

JavaScript Output-Based Questions

1.

```
let a = 10;  
(function () {  
    console.log(a);  
    let a = 20;  
})();
```

2.

```
const obj = {  
    a: 1,  
    b: function () {  
        console.log(this.a);  
    }  
};
```

```
const a = 100;  
obj.b.call(null);
```

3.

```
function foo() {  
    return  
    {  
        bar: "baz"  
    };  
}
```

```
console.log(typeof foo());
```

4.

```
console.log([] + []);  
console.log([] + {});  
console.log({} + []);
```

5.

```
function makeCounter() {  
  let count = 0;  
  return function () {  
    return count++;  
  };  
}  
  
const counter1 = makeCounter();  
const counter2 = makeCounter();  
  
console.log(counter1());  
console.log(counter1());  
console.log(counter2());
```

6.

```
let x = 5;  
const y = x++;  
  
console.log(x);  
console.log(y);
```

7.

```
const obj = {  
  message: 'Hello',  
  greet: () => {  
    console.log(this.message);  
  }  
};  
  
obj.greet();
```

8.

```
(async function() {  
  await Promise.resolve();  
  console.log(1);  
})();  
console.log(2);
```

9.

```
let arr = [10, 20, 30];  
arr[100] = 1000;  
console.log(arr.length);
```

10.

```
var x = 21;  
var fun = function () {  
    console.log(x);  
    var x = 20;  
};  
fun();
```

11.

```
let a = { x: 1 };  
let b = a;  
a.x = 2;  
a = { x: 3 };  
console.log(b.x);
```

12.

```
let a = 1;  
{  
    let a = 2;  
    {  
        let a = 3;  
        console.log(a);  
    }  
    console.log(a);  
}  
console.log(a);
```

13.

```
for (var i = 0; i < 3; i++) {
```

```
        setTimeout(() => console.log(i), 100);  
    }  
}
```

14.

```
console.log(typeof null);  
console.log(null instanceof Object);
```

15.

```
const func = (function() {  
    let counter = 0;  
    return function() {  
        return ++counter;  
    }  
})();
```

```
console.log(func());  
console.log(func());
```

16.

```
let obj = { name: "JS" };  
Object.freeze(obj);  
obj.name = "JavaScript";  
console.log(obj.name);
```

17.

```
let x = NaN;  
console.log(x === x);
```

18.

```
function A() {  
    this.name = "A";  
    return { name: "B" };  
}
```

```
const obj = new A();
```

```
console.log(obj.name);
```

19.

```
function* gen() {  
    yield 1;  
    yield 2;  
    yield 3;  
}  
const g = gen();  
console.log(g.next().value);  
console.log(g.next().value);  
console.log(g.next().value);
```

20.

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

21.

```
console.log(1 < 2 < 3);  
console.log(3 > 2 > 1);
```

22.

```
(function() {  
    var a = b = 5;  
})();  
console.log(typeof b);  
console.log(typeof a);
```

23.

```
const a = [1, 2, 3];  
a.length = 0;  
console.log(a[0]);
```

24.

```
function Person() {}  
Person.prototype.sayHi = () => {  
    console.log("Hi");  
};  
  
const p = new Person();  
p.sayHi();
```

25.

```
let obj = {  
    get value() {  
        return 42;  
    }  
};  
  
console.log(obj.value);
```

26.

```
let count = 0;  
  
const intervalId = setInterval(() => {  
    console.log(count++);  
    if (count > 2) clearInterval(intervalId);  
}, 10);
```

27.

```
let a = [1, 2];  
let b = [1, 2];  
  
console.log(a == b);  
console.log(a === b);
```

28.

```
let x = 0;
```

```
if (true) {  
  let x = 1;  
  if (true) {  
    let x = 2;  
    console.log(x);  
  }  
  console.log(x);  
}  
console.log(x);
```

29.

```
const x = "5";  
const y = 5;  
  
console.log(x == y);  
console.log(x === y);
```

30.

```
let obj = {  
  name: "John",  
  getName: function () {  
    return this.name;  
  }  
};  
  
let getName = obj.getName;  
console.log(getName());
```