



# JAVASCRIPT TRICKY OUTPUT-BASED QUESTIONS



@DimpleKumari

Forming a network of fantastic coders.



## 1. What's the output?

```
function sayHi() {  
    console.log(name);  
    console.log(age);  
    var name = 'Dimple';  
    let age = 21;  
}  
  
sayHi();
```



- A: Dimple and undefined
- B: Dimple and ReferenceError
- C: ReferenceError and 21
- D: undefined and ReferenceError



@DimpleKumari

Forming a network of fantastic coders.



## 2. What's the output?

```
for (var i = 0; i < 3; i++) {  
    setTimeout(() => console.log(i), 1);  
}  
  
for (let i = 0; i < 3; i++) {  
    setTimeout(() => console.log(i), 1);  
}
```

- A: 0 1 2 and 0 1 2
- B: 0 1 2 and 3 3 3
- C: 3 3 3 and 0 1 2



@DimpleKumari

Forming a network of fantastic coders.



### 3. What's the output?

```
const shape = {  
    radius: 10,  
    diameter() {  
        return this.radius * 2;  
    },  
    perimeter: () => 2 * Math.PI * this.radius,  
};  
  
console.log(shape.diameter());  
console.log(shape.perimeter());
```



- A: 20 and 62.83185307179586
- B: 20 and NaN
- C: 20 and 63
- D: NaN and 63



@DimpleKumari

Forming a network of fantastic coders.



#### 4. What's the output?

```
+true;  
!'Dimple';
```

- A: 1 and false
- B: false and NaN
- C: false and false



@DimpleKumari

Forming a network of fantastic coders.



## 5. Which one is true?

```
const bird = {  
    size: 'small',  
};
```

```
const mouse = {  
    name: 'Mickey',  
    small: true,  
};
```

- A: `mouse.bird.size` is not valid
- B: `mouse[bird.size]` is not valid
- C: `mouse[bird["size"]]` is not valid
- D: All of them are valid



@DimpleKumari

Forming a network of fantastic coders.



## 6. What's the output?

```
let c = { greeting: 'Hey!' };
let d;

d = c;
c.greeting = 'Hello';
console.log(d.greeting);
```

- A: Hello
- B: Hey!
- C: undefined
- D: ReferenceError
- E: TypeError



@DimpleKumari

Forming a network of fantastic coders.



## 7. What's the output?

```
let a = 3;  
let b = new Number(3);  
let c = 3;  
  
console.log(a == b);  
console.log(a === b);  
console.log(b === c);
```

- A: true false true
- B: false false true
- C: true false false
- D: false true true



@DimpleKumari

Forming a network of fantastic coders.



## 8. What's the output?

```
class Chameleon {  
    static colorChange(newColor) {  
        this.newColor = newColor;  
        return this.newColor;  
    }  
  
    constructor({ newColor = 'green' } = {}) {  
        this.newColor = newColor;  
    }  
}  
  
const freddie = new Chameleon({ newColor: 'purple' });  
console.log(freddie.colorChange('orange'));
```

- A: orange
- B: purple
- C: green
- D: TypeError



@DimpleKumari

Forming a network of fantastic coders.



## 9. What's the output?

```
let greeting;  
greetign = {};// Typo!  
console.log(greetign);
```

- A: {}
- B: ReferenceError: greetign is not defined
- C: undefined



@DimpleKumari

Forming a network of fantastic coders.



---

## 10. What happens when we do this?

```
function bark() {  
  console.log('Woof!');  
}  
  
bark.animal = 'dog';
```

- A: Nothing, this is totally fine!
- B: `SyntaxError`. You cannot add properties to a function this way.
- C: "Woof" gets logged.
- D: `ReferenceError`



@DimpleKumari

Forming a network of fantastic coders.



## 11. What's the output?

```
function Person(firstName, lastName) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
}  
  
const member = new Person('Dimple', 'Hallie');  
Person.getFullName = function() {  
    return `${this.firstName} ${this.lastName}`;  
};  
  
console.log(member.getFullName());
```

- A: TypeError
- B: SyntaxError
- C: Dimple Hallie
- D: undefined undefined



@DimpleKumari

Forming a network of fantastic coders.



## 12. What's the output?

```
function Person(firstName, lastName) {  
  this.firstName = firstName;  
  this.lastName = lastName;  
}  
  
const Dimple = new Person('Dimple', 'Kumari');  
const Twinkle = Person('Twinkle', 'Kumari');  
  
console.log(Dimple);  
console.log(Twinkle);
```

- A: Person {firstName: "Dimple", lastName: "Kumari"} and undefined
- B: Person {firstName: "Dimple", lastName: "Kumari"} and Person {firstName: "Twinkle", lastName: "Kumari"}
- C: Person {firstName: "Dimple", lastName: "Kumari"} and {}
- D: Person {firstName: "Dimple", lastName: "Kumari"} and ReferenceError



@DimpleKumari

Forming a network of fantastic coders.



13. What are the three phases of event propagation?

- A: Target > Capturing > Bubbling
- B: Bubbling > Target > Capturing
- C: Target > Bubbling > Capturing
- D: Capturing > Target > Bubbling

14. All objects have prototypes.

- A: true
- B: false



@DimpleKumari

Forming a network of fantastic coders.



## 15. What's the output?

```
function sum(a, b) {  
    return a + b;  
}  
  
sum(1, '2');
```

- A: NaN
- B: TypeError
- C: "12"
- D: 3



@DimpleKumari

Forming a network of fantastic coders.



## 16. What's the output?

```
let number = 0;  
console.log(number++);  
console.log(++number);  
console.log(number);
```

- A: 

1	1	2
---	---	---
- B: 

1	2	2
---	---	---
- C: 

0	2	2
---	---	---
- D: 

0	1	2
---	---	---



THINK



@DimpleKumari

Forming a network of fantastic coders.



## 17. What's the output?

```
function getPersonInfo(one, two, three) {  
    console.log(one);  
    console.log(two);  
    console.log(three);  
}  
  
const person = 'Dimple';  
const age = 21;  
  
getPersonInfo`${person} is ${age} years old`;
```



- A: "Dimple" 21 ["", " is ", " years old"]
- B: ["", " is ", " years old"] "Dimple" 21
- C: "Dimple" ["", " is ", " years old"] 21



@DimpleKumari

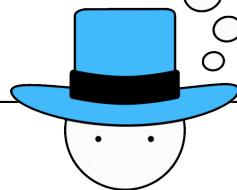
Forming a network of fantastic coders.



## 18. What's the output?

```
function checkAge(data) {  
  if (data === { age: 18 }) {  
    console.log('You are an adult!');  
  } else if (data == { age: 18 }) {  
    console.log('You are still an adult.');//  
  } else {  
    console.log(`Hmm.. You don't have an age I guess`);  
  }  
}  
  
checkAge({ age: 18 });
```

- A: You are an adult!
- B: You are still an adult.
- C: Hmm.. You don't have an age I guess



@DimpleKumari

Forming a network of fantastic coders.

## 19. What's the output?

```
function getAge(...args) {  
  console.log(typeof args);  
}  
  
getAge(21);
```

- A: "number"
- B: "array"
- C: "object"
- D: "NaN"



@DimpleKumari

Forming a network of fantastic coders.



## 20. What's the output?

```
function getAge() {  
  'use strict';  
  age = 21;  
  console.log(age);  
}  
  
getAge();
```

- A: 21
- B: undefined
- C: ReferenceError
- D: TypeError



@DimpleKumari

Forming a network of fantastic coders.



# THANK YOU FOR READING!

If you found this informative and valuable, I'd love for you to connect with me. Follow me [Medium](#), [Codepen](#), and connect with me on [LinkedIn](#) to stay updated on the latest in web development, interviews, and more.

Let's connect!

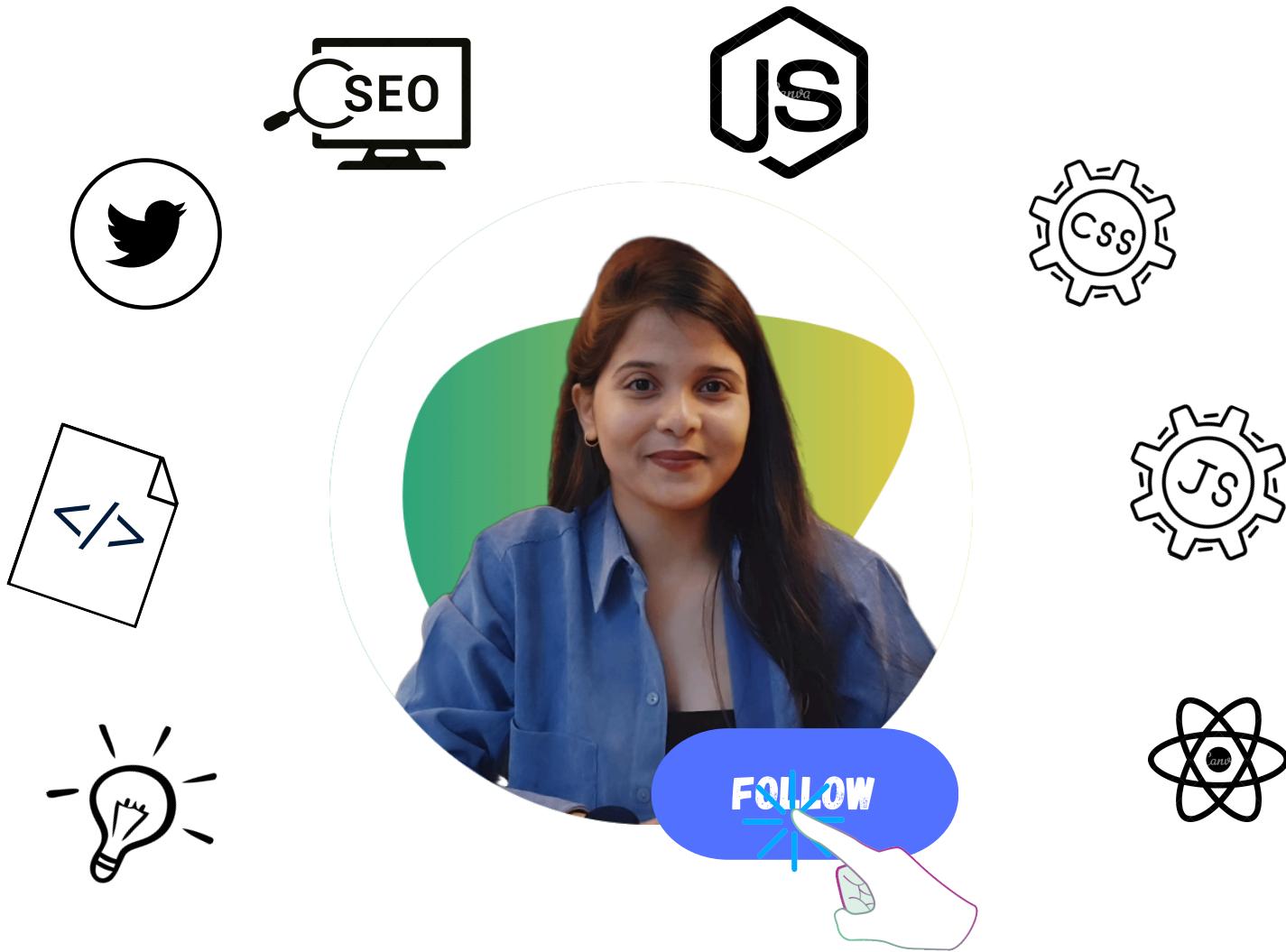
- 💼 [LinkedIn](https://www.linkedin.com/in/dimple-kumari/) — <https://www.linkedin.com/in/dimple-kumari/>
- 🔗 [Medium](https://medium.com/@dimplekumari0228) — <https://medium.com/@dimplekumari0228>
- 🖌️ [Codepen](https://codepen.io/DIMPLE2802) — <https://codepen.io/DIMPLE2802>



@DimpleKumari

Forming a network of fantastic coders.





# DIMPLE KUMARI

Forming a network of fantastic coders.

