

# Function

Block of code

## Pre-define

substring()  
slice()  
splice()  
sort()

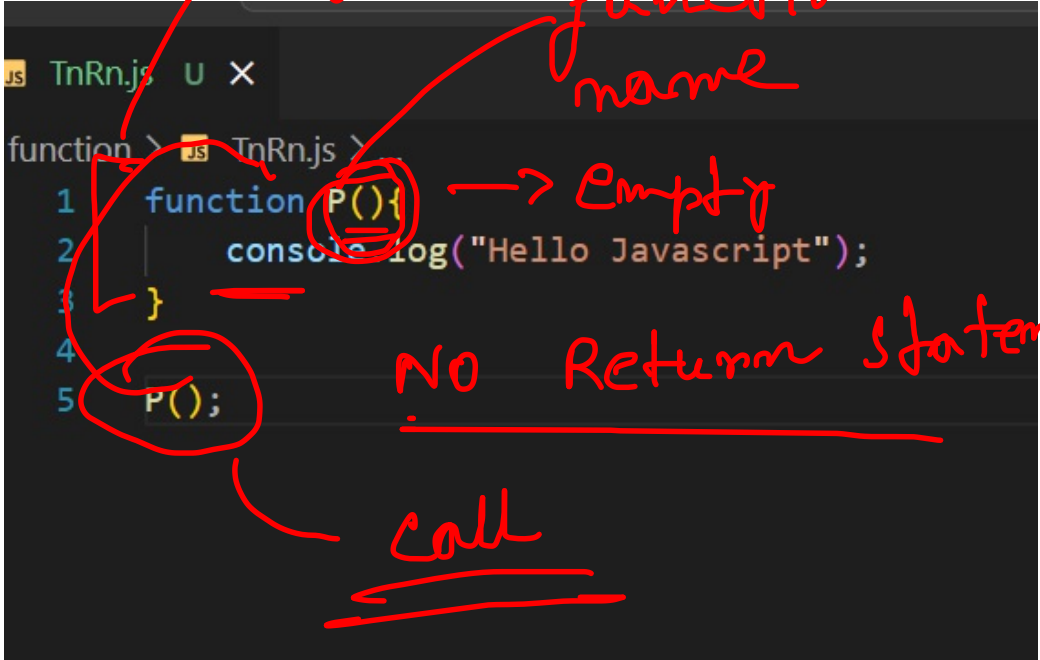
## User define

- ① Takes Nothing Returns Nothing
- ② Takes Nothing Returns Something
- ③ Takes Something Returns Nothing
- ④ Takes Something Returns Something

Takes Nothing Returns Nothing

Syntax  
function fname () {  
    // function body  
}

No Argument  
definition  
function name  
→ Empty  
No Return Statement  
call



```
JS TnRn.js U X  
function > JS TnRn.js >  
1 function P() {  
2     console.log("Hello Javascript");  
3 }  
4  
5 P();
```

The screenshot shows a code editor with a JavaScript file named 'TnRn.js'. The code contains a function definition 'function P() { console.log("Hello Javascript"); }' and a function call 'P();'. Handwritten red annotations include: 'No Argument' pointing to the empty parentheses in the function definition; 'definition' pointing to the 'function P()' part; 'function name' pointing to 'P()'; '→ Empty' pointing to the empty parentheses in the function call; 'No Return Statement' pointing to the absence of a return statement; and 'call' pointing to the 'P()' in the function call.

Takes Something      Returns Nothing

```
function sum(3num1, 2num2) {  
  let res = num1 + num2;  
  console.log(res);  
}
```

<sup>3</sup>  
sum(3, 2);

```
function > Js TsRn.js > ...  
1  function Sum(num1,num2){  
2      let res = num1+num2;  
3      console.log(res);  
4  }  
5  
6  Sum(3,6)  
7  
8  let a = 5;  
9  let b = 3;  
10 Sum(a,b);
```

# Takes Nothing Returns Something

```
function Sum() {  
  let a = 10;  
  let b = 5; let c = a + b;  
  [return a + b; ✓  
   return a + b c; ✓  
   return a, b; X  
}
```

↓ function      ↓ return

Takes Something Returns Something

```
function sum(a, b) {
```

```
  return a + b;
```

```
}
```

```
let n = sum(3, 4);
```

a, b

```
console.log(n);
```

① Swap two numbers using 3rd variable ✓✓

②

without using

# ① Function as a variable

## Syntax

```
{ let sum = function() {  
    console.log("Hello");  
}  
  
sum();
```

# Arrow Function

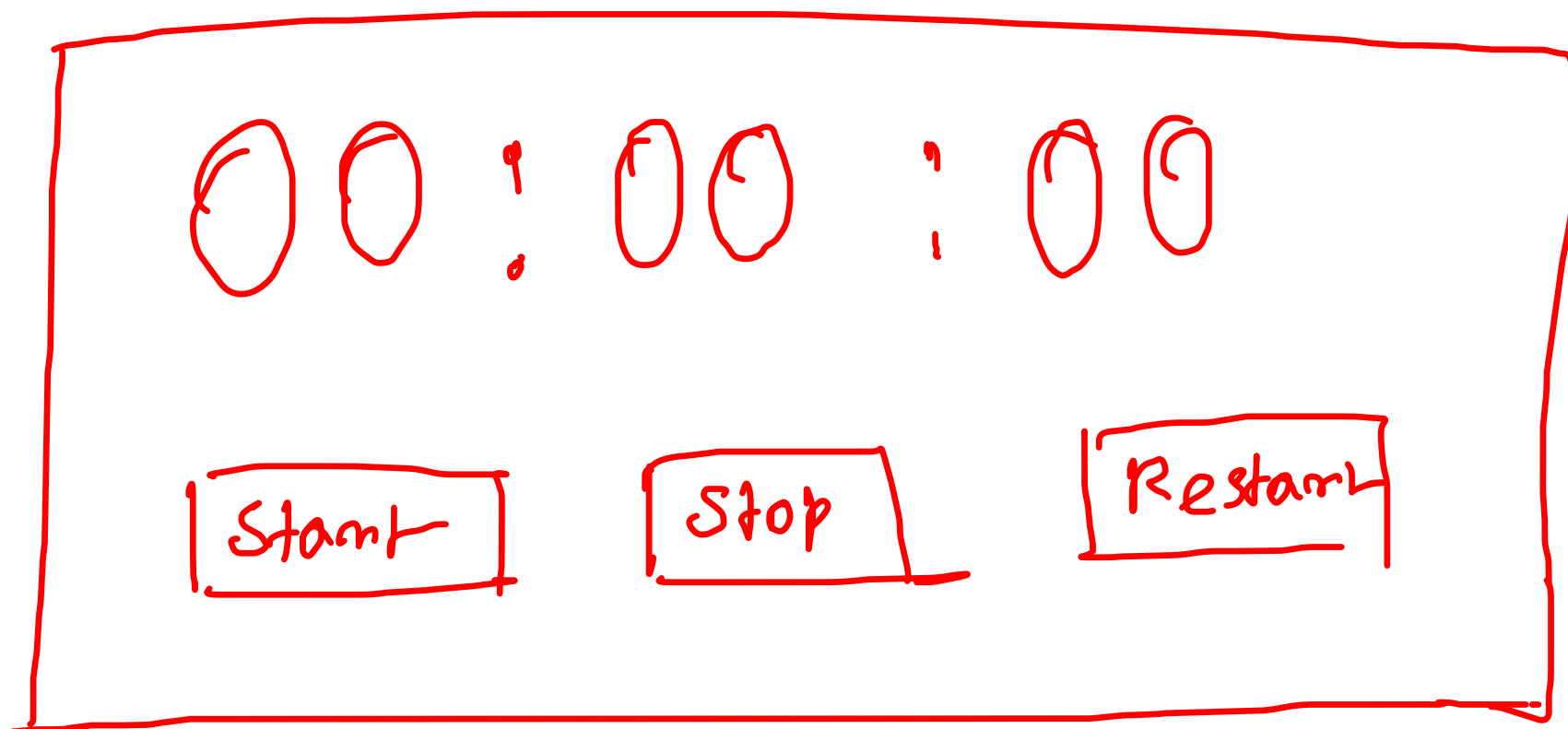
```
function > JS Arf.js > ...
1  let Sum={()=>{
2      let a=10;
3      let b=5;
4      console.log(a+b);
5  }
6  Sum();
7  let Sub=(a,b)=>{
8      let c = a-b;
9      return c;
10 }
11 let r=Sub(4,6);
12 console.log(r);
13
14 let Mult=(a,b)=>a*b;
15 console.log(Mult(5,7));
16
17 let Div=(a,b)=>({
18     a/b+(7)
19 });
20 console.log(Div(9,3));
```



# IIFE

```
function > JS IIFE.js
1  (
2  |    ()=>{
3  |        console.log("Hello");
4  |    }
5  |})();
6
7  (
8  |    (a,b)=>{
9  |        console.log(a+b);
10 |    }
11 |)(3,6);
12
13 (
14 |    ()=>{
15 |
16 |    }
17 |)();
```

# Stop watch



```

10 let Start={()=>{
11     console.log("Start");
12     interval=setInterval(()=>{
13         if(sec<59){
14             sec++;
15         }else if(min<59){
16             sec=0;
17             min++;
18         }else{
19             min=0;
20             hr++;
21         }
22     })
23 }

```

1 2 3 4 ↑ sec