

## \* Functions :

functionname

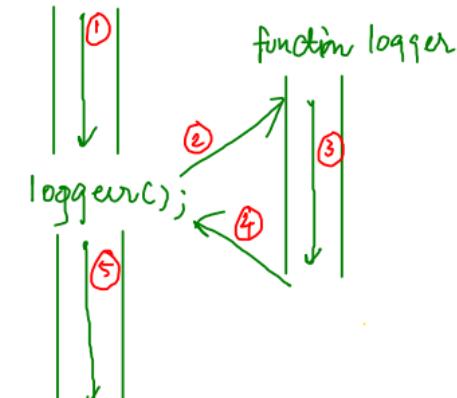
```
499 function logger() {  
500   console.log("Hi I am Anurag");  
501   ↓  
502   console.log("I work at AccioJob");  
503   ↓  
504 }  
505 logger();
```

call/invoke/run

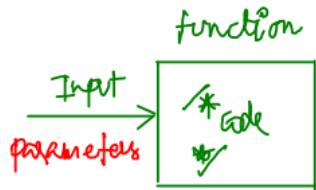
→ When the function is called, all the code inside it is executed after that only remaining code followed by function call is executed.

→ DRY - do not repeat yourself, write code once and use it multiple times.

- ① functions can take inputs in form of parameters / arguments.
- ② function can provide response / output using a return statement.
- ③ return statement shutdowns / terminates the function, any code after this will not be executed.



①



## f: Addition of Two Number

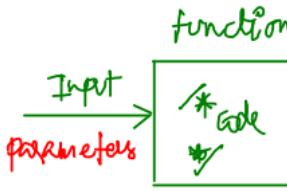
```

519 |   function add(a, b) {
520 |     const c = a + b;
521 |     console.log(c);
522 |   }
523 |
524 |   add(12, 13);
525 |   add(21, 32);
526 |   add(10, 20);
  
```

( $a, b$ ) are variables  
created in the function  
(you can have any choice  
of variables)

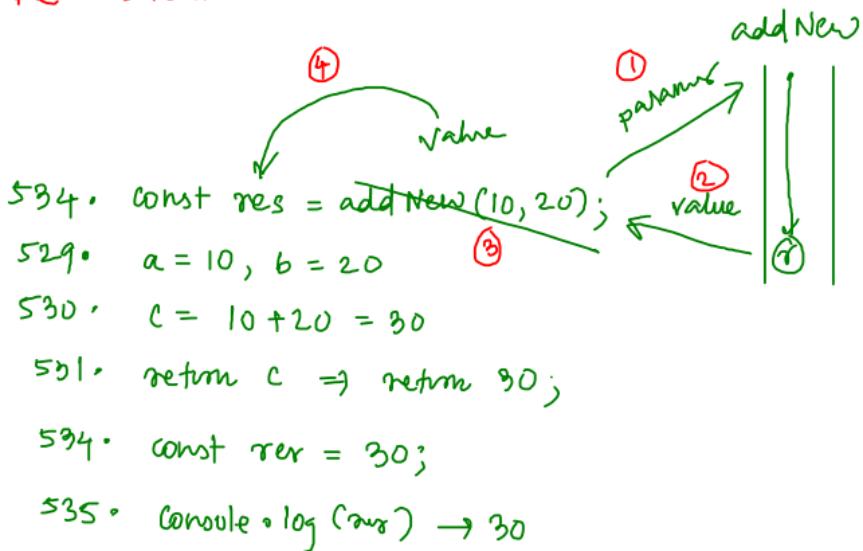
- 524.  $\text{add}(12, 13)$
- 519.  $a = 12, b = 13$
- 520.  $\text{const } c = 12 + 13 = 25$
- 521.  $\text{console.log}(c) \rightarrow 25$
- 525.  $\text{add}(21, 32)$
- 519.  $a = 21, b = 32$
- 520.  $\text{const } c = a + b = 21 + 32 = 53$
- 521.  $\text{console.log}(c) \rightarrow 53$

②



(do not print inside function  
but send the result to the  
caller)

```
529 | function addNew(a, b){  
530 |   const c = a + b;  
531 |   ↴  
532 |   return c;  
533 | }  
534 | ↴  
534 | const res = addNew(10, 20);  
535 | console.log(res);
```



```

529 function addNew(a, b) {
530   const c = a + b;
531   return c;
532 }
533
534 function multiply(a, b) {
535   const c = a * b;
536   return c;
537 }
538
539 const p = 10;
540 const q = 20;
541 const r = 30;
542 const s = 40;
543 // 10+20 = 30
544 // (p + q) * (r + s)
545 // op1 * op2
546 const op1 = addNew(p, q);
547 const op2 = addNew(r, s);
548 const ans = multiply(op1, op2);
549 console.log(ans);

```

\* return will not print anything, It just sends the value back.

546 · const op1 = addNew(p, q)  
= addNew(10, 20)

529 · a = 10, b = 20

530 · c = 10 + 20 = 30

531 · return 30;

546 · const op1 = 30;

547 · const op2 = addNew(r, s)  
= addNew(30, 40)

529 · a = 30, b = 40

530 · c = 30 + 40 = 70

531 · return 70;

547 · const op2 = 70;

548 · const ans = multiply(op1, op2) = multiply(30, 70)

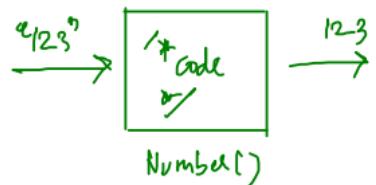
534 · a = 30, b = 70

535 · c = 30 \* 70 = 2100

536 · return 2100

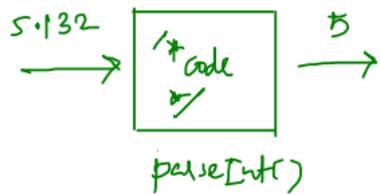
548 · const ans = 2100;

→ Const a = Number("123");



We know what are  
ip, op but  
logic X

→ Const b = parseInt(5.132)



\* console.log(...),

Math.max(1, 2, 3, 4) → 4

Math.min(2, -1, 0, 2) → -1

\* TV Remote,

① volume 'increases',  
do you know how sensor  
b/w remote and TV work?

(physics) ↓  
NO

ip: click

op: vol increased

→ Abstraction

(Hiding Implementation details)

→ reusability (DRY)

givenDigit = 2

$$986272 \% 10 = 2$$

$$\downarrow 986272 / 10$$

$$98627 \% 10 = 7$$

$$\downarrow 10$$

$$9862 \% 10 = 2$$

$$\downarrow 10$$

$$186 \% 10 = 6$$

$$\downarrow 10$$

$$18 \% 10 = 8$$

$$\downarrow 10$$

$$9 \% 10 = 9$$

$$\downarrow 10$$

0

\*  $\text{if}(\text{lastdigit} == \text{givenDigit}) \{$   
     $\text{cnt}++$   
}

$\text{cnt} = \emptyset \neq 2$

\* Frequency of digit in a number :

## \* Binary to decimal:

$$\text{hwhn} = 25$$

decimal (0-9)

$$\Rightarrow 1 \ 2 \ 3 \ 4$$

$$10^3 \ 10^2 \ 10^1 \ 10^0$$

$$(1000 \ 100 \ 10 \ 1)$$

$$\Rightarrow 4 \times 1 + 3 \times 10$$

$$+ 2 \times 100$$

$$+ 1 \times 1000$$

$$= 1234$$

\* `parseInt(s, 2)`

$$\Rightarrow \text{parseInt}(10011, 2) \rightarrow 19$$

binary (0-1)

$$\Rightarrow \begin{matrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{matrix}$$

$$16 \quad 8 \quad 4 \quad 2 \quad 1$$

$$\Rightarrow 1 \times 1 + 1 \times 2 + 0 \times 4 + 0 \times 8 + 1 \times 16$$

$$\Rightarrow 1 + 2 + 0 + 0 + 16$$

$$\Rightarrow 19$$

Eg: 10111 → 23

Eg: 1111 → 15

Eg: 1101 → 13

\* Diamond pattern :

Eg:  $n = 5$

*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*
*	*	*	*	*

Pyramidal of 3 Rows  
Eg:  $n = 3$

$$\begin{aligned} a+1 &= 3 \\ \left(\frac{5}{2}\right) + 1 & \end{aligned}$$

0	1	2	3	3
0		*		
1		*	*	*
2		*	*	*

# Row 8 = 3 = N

cols  
↓  
Spacers      Stars  
↓

$$\# Spacers = n - r - 1$$

$$\# Stars = (2 + r) + 1$$

$$\left(\frac{5}{2}\right) + 1 = 3$$

*	*	*	*	*
*	*	*	*	
*				

$$\begin{array}{ll} r=0 & 2(3-0-1) \quad 1(2+0+1) \\ r=1 & 1(3-1-1) \quad 3(2+1+1) \\ r=2 & 0(3-2-1) \quad 5(2+2+1) \end{array}$$

\* Just run the loop  
in reverse

Inverted pyramidal of 3 Rows

$\text{printDiamond}(5)$  → pyramid  $(\lceil \frac{5}{2} \rceil + 1) = 3$   
 $\text{printDiamond}(5)$  → inv Pyramid  $(\lceil \frac{5}{2} \rceil + 1) = 3$

- \* run the pyramid function one time lesser; instead of  $r=0, r=1, r=2$
- Only run for  $r=0, r=1$