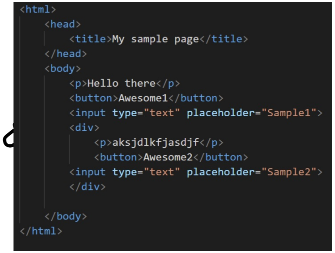
Server $\xrightarrow{html}$ Browser Reads the html

Presents a View

Prepares DOM Document Object Model

<butto



```html
<html>
    <head>
        <title>My sample page</title>
    </head>
    <body>
        <p>Hello there</p>
        <button>Awesome1</button>
        <input type="text" placeholder="Sample1">
        <div>
            <p>aksjdlkfjasdjf</p>
            <button>Awesome2</button>
        <input type="text" placeholder="Sample2">
        </div>

    </body>
</html>
```

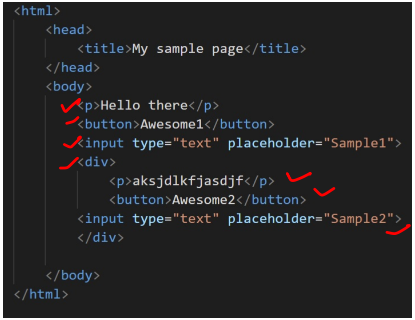
Button

dom . docu

→ Browser

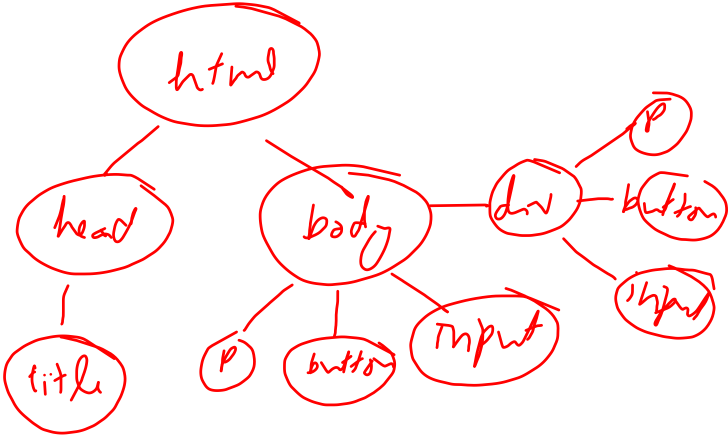
↓

→ DOM

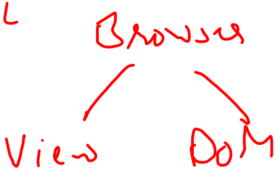


```
<html>
    <head>
        <title>My sample page</title>
    </head>
    <body>
        <p>Hello there</p>
        <button>Awesome1</button>
        <input type="text" placeholder="Sample1">
        <div>
            <p>aksjdlkfjasdjf</p>
            <button>Awesome2</button>
        <input type="text" placeholder="Sample2">
        </div>

    </body>
</html>
```



Json → HTML



1000

LS vs BS

1000 comparisons

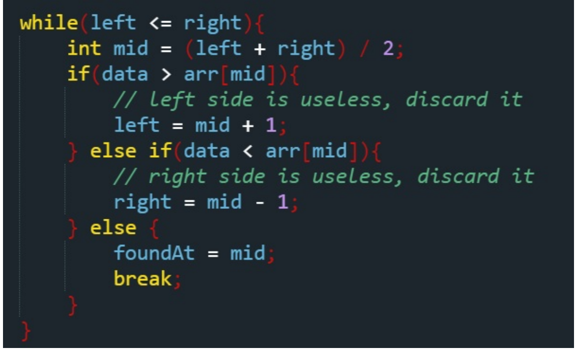
n ''

No pre-requisites

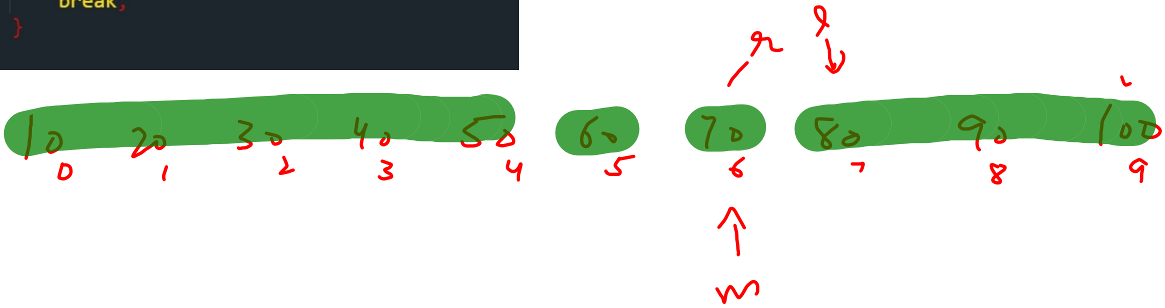
10 comparisons

$\log_2 n$ ''

Sorted array



```
while(left <= right){
    int mid = (left + right) / 2;
    if(data > arr[mid]){
        // left side is useless, discard it
        left = mid + 1;
    } else if(data < arr[mid]){
        // right side is useless, discard it
        right = mid - 1;
    } else {
        foundAt = mid;
        break;
    }
}
```



77

$$\left.\begin{array}{l} n \\ n/2 \\ n/4 \\ \vdots \\ 1 \end{array}\right] x \quad \left.\begin{array}{l} T(n) = k + \cancel{T(n/2)} \\ \cancel{T(n/2)} = k + \cancel{T(n/4)} \\ \cancel{T(n/4)} = k + \cancel{T(n/8)} \\ \quad \vdots \\ \cancel{T(1)} = k + \underline{\quad} \end{array}\right] x$$

$$T(n) = k \cdot x \qquad T(n) = k \cdot (\log_2 n + 1)$$

$$\boxed{T(n) \propto \log_2 n}$$

$$a_x = a_0 r^{x-1}$$

$$1 = n \left(\frac{1}{2}\right)^{x-1}$$

$$1 = \frac{n}{2^{x-1}} \Rightarrow 2^{x-1} = n$$

$$\log_2 2^{x-1} = \log_2 n$$

$$x - 1 = \log_2 n$$

$$x = \log_2 n + 1$$

1	~~1024~~
2	~~512~~
3	~~256~~
4	~~128~~
5	~~64~~
6	~~32~~
7	~~16~~
~~8~~	8
~~4~~	9
~~2~~	10
~~1~~	11

O n^3

a b c d
0 1 2 3

0

00 = a ✓

01 = ab ✓

02 = abc ✓

03 = abcd ✓

1

11 = b ✓

12 = bc ✓

13 = bcd ✓

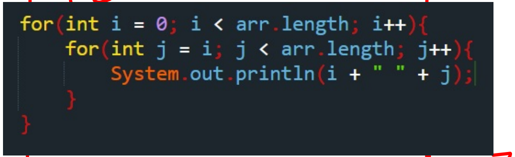
2

22 = c ✓

23 = cd ✓

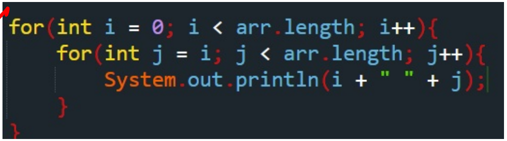
3

33 = d ✓



```
for(int i = 0; i < arr.length; i++){
    for(int j = i; j < arr.length; j++){
        System.out.println(i + " " + j);
    }
}
```

✓



```
for(int i = 0; i < arr.length; i++){
    for(int j = i; j < arr.length; j++){
        System.out.println(i + " " + j);
    }
}
```

a	b	c	d
0	1	2	3

0 0 = a   1 1 = b   2 2 = c   3 3 = d

0 1 = ab   1 2 = bc   2 3 = cd

0 2 = abc   1 3 = bcd

0 3 = abcd

arr	2	4	1	0	3
	0	1	2	3	4
	✓	✓	✓	✓	✓

arr[0] = 2

arr[1] = 4

arr[2] = 1

a[3] = 0

a[4] = 3

inv	3	2	0	4	1
	0	1	2	3	4
	✓	✓	✓	✓	✓

inv[2] = 0 ⇒ inv[arr[0]] = 0

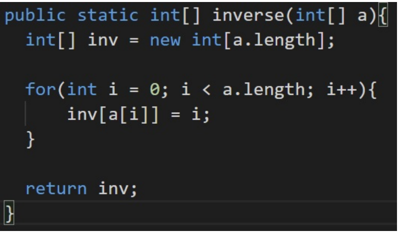
inv[4] = 1 ⇒ inv[arr[1]] = 1

inv[1] = 2 ⇒ inv[arr[2]] = 2

i[0] = 3 ⇒ i[a[3]] = 3

i[3] = 4

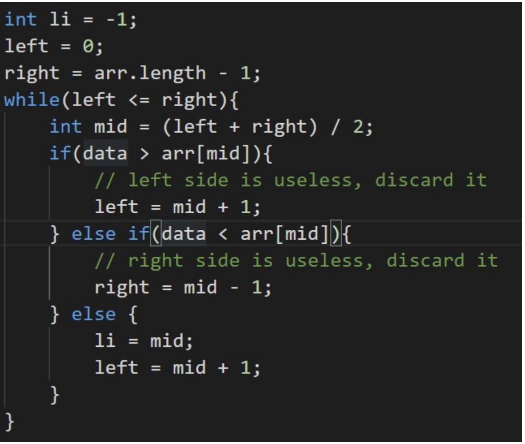
i[a[4]] = 4



```
public static int[] inverse(int[] a){
    int[] inv = new int[a.length];

    for(int i = 0; i < a.length; i++){
        inv[a[i]] = i;
    }

    return inv;
}
```



```
int li = -1;
left = 0;
right = arr.length - 1;
while(left <= right){
    int mid = (left + right) / 2;
    if(data > arr[mid]){
        // left side is useless, discard it
        left = mid + 1;
    } else if(data < arr[mid]){
        // right side is useless, discard it
        right = mid - 1;
    } else {
        li = mid;
        left = mid + 1;
    }
}
```

40

li = ~~8~~ 8

10	20	30	30	30	40	40	40	40	50	60
0	1	2	3	4	5	6	7	8	9	10

Dom is an object tree which is created by browser for the programmer.

[Programmer can interact with view of page using dom]