Time left 0:19:56

Question 1

Not yet answered

Marked out of 1.00

What is the output of the following code?

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

- a.   ReferenceError
- b.   20
- c.   undefined
- d.   10

Question 2

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
const obj = { a: 1 };
const ref = obj;
ref.a = 5;
console.log(obj.a);
```

- a.   undefined
- b.   5
- c.   TypeError
- d.   1

Question 3

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
function test() {

    console.log(this);
}
test();
```

- a.   function test
- b.   undefined
- c.   null
- d.   window object (or global object)

Question 4

Not yet answered

Marked out of 1.00

What is the output of the following code?

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

- ○ a.   Error
- ○ b.   3
- ○ c.   undefined
- ○ d.   2

Question 5

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
let count = 0;
const inc = () => count++;
inc();
inc();
console.log(count);
```

- ○ a.   0
- ○ b.   2
- ○ c.   1
- ○ d.   undefined

Question 6

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
async function f() {
    return 10;
}
f().then(console.log);
```

- ○ a.   Error
- ○ b.   10
- ○ c.   undefined
- ○ d.   Promise

Question 7

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
function outer() {
    let x = 5;
    return function inner() {
        return x++;
    };
}
const fn = outer();
console.log(fn(), fn());
```

- a. undefined undefined
- b. 5 5
- c. 6 6
- d. 5 6

Question 8

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
try {
    throw "error";
} catch (e) {
    console.log(typeof e);
}
```

- a. "Error"
- b. "string"
- c. undefined
- d. "object"

Question 9

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
console.log([1,2,3].map(x => x * 2).filter(x => x > 3));
```

- a. [3,6]
- b. [4,6]
- c. [2,4,6]
- d. Error

Question 10

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
console.log(0 == false);
```

- ○ a. undefined
- ○ b. true
- ○ c. TypeError
- ○ d. false

Question 11

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
console.log(1 + "2" + 3);
```

- ○ a. 6
- ○ b. Error
- ○ c. "33"
- ○ d. "123"

Question 12

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
console.log(typeof (() => {}));
```

- ○ a. "arrow"
- ○ b. "undefined"
- ○ c. "object"
- ○ d. "function"

Question 13

Not yet answered

Marked out of 1.00

What is the output of the following code?

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

- ○ a. []
- ○ b. undefined
- ○ c. ""
- ○ d. TypeError

Question 14

Not yet answered

Marked out of 1.00

What is the output of the following code?

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

- ○ a. "object"
- ○ b. "NaN"
- ○ c. "undefined"
- ○ d. "number"

Question 15

Not yet answered

Marked out of 1.00

What will be the output of the following code?

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

- ○ a. "undefined"
- ○ b. "number"
- ○ c. "object"
- ○ d. "null"

Question 16

Not yet answered

Marked out of 1.00

Which data structure enforces unique values?

- ○ a. Map
- ○ b. Array
- ○ c. Set
- ○ d. Object

Question 17

Not yet answered

Marked out of 1.00

Which feature allows extracting values from arrays or objects?

- ○ a. Rest operator
- ○ b. Spread operator
- ○ c. Cloning
- ○ d. Destructuring

Question 18

Not yet answered

Marked out of 1.00

Which JavaScript feature enables data privacy through function scope?

- ○ a.   Hoisting
- ○ b.   Prototypes
- ○ c.   Closures
- ○ d.   Event delegation

Question 19

Not yet answered

Marked out of 1.00

Which JavaScript mechanism enables non-blocking behavior?

- ○ a.   Call stack
- ○ b.   Scope chain
- ○ c.   Heap
- ○ d.   Event loop

Question 20

Not yet answered

Marked out of 1.00

Which keyword creates a block-scoped variable?

- ○ a.   function
- ○ b.   prototype
- ○ c.   var
- ○ d.   const

Question 21

Not yet answered

Marked out of 1.00

Which keyword prevents reassignment of a variable?

- ○ a.   var
- ○ b.   const
- ○ c.   let
- ○ d.   static

Question 22

Not yet answered

Marked out of 1.00

Which method converts a JSON string into a JavaScript object?

○ a.   JSON.parse

○ b.   JSON.toObject

○ c.   JSON.stringify

○ d.   Object.assign

Question 23

Not yet answered

Marked out of 1.00

Which method is used to combine multiple arrays into one?

○ a.   merge

○ b.   push

○ c.   concat

○ d.   join

Question 24

Not yet answered

Marked out of 1.00

Which method schedules a callback in the microtask queue?

○ a.   Promise.then

○ b.   setInterval

○ c.   setTimeout

○ d.   requestAnimationFrame

Question 25

Not yet answered

Marked out of 1.00

Which operator checks both value and type?

○ a.   =

○ b.   ==

○ c.   ===

○ d.   !=

Question 26

Not yet answered

Marked out of 1.00

Which promise state represents successful completion?

- ○ a.   Rejected
- ○ b.   Settled
- ○ c.   Fulfilled
- ○ d.   Pending

Question 27

Not yet answered

Marked out of 1.00

Which statement about arrow functions is TRUE?

- ○ a.   They support arguments object
- ○ b.   They have their own this
- ○ c.   They inherit this from surrounding scope
- ○ d.   They can be used as constructors

Question 28

Not yet answered

Marked out of 1.00

Which statement about JavaScript hoisting is TRUE?

- ○ a.   Variables declared with let are hoisted and initialized
- ○ b.   Function expressions are hoisted completely
- ○ c.   var declarations are hoisted without initialization
- ○ d.   Class declarations are hoisted and usable before declaration

Question 29

Not yet answered

Marked out of 1.00

Which statement about Map is TRUE?

- ○ a.   Same as Object
- ○ b.   Keys must be strings
- ○ c.   Maintains insertion order
- ○ d.   Does not allow iteration

Question 30

Not yet answered

Marked out of 1.00

Which statement about prototypes is TRUE?

○ a.   Prototypes are immutable

○ b.   Each object has its own prototype

○ c.   Prototypes are only for classes

○ d.   Prototypes enable inheritance