

# BÀI TẬP CÁC CÂU HỎI PHÒNG VẤN

*(các câu hỏi dưới đây được tổng hợp từ nhiều nguồn)*

## ***JavaScript***

1. JavaScript là gì và khác gì so với Java?

.....

.....

2. JS là ngôn ngữ thông dịch hay biên dịch?

.....

.....

3. var, let, const khác nhau như thế nào?

.....

.....

4. Khi nào dùng let thay vì var?

.....

.....

5. Kiểu dữ liệu nguyên thủy (primitive types) gồm những gì?

.....

.....

6. null và undefined khác nhau ra sao?

.....

.....

7. Toán tử typeof trả về gì cho mỗi kiểu dữ liệu?

.....

.....

8. Giải thích Truthy và Falsy values.

.....

.....

9. So sánh `==` và `===` khác nhau thế nào?

.....

.....

10. Kết quả của `"5" + 2` và `"5" - 2` là gì, tại sao?

.....

.....

11. Cách chuyển đổi chuỗi thành số trong JS?

.....

.....

12. NaN là gì và có thể kiểm tra bằng cách nào?

.....

.....

13. Giải thích hoisting là gì.

.....

14. Tại sao nên tránh dùng `var` trong ES6 trở đi?

.....

.....

15. Khi nào thì một biến là undefined?

.....

.....

16. Toán tử typeof null trả về gì, vì sao?

.....

.....

17. Toán tử logic || và && trong JS hoạt động thế nào?

.....

.....

18. Giải thích cơ chế short-circuit evaluation.

.....

.....

19. Toán tử ba ngôi (?:) dùng khi nào?

.....

.....

20 switch và if/else khác nhau gì về hiệu suất?

.....

.....

21. Function declaration và function expression khác nhau thế nào?

.....

.....

22. Arrow function là gì, ưu/nhược điểm?

.....

.....

23. Khi nào không nên dùng arrow function?

.....

.....

24. Từ khóa return có tác dụng gì?

.....

.....

25. Nếu một hàm không có return thì trả về gì?

.....

.....

26. Tham số mặc định (default parameters) là gì?

.....

.....

27. arguments object là gì?

28. Sự khác biệt giữa `call()`, `apply()` và `bind()`?

.....

.....

29. Closure là gì? Cho ví dụ.

.....

.....

30. Giải thích khái niệm lexical scope.

.....

.....

31. Higher-order function là gì?

.....

.....

32. Callback function là gì?

.....

.....

33. Tại sao callback hell lại xảy ra?

.....

.....

34. đệ quy (recursion) là gì?

.....

.....

35. Có thể gọi hàm trước khi khai báo không? Tại sao?

.....

.....

36. Pure function là gì?

.....

.....

37. Function hoisting hoạt động như thế nào?

.....

.....

38. IIFE là gì (Immediately Invoked Function Expression)?

.....

.....

39. Khi nào nên dùng IIFE?

.....

.....

40. Arrow function có this riêng không?

.....

41. Object trong JS là gì?

.....

.....

42. Prototype là gì?

.....

.....

43. Prototype chain hoạt động như thế nào?

.....

.....

44. Tại sao nói “Everything in JS is an object”?

.....

.....

45. Cách thêm và xóa thuộc tính trong object?

.....

.....

46. `hasOwnProperty()` dùng để làm gì?

.....

.....

47. `Object.keys()`, `Object.values()` và `Object.entries()` khác nhau thế nào?

48. Khi nào dùng `Object.freeze()` và `Object.seal()`?

.....

.....

49. So sánh `Object.assign()` và spread operator `{...obj}`.

.....

.....

50. Làm sao kiểm tra một biến có phải là array?

.....

.....

51. `map()`, `filter()`, `reduce()` khác nhau như thế nào?

.....

.....

52. `find()` và `findIndex()` khác nhau thế nào?

.....

.....

53. Khi nào nên dùng `reduce()` thay vì `for`?

.....

.....

54. `every()` và `some()` dùng để làm gì?

.....



.....

55. `sort()` trong JS có gì đặc biệt?

.....

.....

56. Class trong ES6 là gì?

.....

.....

57. Constructor trong class dùng để làm gì?

.....

.....

58. Kế thừa (extends) trong class hoạt động ra sao?

.....

.....

59. `super()` trong class có vai trò gì?

.....

.....

60. Static method là gì?

.....

61. DOM là gì?

.....

.....

62. Cách truy cập phần tử trong DOM (querySelector, getElementById, ...)?

.....

.....

63. innerText, textContent, innerHTML khác nhau thế nào?

.....

.....

64. Cách thêm phần tử mới vào DOM?

.....

.....

65. append() khác gì appendChild()?

.....

.....

66. addEventListener() là gì?

.....

.....

67. Event bubbling và capturing là gì?

68. Event delegation là gì?

.....

.....

69. `preventDefault()` và `stopPropagation()` khác nhau ra sao?

.....

.....

70. Cách lấy value từ input trong form bằng JS?

.....

.....

71. `this` trong event handler trỏ đến gì?

.....

.....

72. Cách gỡ bỏ event listener?

.....

.....

73. `classList` dùng làm gì?

.....

.....

74. Khi nào nên thao tác DOM bằng JS thuần, khi nào nên dùng framework?

.....

.....

75. dataset trong DOM là gì?

.....

.....

76. Đồng bộ (synchronous) và bất đồng bộ (asynchronous) khác nhau thế nào?

.....

.....

77. Callback hell là gì?

.....

.....

78. Promise là gì?

.....

.....

79. Các trạng thái của Promise?

.....

.....

80. Promise chaining là gì?

.....

.....

81. Promise.all() hoạt động ra sao?

.....

.....

82. Promise.race() khác gì với Promise.all()?

.....

.....

83. async và await là gì?

.....

.....

84. Khi nào nên dùng await?

.....

.....

85. Cách bắt lỗi trong async/await?

.....

.....

86. Event loop là gì?

.....

.....

87. Microtask và Macrotask khác nhau thế nào?

88. `setTimeout` và `setInterval` khác nhau ra sao?

.....

.....

89. Cách hủy `setInterval`?

.....

.....

90. Khi nào cần dùng `API Fetch`?

.....

.....

91. Module trong JS là gì?

.....

.....

92. `import` và `export` khác nhau thế nào?

.....

.....

93. Optional chaining (`?.`) là gì?

.....

.....

94. Nullish coalescing (`??`) khác `||` thế nào?

.....

.....

95. Babel là gì và dùng để làm gì?

.....

.....

96. Polyfill là gì?

.....

.....

97. Strict mode là gì?

.....

.....

98. Currying là gì?

.....

.....

99. Debounce và Throttle khác nhau thế nào?

.....

.....

100. Giải thích cơ chế single-thread trong JS.

