Arrow fun vs normal function

let sunday = {

a:1,

b:2,

c: ()=>{

console.log(this.a)

}

}

// sunday.a = 5;

console.log(sunday);

Arrow functions cannot be used to write object methods because, as you have found, since arrow functions close over the this of the lexically enclosing context, the this within the arrow is the one that was current where you defined the object. Which is to say:

Const vs let - Study properly

Destructive

Async , await

Promises

Search bar to filter list of object items{ question, answers }

Redux flow

var sample = funtion sampleOne(){

Console.log(‘dfdfddf’);

}

Function anonymous and sampleOne ??

Maxwidth and width ?

0 + true

A + true

Map vs filter => [{id:1,check:true},{id:2,check:false},{id: 3, check:true}].map( t=> if(t.check) return t.id); // return id1,null,id3

Comes the rescue. Filter => it does filtering only as results // id1, id2 // checks for truthy.

How do you check error trace back happened in child from parent component itself