

JavaScript Model Test 01

1. What's the output?

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

sayHi();

- a. Lydia and undefined
- b. Lydia and ReferenceError
- c. ReferenceError and 21
- d. undefined and ReferenceError

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
- d. None of above

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

4. 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

5. 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

6. 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

7. What's the output?

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

- a. {}
- b. ReferenceError: greetign is not defined
- c. Undefined
- d. None of these

8. 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

9. All object have prototypes.

- a. True
- b. False

10. What's the output?

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

```
sum(1, '2');
```

- a. NaN
- b. TypeError
- c. "12"
- d. 3

JavaScript Model Test 01

11. 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

12. 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

13. What's the output?

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

```
getAge(21);
```

- a. "numbe"
- b. "array"
- c. "object"
- d. "NaN"

14. What's the output?

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

```
getAge();
```

- a. 21
- b. undefined
- c. ReferenceError
- d. TypeError

15. What's the value of sum?

```
const sum = eval('10*10+5');
```

- a. 105
- b. "105"
- c. TypeError
- d. "10*10+5"

16. How long is cool_secret accessible?

```
sessionStorage.setItem('cool_secret', 123);
```

- a. Forever, the data doesn't get lost
- b. When the user closes the tab.

- c. When the user closes the entire browser, not only the tab.
- d. When the user shuts off their computer.

17. What's the output?

```
var num = 8;
var num = 10;
```

```
console.log(num);
```

- a. 8
- b. 10
- c. SyntaxError
- d. ReferenceError

18. What's the output?

```
const obj = { 1: 'a', 2: 'b', 3: 'c' };
const set = new Set([1, 2, 3, 4, 5]);
obj.hasOwnProperty('1');
obj.hasOwnProperty(1);
set.has('1');
set.has(1);
```

- a. false true false true
- b. false true true true
- c. true true false true
- d. true true true true

19. What's the output?

```
const obj = { a: 'one', b: 'two', a: 'three' };
console.log(obj);
```

- a. { a: "one", b: "two" }
- b. { b: "two", a: "three" }
- c. { a: "three", b: "two" }
- d. SyntaxError

20. The JavaScript global execution context creates two things for you: the global object, and the "this" keyword.

- a. True
- b. false
- c. it depends

21. What's the output?

```
for (let i = 1; i < 5; i++) {
  if (i === 3) continue;
  console.log(i);
}
```

- a. 1 2a
- b. 1 2 3
- c. 1 2 4
- d. 1 3 4

22. What's the output?

```
const a = {};
const b = { key: 'b' };
const c = { key: 'c' };
```

```
a[b] = 123;
a[c] = 456;
console.log(a[b]);
```

- a. 123
- b. 456
- c. undefined
- d. ReferenceError

23. What's the output?

```
const a = {};
const b = { key: 'b' };
const c = { key: 'c' };
```

JavaScript
Model Test 01

```
a[b] = 123;  
a[c] = 456;  
console.log(a[b]);
```

- a. 123
- b. 456
- c. Undefined
- d. ReferenceError

24. What is the event.target when clicking the button?

```
<div onclick="console.log('first div')">  
  <div onclick="console.log('second div')">  
    <button onclick="console.log('button')">  
      Click!  
    </button>  
  </div>  
</div>
```

- a. Outer div
- b. Inner div
- c. Button
- d. An array of all nested elements.

25. When you click the paragraph, what's the logged output?

```
<div onclick="console.log('div')">  
  <p onclick="console.log('p')">  
    Click here!  
  </p>  
</div>
```

- a. p div
- b. div p
- c. p
- d. div

26. Everything in JavaScript is either a...

- a. primitive or object
- b. function or object
- c. trick question! only objects
- d. number or object

27. What's the output?

```
console.log(3 + 4 + '5');
```

- a. "345"
- b. "75"
- c. 12
- d. "12"

28. What's the value of num?

```
const num = parseInt('7*6', 10);
```

- a. 42
- b. "42"
- c. 7
- d. NaN

29. What's the output?

```
const set = new Set([1, 1, 2, 3, 4]);  
console.log(set);
```

- a. [1, 1, 2, 3, 4]
- b. [1, 2, 3, 4]
- c. {1, 1, 2, 3, 4}
- d. {1, 2, 3, 4}

30. Is this a pure function?

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

- a. Yes
- b. No