

```
TS arrowfunction.ts •  
TS arrowfunction.ts  
1 //Arrow function : => fat arrow or lambda function  
2 //  
3
```

```
TS arrowfunction.ts 2 •  
TS arrowfunction.ts > ...  
1 //Arrow function : => fat arrow or lambda function  
2 //Anonymous function: without name  
3  
4 let info = ()=>{  
5 |   console.log("hello ts");  
6 | }
```

```
nal Help ← → tsnaven  
... TS arrow1.ts X TS arrow2.ts 3 TS arrow3.ts 1 TS ar  
TS arrow1.ts > ...  
1 //anonymous function without name.  
2 //arrow sign.  
3  
4 let info=()=>{  
5 |   console.log("hello")  
6 | }  
7  
8 info() //hello
```

Not allowed to give name



The screenshot shows a VS Code editor with four tabs: arrow2.ts, arrow3.ts, arrow4.ts, and arrow5.ts. The arrow2.ts tab is active and shows the following code:

```
1 //we cannot give name to arrow.
2 //compile error.
3
4 //Cannot find name 'getinfo'.ts(2304)
5 getinfo()=>{
6     console.log("hello")
7 }
8
9 getinfo() //hello
```

Line 5 has a red squiggly line under 'getinfo()' with the error message '//Cannot find name 'getinfo'.ts(2304)'. Line 9 also has a red squiggly line under 'getinfo()'.

This also ok



The screenshot shows a VS Code editor with five tabs: arrow3.ts, arrow4.ts, arrow5.ts, arrow6.ts, and arrow7.ts. The arrow3.ts tab is active and shows the following code:

```
1 //another way to write arrow function.
2 //without any name cannot call the arrow function.
3
4 ()=>{
5     console.log("hello")
6 }
7
8 () //Expression expected.ts(1109)
```

Line 8 has a red squiggly line under '()' with the error message '//Expression expected.ts(1109)'.

No curly needed for one liners

```
... TS arrow4.ts X TS arrow5.ts TS arrow6.ts 2 TS arrow7
TS arrow4.ts > ...
1 //no curly needed for one liners.
2
3 let info1={()=>console.log("hello")}
4
5 info1() //hello
```

Call arrow

```
8 let printInfo = ()=
9
10 info();
11 printInfo();
12
```

Parameters to arrow

```
... TS arrow5.ts X TS arrow6.ts 2 TS arrow7.ts
TS arrow5.ts > ...
1 //parameters and return from arrow
2
3 let add1=(a:number, b:number):number=>{
4     return a+b
5 }
6
7 let i1=add1(123123,-2322.231132)
8 console.log(i1) //120800.768868
```

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

```
...  TS arrow6.ts 2 X TS arrow7.ts
TS arrow6.ts > [e] add2 > [e] i1
1 //parameters and return from arrow.
2 //when return is used then braces mandatory else compiler error.
3
4 let add2=(a:number, b:number):number=>return a+b //{' expected.ts(1005)
5
6 let i1=add2(123123,-2322.231132)
7 console.log(i1)
```

So return not mandatory for arrow-

```
...  TS arrow7.ts X
TS arrow7.ts > ...
1 //return not mandatory for arrow
2
3 let add3=(a:number, b:number):number=> a+b
4
5 let i2=add3(123123,-2322.231132)
6 console.log(i2) //120800.768868
```

Classes

Inside class dont use let var const for variables.

Dont use function word for functions.

```
... classes1.ts X
classes1.ts > ...
1  class person{
2
3      firstname:string
4      age:number
5
6      constructor(firstname:string, age:number){
7          this.firstname=firstname
8          this.age=age
9      }
10
11
12      display()=>console.log(this.firstname+ " " + this.age)
13  }
14
```

```
13  }
14
15  let p=new person("karan", -2343.3432)
16  console.log(p)
17  p.display() //karan -2343.3432
18
19
20  // person {
21  //    display: [Function (anonymous)],
22  //    firstname: 'karan',
23  //    age: -2343.3432
24  // }
25
```