

1.箭头的函数的外层函数中的this就是箭头函数中的this,当外层函数中的this发生改变时,箭头函数中的this也会发生改变

例1

```
let obj = {
  xx:20000
}
let stu2 = {
  xx:10000,
  m1:function(){
    let x = ()=>{
      console.log("x",this.xx);
    }
    x();
  }
}
stu2.m1(); //箭头函数中的this是stu2
stu2.m1.call(obj); //箭头函数中的this是obj
```

例2 :

```
let obj = {
  name:'obj'
}
function Stu(name){
  this.name = name;
  this.showName = ()=>{
    console.log(this.name);
  }
}
let s1 = new Stu("小明");
s1.showName();
```

```
var name = 'window'
var obj = {
  name:'obj',
  say2:function() {
    window.setTimeout(()=>{
      console.log(this.name);
    },20)
  },
}
obj.say2();//obj
obj.say2.call(window);//window
```

```
var Animal = function() {
  this.name = "Animal";
  this.speak = (words) => {
    //animal
    console.log(this.name + ' is saying ' + words + '.');
  }
}
```

```
var cat = new Animal();
cat.speak("miao ~"); // Animal is saying miao ~.
var speak = cat.speak;
speak("miao ~"); // Animal is saying miao ~.
```

```
var x = 10;
var obj = {
  x:20,
  e:function(){
    return ()=>{
      console.log(this.x);
    }
  }
}
obj.e();
var m = obj.e();
m();
obj.e.call(window)();
var f = obj.e.call(window);
f();
```

2.如果没有外部普通函数，则this就是window

```
var x = 0;
var a = {
  x: 10,
  bm: () => {
    console.log(this, this.x)
  }
}
a.bm(); //window
```

```
var x = 0;
var a = {
  x:10,
  b:{
    x:20,
    bm:()=>{
      console.log(this, this.x)
    }
  }
}
a.b.bm();//bm中的this是window
```