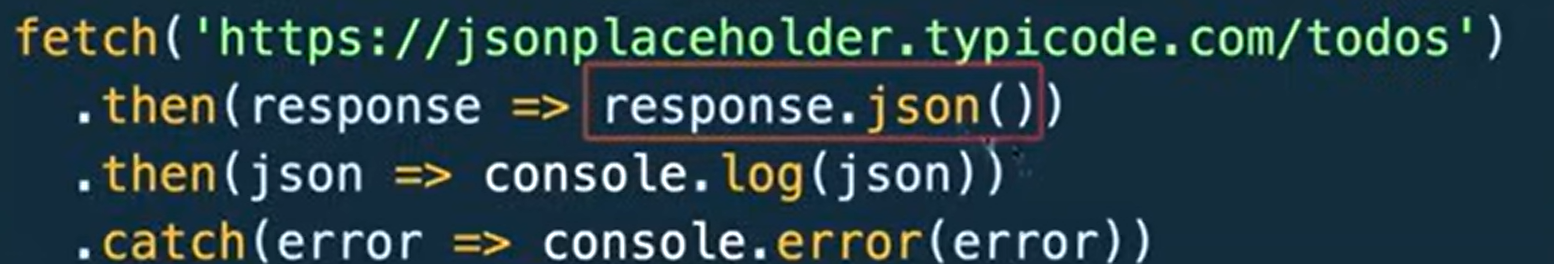


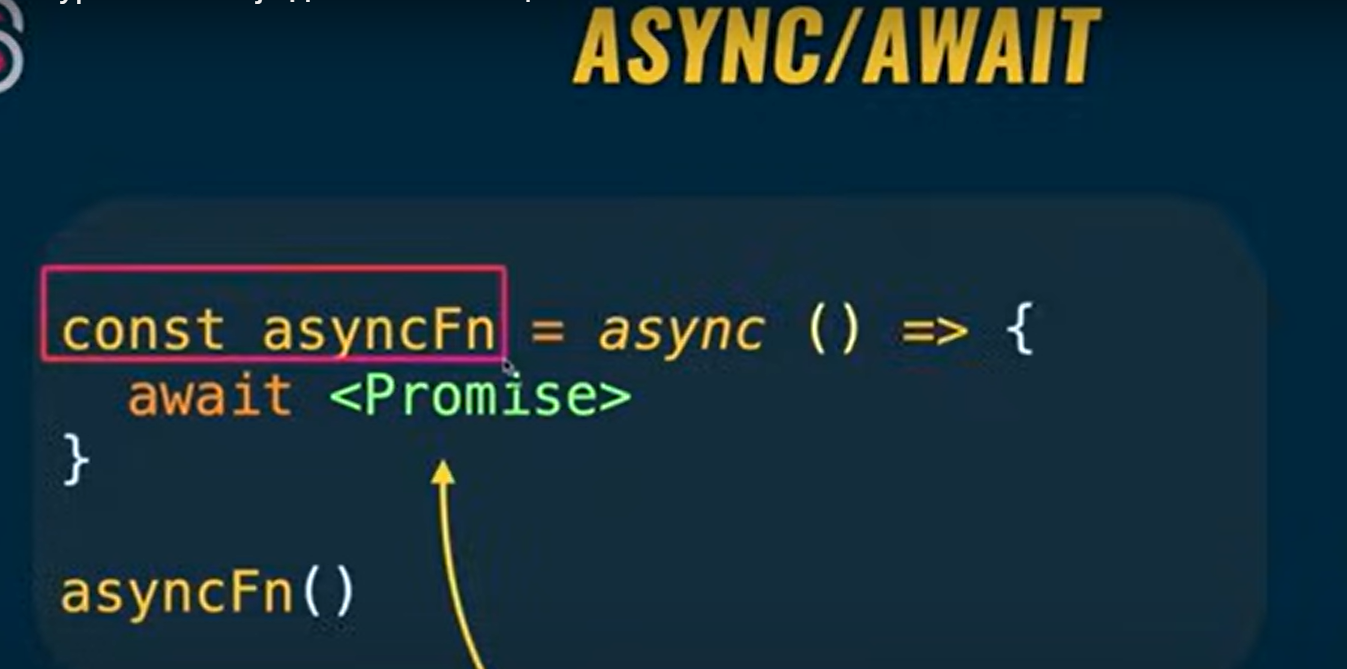


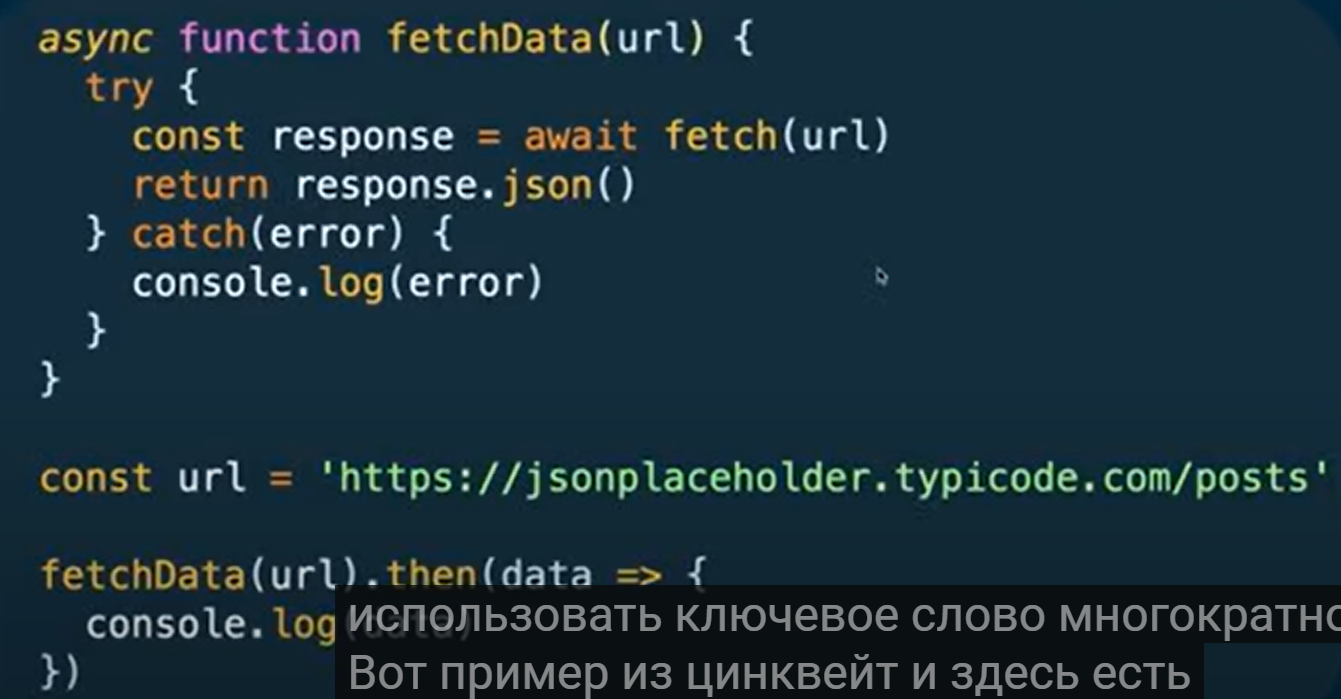




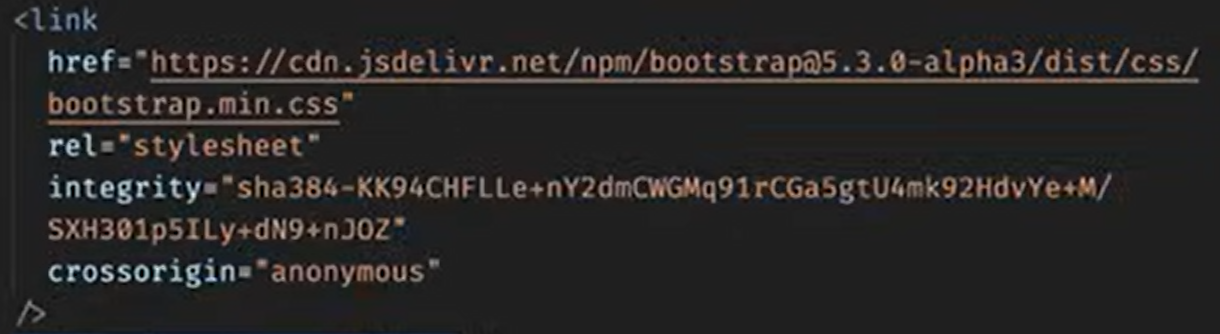
**ПРОМИСЫ**



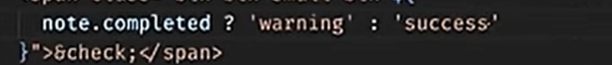




BOOTSTRAP



+ библеотека внутри

Цвет кнопок

Переменная: let num =’hello’;

Константа: const y=65;

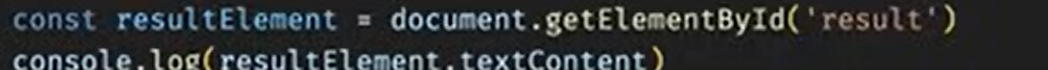
КОММЕНТ // or /\* \*/

Вывод: alert(?) OR console.log(‘Test’, max)

**Обратиться к эл-ту по Id:**

****

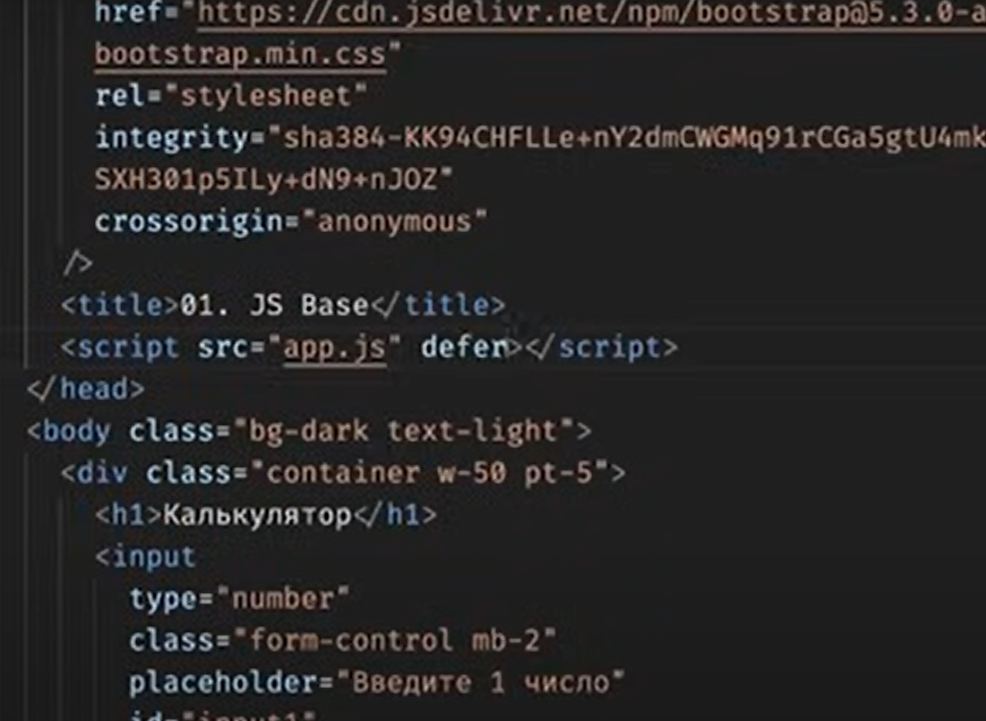
Вывод:



ИЛИ

*text*.innerText

Отраб когда весь контент загружен: defer:



**ПРИВЕДЕНИЕ ТИПОВ:**

Int(?)

Float(?)

**РЕАГИРОВАНИЕ НА КНОПКУ:**

Const submitbtn = document.getelementbyid(‘submit’) //Берем элемент в константу

Submitbtn.onclick = function ()

{

Console. Log ?

}

**Ф-ии**

Function ?( ? ) {

?

}

**HTML в Ф-ИИ КАЖДЫЙ РАЗ Update:**

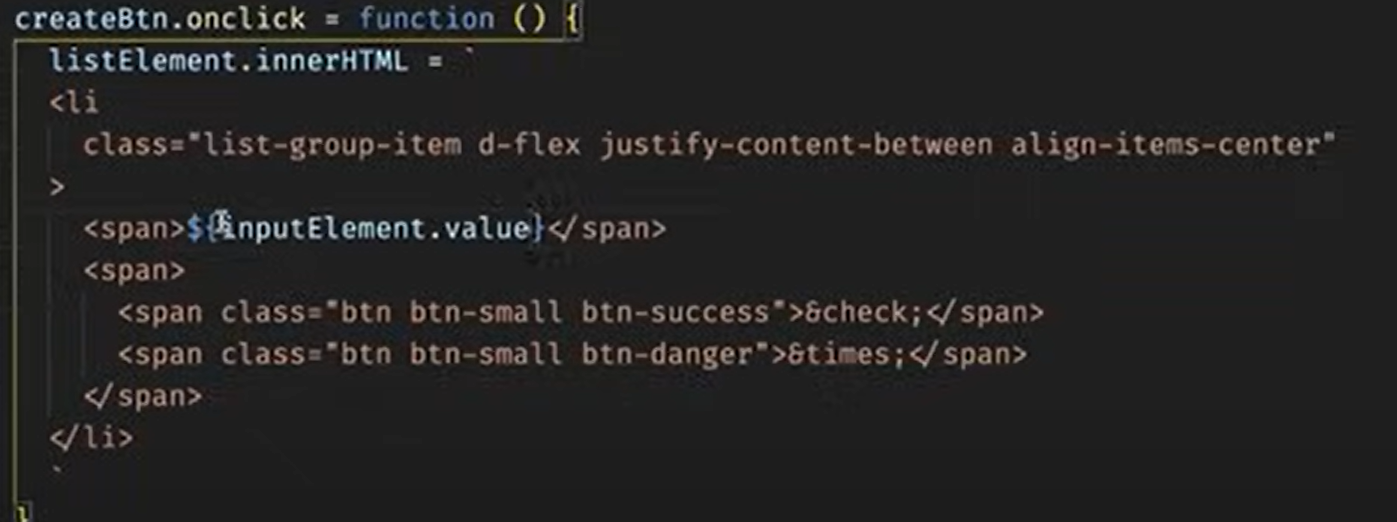
Hel.onclick=function() {

?.innerHtml = ` (обяз эти ковычки)

//Код HTML

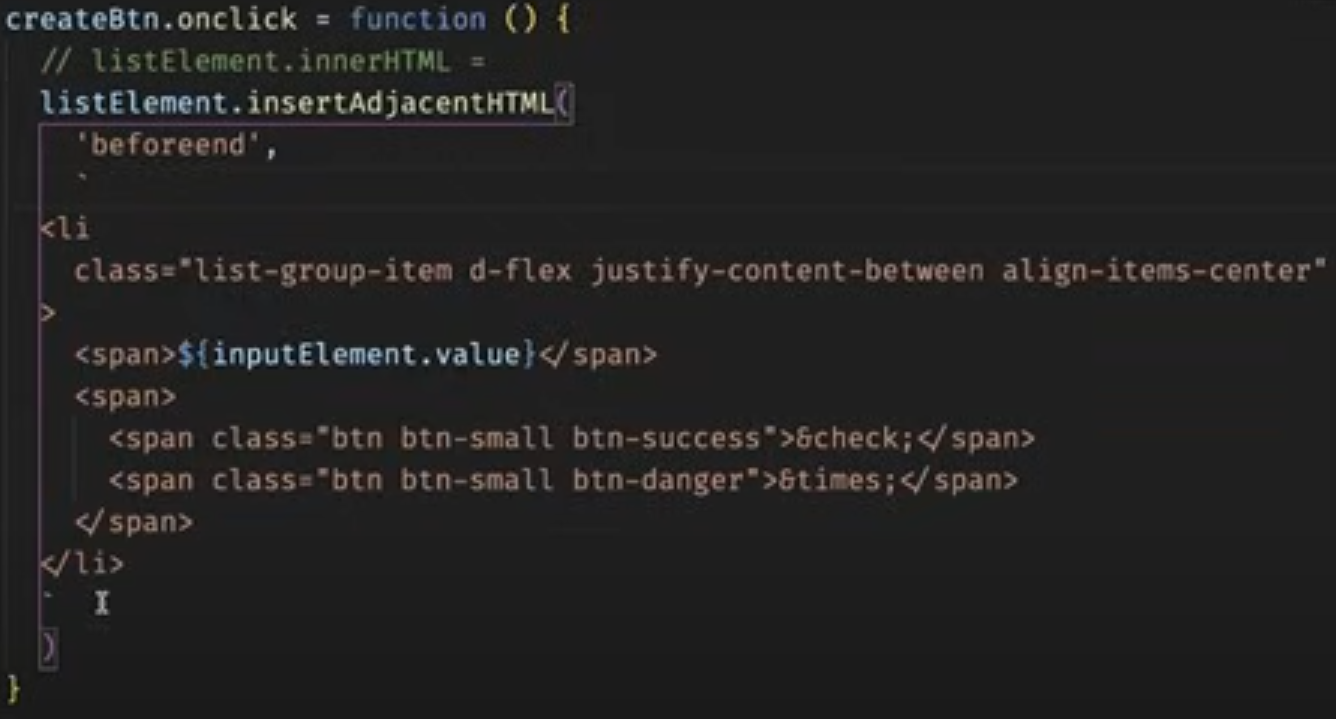
`

}



**HTML в Ф-ИИ ДОБАВЛ К ПРОШЛОМУ:**

Beforeend перед завершением



**МАССИВЫ**

Const array =[1,2,3,4,5…]

const array = new Array(?)

array[0]=’hel’

**ЦИКЛЫ**

****

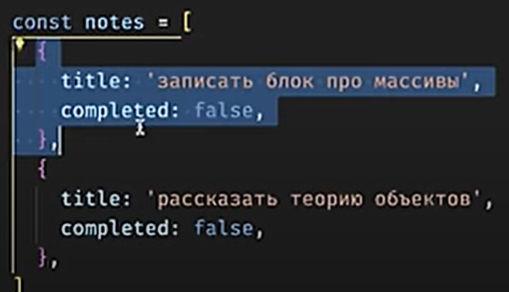
OR



Объекты

Массив из обьектов



Дилигирование событий:



MDN – докум-я