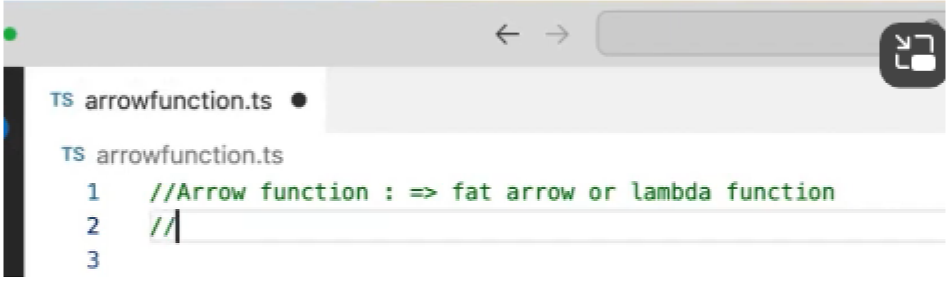
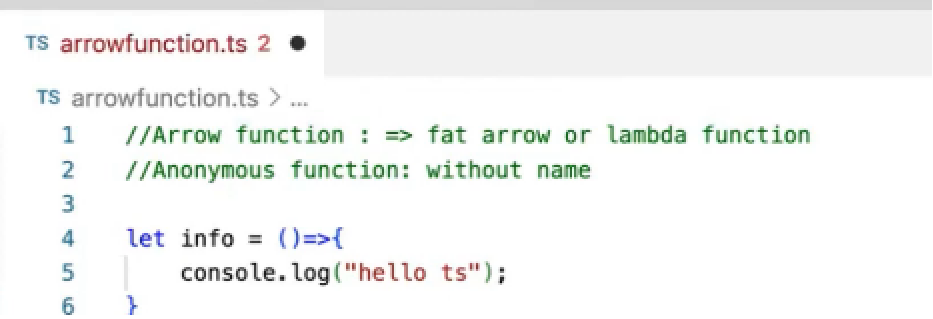
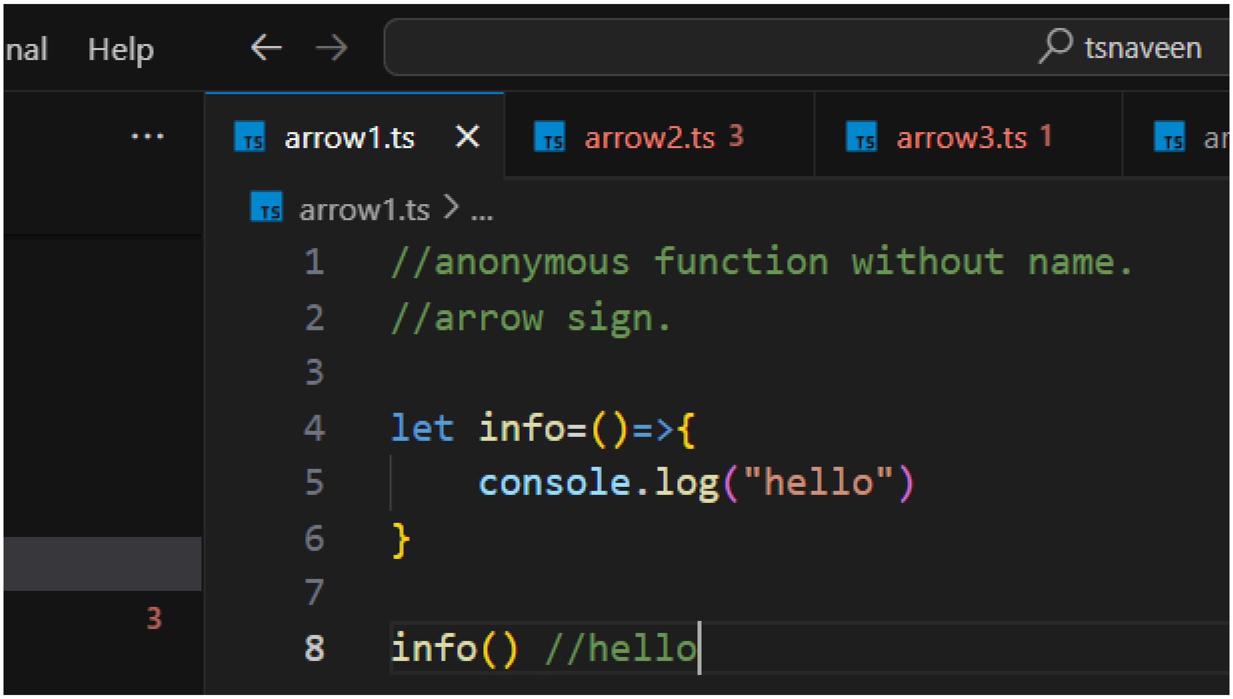


```
//Arrow function : => fat arrow or lambda function
//
```



```
//Arrow function : => fat arrow or lambda function
//Anonymous function: without name

let info = ()=>{
    console.log("hello ts");
}
```

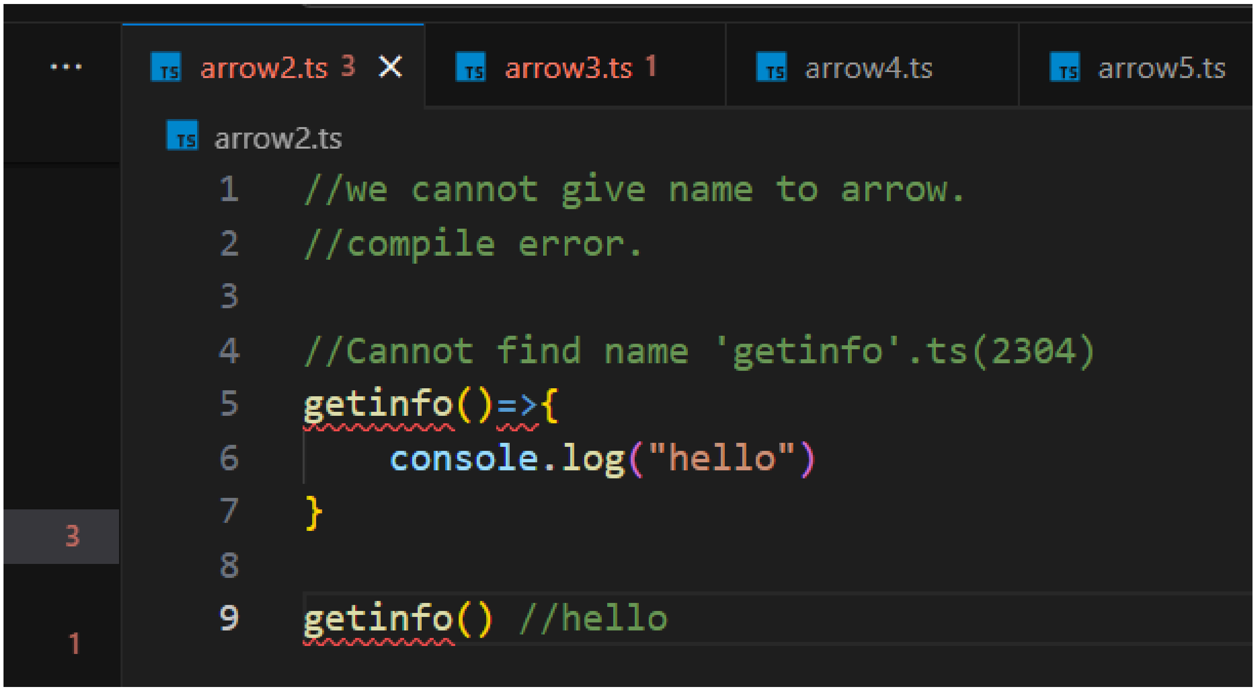


```
//anonymous function without name.
//arrow sign.

let info=()=>{
    console.log("hello")
}

info() //hello
```

Not allowed to give name

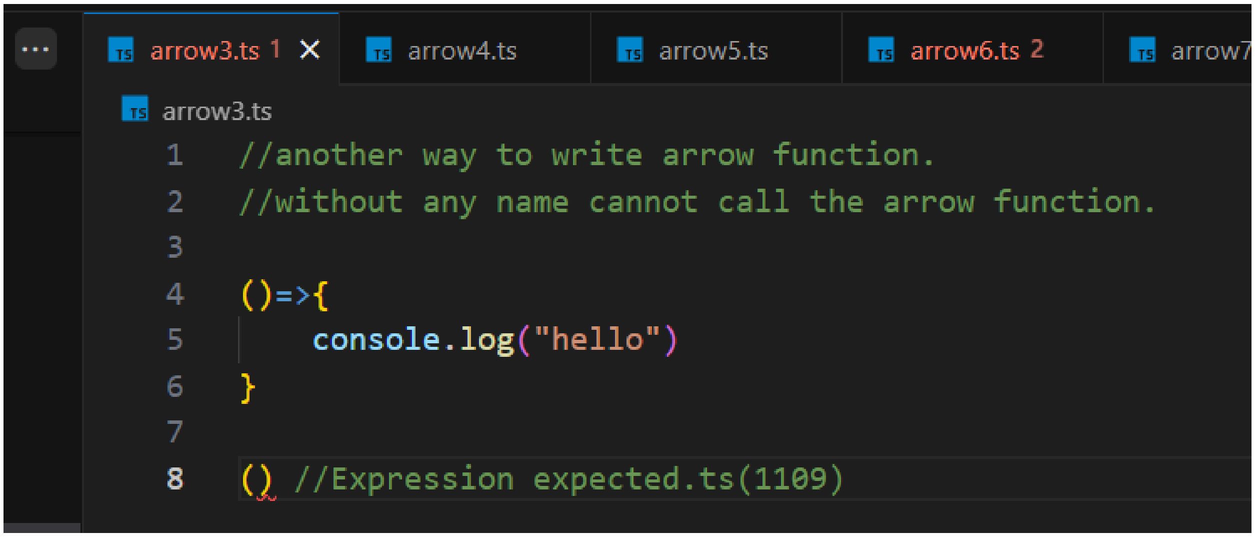


```
//we cannot give name to arrow.
//compile error.

//Cannot find name 'getinfo'.ts(2304)
getinfo()=>{
    console.log("hello")
}

getinfo() //hello
```

This also ok

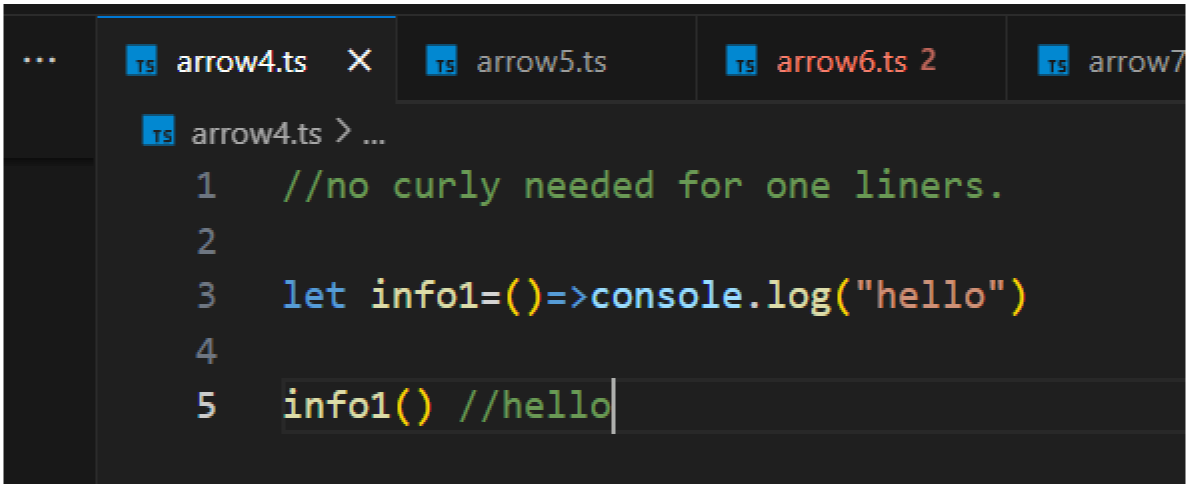


```
//another way to write arrow function.
//without any name cannot call the arrow function.

()=>{
    console.log("hello")
}

() //Expression expected.ts(1109)
```

No curly needed for one liners

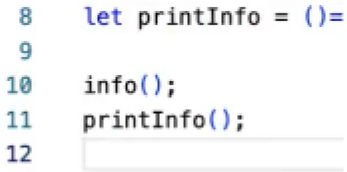


```
//no curly needed for one liners.

let info1=()=>console.log("hello")

info1() //hello
```

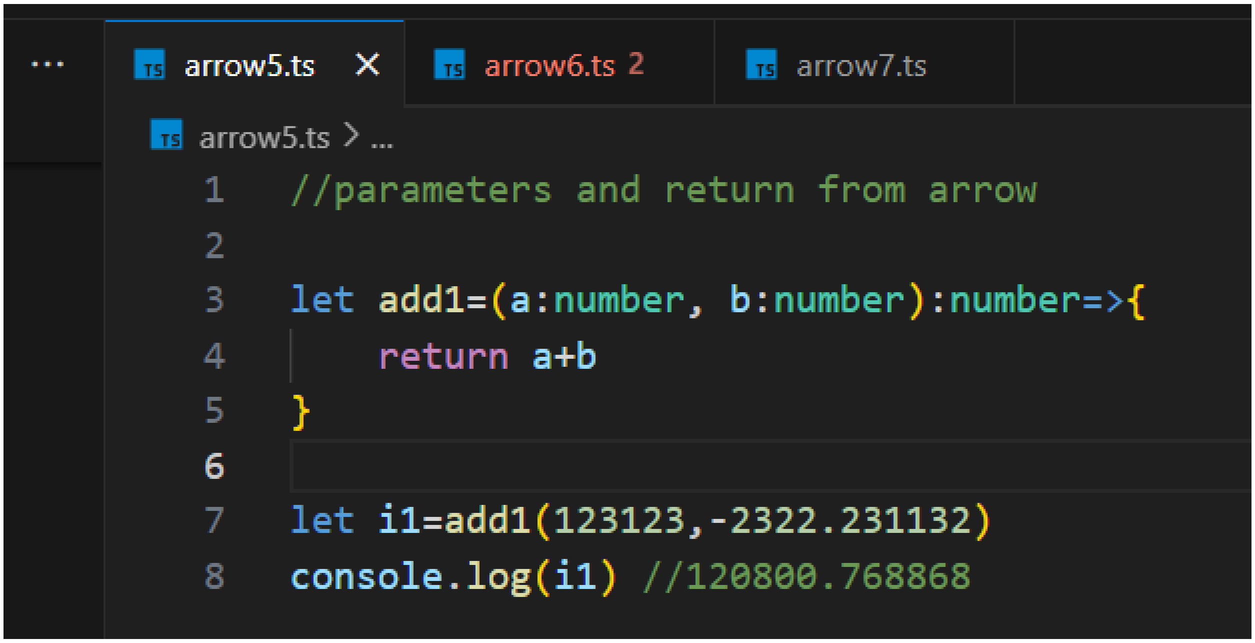
Call arrow



```
let printInfo = ()=

info();
printInfo();
```

Parameters to arrow

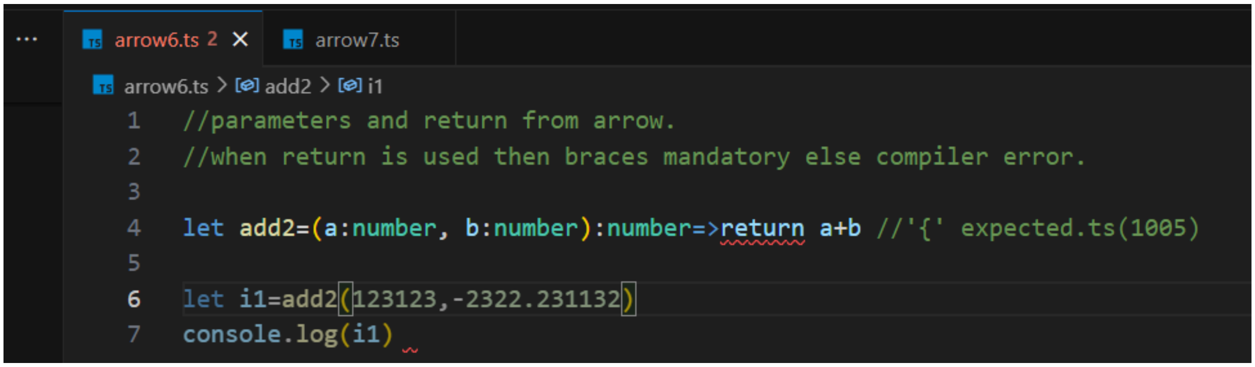


```
//parameters and return from arrow

let add1=(a:number, b:number):number=>{
    return a+b
}

let i1=add1(123123,-2322.231132)
console.log(i1) //120800.768868
```

When you use return in arrow then braces mandatory, else compile error-

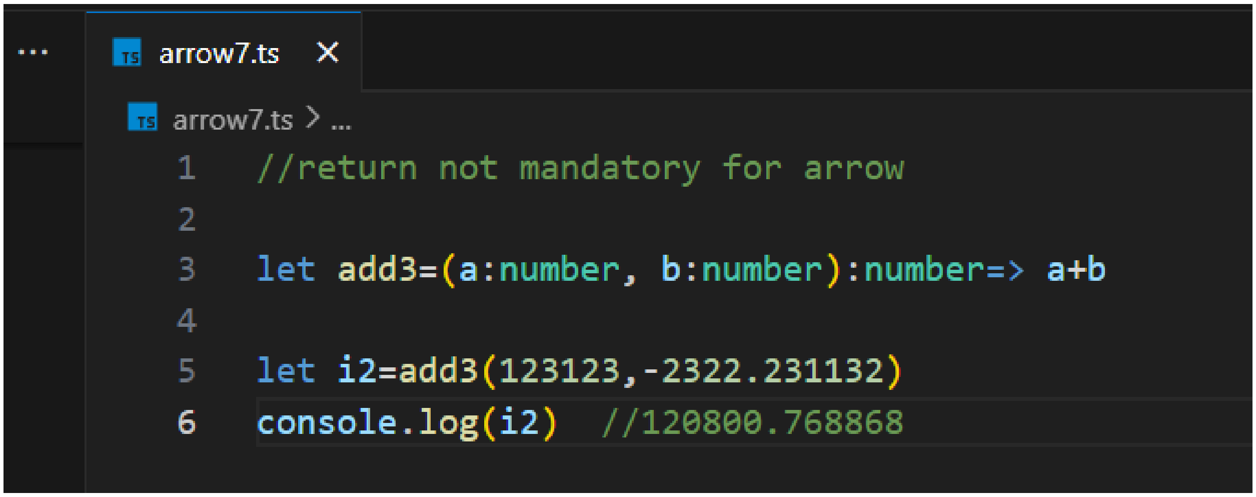


```
//parameters and return from arrow.
//when return is used then braces mandatory else compiler error.

let add2=(a:number, b:number):number=>return a+b //'{' expected.ts(1005)

let i1=add2(123123,-2322.231132)
console.log(i1)
```

So return not mandatory for arrow-



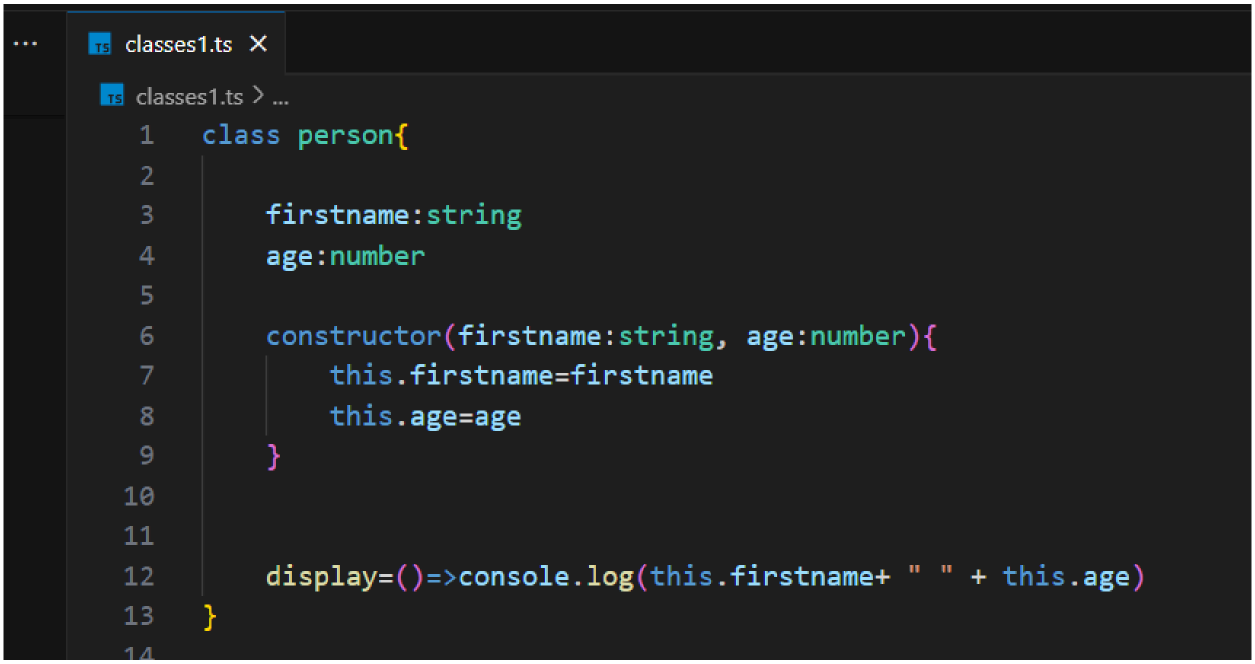
```
//return not mandatory for arrow

let add3=(a:number, b:number):number=> a+b

let i2=add3(123123,-2322.231132)
console.log(i2)  //120800.768868
```

Classes

Inside class dont use let var const for variables.
Dont use function word for functions.

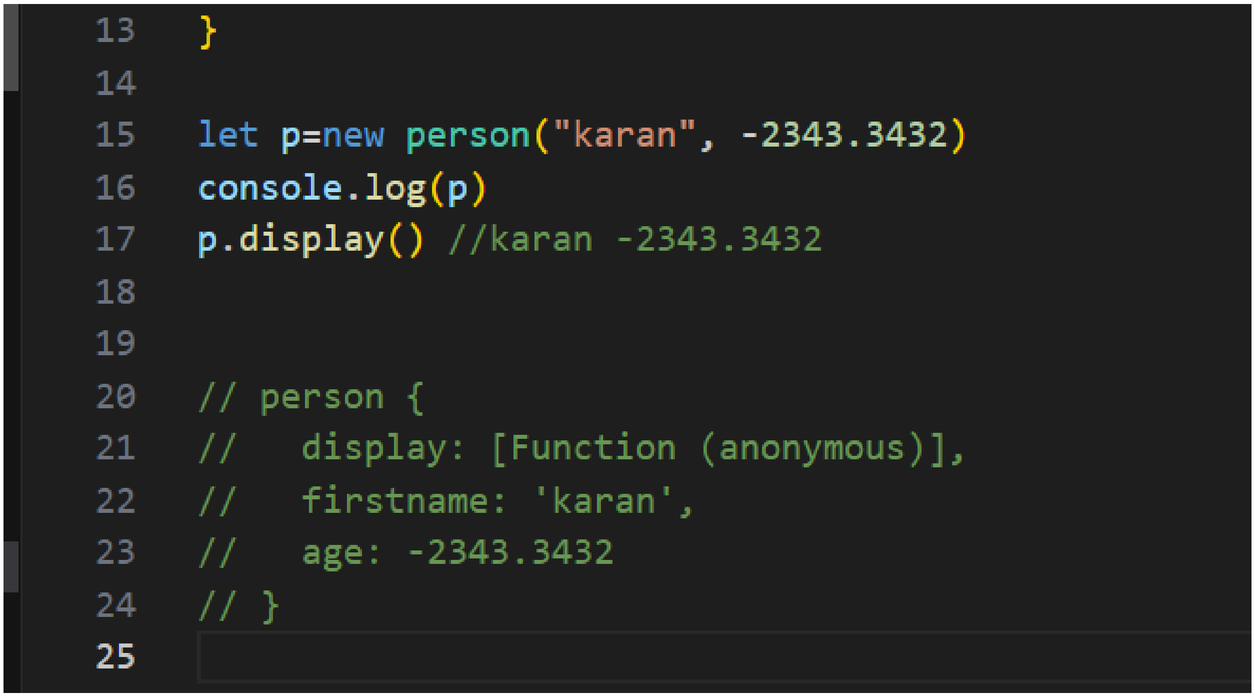


```
class person{

    firstname:string
    age:number

    constructor(firstname:string, age:number){
        this.firstname=firstname
        this.age=age
    }


    display=()=>console.log(this.firstname+ " " + this.age)
}
```



```
}

let p=new person("karan", -2343.3432)
console.log(p)
p.display() //karan -2343.3432


// person {
//   display: [Function (anonymous)],
//   firstname: 'karan',
//   age: -2343.3432
// }

```