

MCQ QUESTION

JAVASCRIPT

Asynchronous JavaScript

Question 1: What does `async/await` do?

Answer: a) Simplifies working with promises

Question 2: What does `async/await` do? (Alternative phrasing)

Answer: b) Simplifies promise chains

Question 3: What is `callback hell` ?

Answer: b) Nested callbacks making code hard to read

Question 4: What is a promise in JavaScript?

Answer: b) An object representing eventual completion/failure

Question 5: What is the output of:

```
setTimeout(() => console.log(1), 0);  
Promise.resolve().then(() => console.log(2));  
console.log(3);
```

Answer: b) `3, 2, 1`

Question 6: What is the purpose of `fetch()` ?

Answer: a) To make HTTP requests

Arrays and Methods

Question 1: How do you check if an element exists in an array?

Answer: c) `array.includes()`

Question 2: How do you check if an object is an array?

Answer: c) `Array.isArray(obj)`

Question 3: What does `array.shift()` do?

Answer: a) Removes the first element

Question 4: What does `[1,2,3].reduce((a,b) => a + b)` return?

Answer: a) 6

Question 5: What does `[1, 2, 3].slice(1)` return?

Answer: b) `[2, 3]`

Question 6: What is the output of `[1, 2, 3].map(x => x * 2)` ?

Answer: b) `[2, 4, 6]`

Question 7: Which method adds an element to the end of an array?

Answer: a) `array.push()`

Question 8: Which method creates a new array with all sub-array elements concatenated?

Answer: c) `flat()`

Question 9: Which method removes the last element from an array?

Answer: b) `array.pop()`

Data Types and Operators

Question 1: What does `===` operator check?

Answer: c) Both value and type

Question 2: What is `NaN` in JavaScript?

Answer: b) A special value representing "Not a Number"

Question 3: What is the output of `console.log(2 + "2")` ?

Answer: b) `"22"`

Question 4: What is the output of `console.log(3 + 2 + "7")` ?

Answer: c) `"57"`

Question 5: What is the output of `console.log(3 > 2 > 1)` ?

Answer: b) `false`

Question 6: What is the output of `console.log(5 == "5")` ?

Answer: a) `true`

Question 7: What is the output of `console.log(typeof NaN)` ?

Answer: a) "number"

Question 8: What is the output of `console.log(typeof null)` ?

Answer: a) `"object"`

Question 9: Which operator has the highest precedence?

Answer: b) `()`

Question 10: Which operator returns the remainder of division?

Answer: b) `%`

ES6+ Features

Question 1: What does the spread operator (`...`) do?

Answer: a) Expands iterables into individual elements

Question 2: What is a `Set` in JavaScript?

Answer: a) A collection of unique values

Question 3: What is destructuring?

Answer: b) Unpacking values from objects/arrays

Question 4: What is the output of:

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

Answer: b) `2`

Question 5: What is the output of:

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

Answer: b) `2`

Question 6: What is the output of:

```
const obj = { a: 1 };  
const obj2 = { ...obj, b: 2 };  
console.log(obj2);
```

Answer: c) `{ a: 1, b: 2 }`

Question 7: Which keyword declares a block-scoped variable?

Answer: d) Both `let` and `const`

Functions and Scope

Question 1: What does the bind() method do?

Answer: a) Creates a new function with a specific this value

Question 2: What is `this` in a global function?

Answer: c) `window` (in browsers)

Question 3: What is `this` in an arrow function?

Answer: c) Inherits from parent scope

Question 4: What is a closure in JavaScript?

Answer: a) A function with access to its outer function's scope

Question 5: What is a pure function?

Answer: b) A function that always returns the same output for same inputs

Question 6: What is an arrow function missing that regular functions have?

Answer: c) Both a and b

Question 7: What is an arrow function's main difference from regular functions?

Answer: b) Cannot be used as constructors

Question 8: What is an IIFE?

Answer: b) Immediately Invoked Function Expression

Question 9: What is the output of:

```
function foo() { return arguments.length; }  
console.log(foo(1,2,3));
```

Answer: c) 3

Question 10: What is the output of:

```
function foo(a = 1, b = 2) {  
  return a + b;  
}  
console.log(foo(undefined, 3));
```

Answer: b) 4

Question 11: What is the output of:

```
function greet() { return "Hello"; }  
console.log(typeof greet());
```

Answer: b) "string"

Question 12: What is the output of:

```
function test() { return this; }  
console.log(test() === window);
```

Answer: a) `true` (in browser)

Question 13: What is the output of:

```
console.log(((() => "Hello"))());
```

Answer: a) `"Hello"`

Question 14: What is the output of:

```
console.log(typeof foo);  
function foo() { return "Hello"; }
```

Answer: a) `"function"`

Question 15: What is the output of:

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

Answer: b) 10

Question 16: What is the output of:

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

Answer: d) ReferenceError

Objects and Prototypes

Question 1: How do you access a property named "first-name" in an object?

Answer: b) obj['first-name']

Question 2: How do you check if a property exists in an object?

Answer: c) "key" in obj

Question 3: How do you clone an object obj shallowly?

Answer: b) { ...obj }

Question 4: How do you create an object in JavaScript?

Answer: a) let obj = {};

Question 5: What does Object.freeze() do?

Answer: a) Makes an object immutable

Question 6: What is prototypal inheritance?

Answer: a) Objects inherit from other objects

Question 7: What is the output of:

```
const obj = { a: 1 };
Object.freeze(obj);
```

```
obj.a = 2;  
console.log(obj.a);
```

Answer: a) 1

Question 8: What is the output of:

```
console.log(Object.keys({ a: 1, b: 2 }));
```

Answer: a) ["a", "b"]

Question 9: What is the output of:

```
let obj = { a: 1 };  
let obj2 = obj;  
obj2.a = 2;  
console.log(obj.a);
```

Answer: b) 2

Strict Mode

Question 1: What does "use strict" do?

Answer: a) Enforces stricter parsing and error handling

Type Coercion and Truthiness

Question 1: What is the output of `console.log(!!"false")` ?

Answer: a) true

Variables and Scoping

Question 1: What is the output of:

```
console.log(a);  
var a = 5;
```

Answer: b) undefined

Question 2: What is the output of:

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

Answer: c) undefined

Question 3: What is the scope of `let` ?

Answer: b) Block scope

Question 4: Which keyword creates a constant that cannot be reassigned?

Answer: c) const

Node.js Questions

Core Concepts

Question 1: How do you install a Node.js package locally?

Answer: a) `npm install package-name`

Question 2: What is the purpose of the cluster module?

Answer: b) To create child processes

Question 3: What is the purpose of `require()` in Node.js?

Answer: a) To include modules

Question 4: Which global object is used in Node.js for timing functions?

Answer: b) `global`

Question 5: Which global object provides information about the current process?

Answer: c) process

Question 6: Which method is used to parse query strings?

Answer: a) `querystring.parse()`

Question 7: What does `require.cache` contain?

Answer: a) Currently loaded modules

Core Modules

Question 1: What does `require('path').join()` do?

Answer: a) Joins path segments

Question 2: What is `__dirname` in Node.js?

Answer: b) Current directory path

Question 3: What is `process.env` used for?

Answer: b) Access environment variables

Question 4: Which method starts a Node.js server?

Answer: b) `server.listen()`

Question 5: Which module creates a server?

Answer: a) `http`

Question 6: Which module handles HTTP requests?

Answer: a) `http`

Question 7: Which module is used for file operations?

Answer: b) `fs`

Question 8: Which module is used to create a web server?

Answer: a) `http`

Question 9: Which module is used to handle file operations in Node.js?

Answer: b) `fs`

Event Loop and Asynchronous Operations

Question 1: What does `stream.pipe()` do?

Answer: b) Directs data from one stream to another

Question 2: What is `Buffer` in Node.js?

Answer: a) Handles binary data

Question 3: What is `process.nextTick()` ?

Answer: b) Executes a callback before the next event loop iteration

Question 4: What is `setImmediate()` ?

Answer: b) Runs after I/O events

Question 5: What is the Node.js event loop?

Answer: a) Handles async operations

Question 6: What is the output of:

```
setTimeout(() => console.log(1), 0);  
setImmediate(() => console.log(2));
```

Answer: c) Order is not guaranteed

Question 7: What does `process.argv` contain?

Answer: a) Command-line arguments

Events

Question 1: What is the output of:

```
const EventEmitter = require('events');  
const emitter = new EventEmitter();  
emitter.on('event', () => console.log(1));  
emitter.prependListener('event', () => console.log(2));  
emitter.emit('event');
```

Answer: b) 2 1

Express.js

Question 1: How do you define a route in Express?

Answer: a) `app.get('/', (req, res) => { ... })`

Question 2: How do you handle errors in Express?

Answer: c) `app.use((err, req, res, next) => { ... })`

Question 3: What does `res.send()` do?

Answer: a) Sends a response

Question 4: What is Express.js?

Answer: a) A web framework for Node.js

Question 5: What is middleware in Express?

Answer: b) Functions that process requests

File System

Question 1: What does `fs.readdir()` do?

Answer: b) Lists directory contents

Question 2: What does `fs.writeFile()` do?

Answer: b) Overwrites a file

Question 3: What is the purpose of `__dirname`?

Answer: b) Gets the current directory path

Question 4: Which method appends to a file asynchronously?

Answer: b) `fs.appendFile()`

Question 5: Which method reads a file asynchronously?

Answer: a) `fs.readFile()`

Question 6: Which method reads a file synchronously?

Answer: b) `fs.readFileSync()`

Modules

Question 1: What is the purpose of `module.exports`?

Answer: b) To export functionality from a module

Question 2: Which function is used to import ES modules in Node.js?

Answer: b) `import()`

MongoDB Questions

Aggregation

Question 1: What does \$group do?

Answer: a) Groups documents by specified identifier

Question 2: Which pipeline stage filters documents?

Answer: b) \$match

Basics and Fundamentals

Question 1: What is MongoDB?

Answer: a) A NoSQL database

Question 2: What is the default MongoDB port?

Answer: a) 27017

CRUD Operations

Question 1: What does db.collection.deleteMany() return?

Answer: c) A document with operation stats

Question 2: Which command inserts a document?

Answer: a) `db.collection.insertOne()`

Question 3: Which command inserts a document in MongoDB?

Answer: a) `db.collection.insertOne()`

Question 4: Which method updates multiple documents in MongoDB?

Answer: b) `updateMany()`

Indexes

Question 1: What is the purpose of createIndex()?

Answer: b) To improve query performance

Querying and Finding Documents

Question 1: What does `db.collection.countDocuments()` do?

Answer: a) Returns the number of documents

Question 2: What does `db.collection.find()` return?

Answer: c) A cursor object

Question 3: Which projection operator excludes fields?

Answer: d) `$unset`

Query Operators

Question 1: What does `$regex` do?

Answer: a) Performs pattern matching

Question 2: What is the purpose of `$in` operator?

Answer: a) Checks if value is in array

Question 3: Which operator checks equality?

Answer: a) `$eq`

Question 4: Which operator checks if a field exists?

Answer: a) `$exists`

Question 5: Which operator is used for "not equal" in MongoDB?

Answer: b) `$ne`

Question 6: Which operator performs a logical AND?

Answer: a) `$and`

Update Operations

Question 1: What does `$inc` do?

Answer: a) Increments a field

Question 2: Which operator sets a field value?

Answer: a) `$set`

Programming Questions

Arithmetic and Mathematical Operations

Question 1: What is the output of:

```
console.log(+ "1" - - "1");
```

Answer: b) 2

Question 2: What is the output of:

```
let x = 1;  
console.log(x++ + x++);
```

Answer: b) 3

Question 3: What is the output of:

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

Answer: c) 12

Array and Object Behavior

Question 1: What does `console.log([] == ![])` return?

Answer: a) `true`

Question 2: What does `console.log(typeof [])` print?

Answer: b) `"object"`

Question 3: What is the output of:

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

Answer: b) 2

Boolean and Truthiness Operations

Question 1: What is the output of:

```
console.log(!"false" === !!"true");
```

Answer: a) `true`

Floating Point Precision

Question 1: What is the output of `console.log(0.1 + 0.2 === 0.3)` ?

Answer: b) `false`

String Concatenation and Type Coercion

Question 1: What is the output of:

```
console.log("b" + "a" + +"a" + "a");
```

Answer: b) `"baNaN"`

Question 2: What is the output of:

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

Answer: b) `"21"`

Question 3: What is the output of:

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

Answer: b) `"21"`

Type Checking and Conversion

Question 1: What is the output of `console.log(5 == "5")` ?

Answer: a) `true`

Question 2: What is the output of:

```
console.log(typeof typeof 1);
```

Answer: b) `"string"`

HTML Questions

Basic HTML Elements and Attributes

Question 1: What does `
` tag do?

Answer: b) Creates a line break

Question 2: What is the correct way to include JavaScript in HTML?

Answer: a) `<script src="file.js"></script>`

Question 3: Which attribute specifies an image source?

Answer: a) `src`

Question 4: Which tag defines a table row?

Answer: a) `<tr>`

Question 5: Which tag is used for a hyperlink?

Answer: a) `<a>`

CSS Layout (HTML-related)

Question 1: What does `margin: auto` do?

Answer: a) Centers horizontally in flex/grid

Question 2: Which property creates a flex container?

Answer: b) `display: flex`

Forms and Input Elements

Question 1: What does `<fieldset>` do?

Answer: a) Groups form elements

Question 2: What does the `<label>` tag do?

Answer: c) Both a and b

Question 3: Which input type creates a checkbox?

Answer: b) `<input type="checkbox">`

Question 4: Which input type validates an email?

Answer: a) `<input type="email">`

Multimedia Elements

Question 1: What does the poster attribute do in `<video>` ?

Answer: a) Sets the thumbnail image

Question 2: Which tag embeds a video?

Answer: b) `<video>`

Semantic HTML Elements

Question 1: What does `<article>` represent?

Answer: a) A blog post

Question 2: Which tag represents navigation links?

Answer: a) `<nav>`

CSS Questions

Basic CSS Properties and Values

Question 1: What does `display: none;` do?

Answer: b) Removes an element completely

Question 2: What does `padding` do?

Answer: b) Adds space inside an element

Question 3: Which CSS unit is relative to font size?

Answer: b) `em`

Question 4: Which property adds space between border and content?

Answer: b) `padding`

Question 5: Which property changes text color?

Answer: c) `color`

CSS Layout and Positioning

Question 1: What does `position: absolute` do?

Answer: b) Positions relative to nearest positioned ancestor

Question 2: What does `position: relative` do?

Answer: b) Positions relative to normal flow

Question 3: Which property controls element stacking order?

Answer: c) `z-index`

CSS Selectors

Question 1: How do you select an element with ID "header"?

Answer: a) `#header`

Question 2: Which selector matches all `<p>` elements inside a `<div>` ?

Answer: a) `div p`

CSS Transforms and Animations

Question 1: What does `animation-fill-mode` do?

Answer: a) Specifies styles before/after animation

Question 2: Which property creates 3D transformations?

Answer: b) `perspective`

