**✅ BỘ CÂU HỎI TRẮC NGHIỆM PHP – MySQL**

**1. Hàm nào sau đây dùng để kết nối đến cơ sở dữ liệu MySQL trong PHP?**

A. mysql\_open()  
B. mysqli\_connect() ✅  
C. connect\_db()  
D. db\_connect()

**2. Hàm mysqli\_connect() có bao nhiêu tham số bắt buộc?**

A. 2  
B. 3  
C. 4 ✅  
D. 5

**3. Cách kiểm tra kết nối MySQL có thành công không là?**

A. if (!$conn)  
B. if (mysqli\_error())  
C. if (mysqli\_connect\_errno()) ✅  
D. if ($conn == null)

**4. Hàm nào dùng để thực thi câu truy vấn SQL trong PHP?**

A. execute\_sql()  
B. mysqli\_query() ✅  
C. run\_sql()  
D. mysqli\_exec()

**5. mysqli\_query() trả về gì nếu là truy vấn SELECT?**

A. Số lượng dòng  
B. Mảng dữ liệu  
C. Đối tượng kết quả ✅  
D. Chuỗi dữ liệu

**6. Hàm nào dùng để lấy dữ liệu từ kết quả truy vấn?**

A. mysqli\_fetch()  
B. mysqli\_get\_row()  
C. mysqli\_fetch\_assoc() ✅  
D. mysqli\_result()

**7. Kết quả của mysqli\_fetch\_assoc() là gì?**

A. Mảng số  
B. Mảng kết hợp ✅  
C. Chuỗi JSON  
D. Đối tượng

**8. Để đếm số dòng trả về, dùng hàm nào?**

A. mysqli\_count\_rows()  
B. mysqli\_num\_rows() ✅  
C. count()  
D. get\_row\_count()

**9. Hàm nào giúp ngăn SQL Injection?**

A. mysqli\_block\_sql()  
B. mysqli\_secure()  
C. mysqli\_real\_escape\_string() ✅  
D. secure\_input()

**10. Tại sao nên dùng prepared statements?**

A. Viết code ngắn hơn  
B. Dễ debug  
C. Bảo mật chống SQL Injection ✅  
D. Không cần kết nối CSDL

**11. Lệnh nào dưới đây dùng để đóng kết nối MySQL?**

A. close\_db()  
B. mysqli\_disconnect()  
C. mysqli\_close() ✅  
D. stop\_connection()

**12. Hàm nào trả về lỗi kết nối nếu có?**

A. mysqli\_error() ✅  
B. error\_msg()  
C. mysqli\_fail()  
D. get\_error()

**13. Khi nào nên dùng mysqli\_close()?**

A. Trước khi kết nối  
B. Sau khi kết nối  
C. Sau khi truy vấn xong ✅  
D. Không cần thiết dùng

**14. mysqli\_fetch\_array() trả về gì?**

A. Mảng kết hợp  
B. Mảng số  
C. Cả A và B ✅  
D. Đối tượng

**15. Đoạn nào sau là truy vấn đúng?**

A. mysqli\_query("SELECT \* FROM table")  
B. mysqli\_query($sql)  
C. mysqli\_query($conn, $sql) ✅  
D. query($sql)

**16. Cổng mặc định MySQL là bao nhiêu?**

A. 80  
B. 3306 ✅  
C. 8080  
D. 443

**17. Biểu mẫu HTML gửi dữ liệu bằng cách nào?**

A. URL  
B. Cookie  
C. <form method="POST"> ✅  
D. PHP Function

**18. Hàm nào dùng để lấy giá trị từ form gửi lên PHP?**

A. $\_DATA  
B. $\_FORM  
C. $\_GET và $\_POST ✅  
D. $\_REQUESTED

**19. Tên biến POST nên được kiểm tra bằng gì?**

A. isset() ✅  
B. exists()  
C. check()  
D. not\_null()

**20. Đoạn nào sau đây tạo bảng HTML từ dữ liệu?**

A. echo "<tr><td>$row</td></tr>"; ✅  
B. echo $row;  
C. mysqli\_fetch\_row();  
D. print\_r($row);

**21. Hàm nào trả về ID mới chèn bằng INSERT?**

A. mysqli\_insert\_id() ✅  
B. mysqli\_last\_id()  
C. insert\_id()  
D. get\_id()

**22. Thư viện mysqli có hỗ trợ hướng đối tượng không?**

A. Không  
B. Có ✅  
C. Chỉ một phần  
D. Tùy cấu hình

**23. Hàm htmlspecialchars() giúp ngăn gì?**

A. SQL Injection  
B. Cookie giả mạo  
C. Cross-site scripting (XSS) ✅  
D. Tràn bộ nhớ

**24. Dữ liệu nhập từ người dùng nên kiểm tra bằng gì trước khi xử lý?**

A. echo  
B. trim  
C. filter\_input hoặc escape ✅  
D. connect()

**25. Tập tin nào thường dùng để lưu thông tin kết nối DB?**

A. dbinfo.html  
B. connect.php  
C. config.php ✅  
D. main.php

**26. Dữ liệu MySQL nên hiển thị bằng thẻ nào trong HTML?**

A. <div>  
B. <p>  
C. <table> ✅  
D. <span>

**27. Hàm mysqli\_prepare() thuộc chức năng nào?**

A. Đóng kết nối  
B. Kết nối  
C. Truy vấn thường  
D. Prepared statements ✅

**28. Hàm nào dùng để bind dữ liệu cho câu lệnh chuẩn bị?**

A. mysqli\_bind\_param()  
B. mysqli\_stmt\_bind\_param() ✅  
C. bind\_param()  
D. mysqli\_bind()

**29. Khi dùng mysqli\_fetch\_assoc(), bạn cần truyền vào gì?**

A. Kết nối  
B. Chuỗi SQL  
C. Biến $result ✅  
D. Tên cột

**30. Đoạn mã nào sau đây là đúng khi kết nối CSDL?**

A. mysqli\_connect("localhost")  
B. mysqli\_connect("localhost", "user", "pass", "db") ✅  
C. connect\_db()  
D. open\_sql("db")

**31. Biến $conn thường dùng trong PHP để làm gì?**

A. Lưu truy vấn SQL  
B. Lưu dữ liệu người dùng  
C. Lưu thông tin kết nối CSDL ✅  
D. Lưu kết quả truy vấn

**32. Kết quả trả về của mysqli\_query() với câu INSERT là gì?**

A. Object  
B. TRUE ✅  
C. FALSE  
D. Mảng

**33. Hàm mysqli\_error($conn) trả về gì?**

A. Chuỗi lỗi nếu có ✅  
B. Mảng lỗi  
C. NULL  
D. Mã lỗi

**34. Để hiển thị dữ liệu trong vòng lặp, bạn nên dùng:**

A. echo $result;  
B. echo $row['cot']; ✅  
C. print\_r($conn);  
D. mysqli\_show();

**35. mysqli\_stmt\_bind\_param($stmt, "s", $ten) – ký tự "s" nghĩa là gì?**

A. Số nguyên  
B. Chuỗi ✅  
C. Boolean  
D. Null

**36. Ký tự "i" trong bind\_param biểu thị kiểu gì?**

A. int ✅  
B. input  
C. index  
D. invalid

**37. Lỗi SQL Injection thường xảy ra ở đâu?**

A. Trong vòng lặp  
B. Khi in dữ liệu  
C. Khi người dùng nhập dữ liệu trực tiếp vào truy vấn ✅  
D. Khi kết nối DB

**38. Để tránh lỗi bảo mật, ta không nên:**

A. Sử dụng $\_GET cho thông tin nhạy cảm ✅  
B. Kiểm tra dữ liệu đầu vào  
C. Dùng bind\_param  
D. Escape dữ liệu đầu vào

**39. Nếu không xử lý $\_POST['email'] đúng cách, điều gì có thể xảy ra?**

A. Lỗi cú pháp PHP  
B. Gửi mail thất bại  
C. SQL Injection hoặc lỗi logic ✅  
D. Vòng lặp vô hạn

**40. Trong PHP, phương thức nào an toàn hơn để gửi dữ liệu?**

A. GET  
B. POST ✅  
C. SESSION  
D. COOKIE

**41. Hàm nào trả về cả tên cột lẫn chỉ số khi lấy dữ liệu?**

A. mysqli\_fetch\_array() ✅  
B. mysqli\_fetch\_assoc()  
C. mysqli\_fetch\_row()  
D. fetch\_all()

**42. mysqli\_fetch\_row() trả về:**

A. Mảng liên kết  
B. Mảng chỉ số số ✅  
C. Đối tượng  
D. JSON

**43. Câu lệnh nào tạo bảng mới trong SQL?**

A. INSERT  
B. NEW TABLE  
C. CREATE TABLE ✅  
D. MAKE TABLE

**44. Câu lệnh SQL nào cập nhật dữ liệu?**

A. SELECT  
B. UPDATE ✅  
C. INSERT  
D. MODIFY

**45. Câu lệnh SQL nào xóa dữ liệu?**

A. DELETE ✅  
B. REMOVE  
C. ERASE  
D. CUT

**46. $\_POST['ten'] sẽ lỗi nếu:**

A. Biến chưa tồn tại ✅  
B. Dữ liệu trống  
C. Không kết nối DB  
D. Biến là số

**47. Phương thức nào được dùng khi truyền biến trên thanh địa chỉ?**

A. POST  
B. GET ✅  
C. PUT  
D. SEND

**48. Khi dùng mysqli\_num\_rows(), biến đầu vào là:**

A. Câu lệnh SQL  
B. Kết nối DB  
C. Đối tượng kết quả ✅  
D. Mảng dữ liệu

**49. Chuỗi kết nối CSDL thường đặt trong file nào?**

A. connect.html  
B. login.php  
C. db.php ✅  
D. index.html

**50. mysqli\_stmt\_execute($stmt) dùng để:**

A. Gán biến  
B. Chạy câu SQL đã chuẩn bị ✅  
C. Đóng kết nối  
D. Hiển thị kết quả

**51. Để xử lý file form.php khi bấm submit, ta cần:**

A. Viết trong cùng file  
B. Dùng form action="form.php" ✅  
C. Dùng JavaScript  
D. Sử dụng cookie

**52. Trong PHP, dùng hàm gì để chuyển đổi ký tự HTML?**

A. htmlentities()  
B. htmlspecialchars() ✅  
C. html\_encode()  
D. escape\_html()

**53. Tại sao nên escape dữ liệu trước khi đưa vào HTML?**

A. Giảm dung lượng  
B. Chống XSS ✅  
C. Làm đẹp giao diện  
D. Đỡ lỗi SQL

**54. Để tránh lỗi khi dùng $\_POST, bạn nên:**

A. Kiểm tra bằng isset() ✅  
B. In trực tiếp  
C. Dùng var\_dump()  
D. Không làm gì

**55. $\_POST là biến gì trong PHP?**

A. Biến hệ thống  
B. Biến cục bộ  
C. Biến toàn cục ✅  
D. Biến tạm thời

**56. Kết nối với CSDL thất bại thường do:**

A. PHP sai phiên bản  
B. Mật khẩu sai ✅  
C. Không có file PHP  
D. Tên biến trùng

**57. Biến trong SQL thường phân biệt:**

A. Dấu gạch dưới  
B. Chữ HOA và chữ thường ✅  
C. Số và chữ  
D. Ký tự đặc biệt

**58. $\_GET được dùng chủ yếu trong:**

A. Dữ liệu nhạy cảm  
B. Gửi mật khẩu  
C. Thanh địa chỉ URL ✅  
D. Tệp đính kèm

**59. Nếu truy vấn SELECT không trả về kết quả, mysqli\_fetch\_assoc() trả:**

A. FALSE ✅  
B. NULL  
C. Chuỗi rỗng  
D. 0

**60. mysqli\_insert\_id() chỉ hoạt động sau:**

A. SELECT  
B. UPDATE  
C. INSERT ✅  
D. DELETE

**61. Tại sao nên đóng kết nối sau khi thao tác xong với CSDL?**

A. Để xóa dữ liệu  
B. Giải phóng tài nguyên ✅  
C. Tăng tốc độ truy vấn  
D. Bảo mật thông tin

**62. Hàm mysqli\_select\_db() dùng để làm gì?**

A. Kết nối CSDL  
B. Chọn cơ sở dữ liệu ✅  
C. Thực thi truy vấn  
D. Đóng kết nối

**63. Nếu không gọi mysqli\_close(), điều gì xảy ra?**

A. Mất dữ liệu  
B. PHP sẽ tự đóng sau khi script kết thúc ✅  
C. CSDL sẽ bị lỗi  
D. Giao diện không hiển thị

**64. Hàm mysqli\_real\_escape\_string() cần tham số nào?**

A. Chuỗi cần escape  
B. Kết nối và chuỗi ✅  
C. Câu SQL  
D. Không cần tham số

**65. Tên CSDL được khai báo trong tham số thứ mấy của mysqli\_connect()?**

A. 1  
B. 2  
C. 3  
D. 4 ✅

**66. Dữ liệu người dùng nên được kiểm tra khi nào?**

A. Sau khi lưu DB  
B. Sau khi truy vấn  
C. Trước khi xử lý ✅  
D. Khi hiển thị ra màn hình

**67. Kỹ thuật nào sau đây KHÔNG nên dùng?**

A. Escape chuỗi  
B. Kiểm tra input  
C. Gộp chuỗi truy vấn từ input người dùng ✅  
D. Prepared statement

**68. Nếu bạn muốn giới hạn số dòng kết quả, dùng lệnh SQL nào?**

A. STOP  
B. LIMIT ✅  
C. END  
D. COUNT

**69. Kết quả trả về của mysqli\_fetch\_all() là gì?**

A. Mảng 2 chiều ✅  
B. JSON  
C. Object  
D. Chuỗi

**70. isset($\_POST["user"]) có ý nghĩa gì?**

A. Kiểm tra biến tồn tại ✅  
B. So sánh chuỗi  
C. Hiển thị dữ liệu  
D. Xóa biến

**71. Biến $\_SESSION có thể lưu thông tin đăng nhập không?**

A. Không  
B. Có ✅  
C. Chỉ khi dùng cookie  
D. Tùy trình duyệt

**72. Để truyền dữ liệu giữa các trang, cách nào KHÔNG an toàn?**

A. $\_POST  
B. $\_SESSION  
C. $\_COOKIE ✅  
D. $\_FILES

**73. Phần mở rộng phổ biến của tập tin PHP là gì?**

A. .ph  
B. .php ✅  
C. .phtml  
D. .htaccess

**74. Câu lệnh SQL SELECT dùng để:**

A. Chèn dữ liệu  
B. Xóa dữ liệu  
C. Truy xuất dữ liệu ✅  
D. Cập nhật dữ liệu

**75. Trong PHP, để lặp qua mảng kết quả, ta dùng:**

A. while ✅  
B. if  
C. switch  
D. return

**76. mysqli\_fetch\_assoc() trả về gì nếu hết dữ liệu?**

A. NULL  
B. FALSE ✅  
C. Chuỗi rỗng  
D. 0

**77. Muốn hiển thị toàn bộ bảng HTML từ kết quả SELECT, cần:**

A. for  
B. while ✅  
C. goto  
D. do...until

**78. SQL Injection có thể dẫn đến:**

A. Mất dữ liệu ✅  
B. Giao diện xấu  
C. Lỗi HTML  
D. Không ảnh hưởng

**79. PHP xử lý dữ liệu với CSDL bằng mô hình nào?**

A. MVC  
B. Client-server ✅  
C. Peer-to-peer  
D. DNS-based

**80. Cấu trúc đúng khi lặp qua dữ liệu là:**

while ($row = \_\_\_\_\_\_\_\_\_) { }

A. mysqli\_query()  
B. mysqli\_num\_rows()  
C. mysqli\_fetch\_assoc($result) ✅  
D. fetch\_data()

**81. Để kiểm tra một biến có phải mảng hay không:**

A. is\_string()  
B. is\_array() ✅  
C. isset()  
D. array\_check()

**82. Khi viết câu SQL UPDATE, cần có phần nào?**

A. VALUES  
B. SET ✅  
C. INTO  
D. FROM

**83. Lệnh INSERT cần từ khóa nào?**

A. INTO ✅  
B. TO  
C. SET  
D. FETCH

**84. Sau khi submit form, biến $\_POST chứa dữ liệu từ:**

A. URL  
B. JavaScript  
C. input, textarea, select ✅  
D. cookie

**85. Sử dụng JavaScript để kiểm tra dữ liệu có bắt buộc không?**

A. Có  
B. Không ✅  
C. Bắt buộc nếu dùng POST  
D. Chỉ dùng khi có SQL

**86. mysqli\_stmt\_bind\_param($stmt, "si", $name, $age) nghĩa là:**

A. name là string, age là integer ✅  
B. age là string, name là integer  
C. cả 2 là string  
D. không hợp lệ

**87. mysqli\_fetch\_row() lấy dữ liệu theo:**

A. Chỉ tên cột  
B. Chỉ số ✅  
C. ID  
D. Không xác định

**88. htmlspecialchars() nên dùng ở đâu?**

A. Trước khi ghi DB  
B. Khi kết nối  
C. Khi in ra HTML ✅  
D. Khi chạy SQL

**89. Khi thực hiện INSERT, làm sao biết có thành công không?**

A. Kiểm tra chuỗi SQL  
B. Kiểm tra kết nối  
C. if (mysqli\_query(...)) ✅  
D. In ra SQL

**90. Để chuyển mảng thành chuỗi JSON:**

A. implode()  
B. array\_to\_string()  
C. json\_encode() ✅  
D. json\_parse()

**91. Hàm filter\_var() dùng để:**

A. Kiểm tra biến số  
B. Lọc và kiểm tra dữ liệu ✅  
C. Biến mảng thành chuỗi  
D. Gán biến

**92. mysqli\_fetch\_assoc() có trả về nhiều dòng không?**

A. Có  
B. Không, chỉ 1 dòng/lần gọi ✅  
C. Tùy cấu hình  
D. Chỉ trả về mảng rỗng

**93. SQL Injection dễ xảy ra khi dùng:**

A. mysqli\_query() với biến trực tiếp ✅  
B. Prepared statements  
C. PDO  
D. escape đầy đủ

**94. Kết nối CSDL thất bại có thể do:**

A. Câu SQL sai  
B. Chưa gọi mysqli\_query()  
C. Sai tài khoản hoặc CSDL không chạy ✅  
D. echo sai

**95. Để thêm dấu nháy đơn vào dữ liệu, dùng:**

A. addslashes() ✅  
B. stripslashes()  
C. htmlentities()  
D. bind\_param()

**96. Đâu là truy vấn SELECT đúng?**

A. SELECT ALL FROM table  
B. SELECT \* FROM table ✅  
C. GET \* FROM table  
D. PICK \* FROM table

**97. Biến $\_SERVER['REQUEST\_METHOD'] trả về gì?**

A. Loại trình duyệt  
B. Địa chỉ IP  
C. Phương thức gửi dữ liệu: GET hoặc POST ✅  
D. URL

**98. Nếu không xử lý form đúng cách, điều gì xảy ra?**

A. Lỗi trình duyệt  
B. Không gửi dữ liệu hoặc gây lỗi bảo mật ✅  
C. Lỗi CSDL  
D. Lỗi server

**99. mysqli\_num\_rows() thường dùng với câu lệnh nào?**

A. INSERT  
B. DELETE  
C. SELECT ✅  
D. UPDATE

**100. Dòng sau có nghĩa gì?**

$conn = mysqli\_connect("localhost", "root", "", "mydb");

A. Gán biến HTML  
B. Truy vấn SQL  
C. Kết nối đến MySQL database tên "mydb" ✅  
D. Tạo bảng mới

**HTML DOM**

**PHẦN 1: CƠ BẢN VỀ DOM**

1. **DOM viết tắt của từ gì?  
   A. Document of Model  
   B. Document Object Model  
   C. Data Object Model  
   D. Document Oriented Model  
   → Đáp án: B**
2. **DOM định nghĩa tài liệu như là:  
   A. Một tập hợp các hàm  
   B. Một danh sách các lớp  
   C. Một cấu trúc cây phân cấp các node  
   D. Một bảng dữ liệu  
   → Đáp án: C**
3. **DOM là chuẩn của tổ chức nào?  
   A. ISO  
   B. IEEE  
   C. ECMA  
   D. W3C  
   → Đáp án: D**
4. **Node đại diện toàn bộ trang HTML là:  
   A. element node  
   B. document node  
   C. text node  
   D. attribute node  
   → Đáp án: B**
5. **Tất cả mọi thứ trên trang HTML đều là gì?  
   A. Tag  
   B. Thuộc tính  
   C. Node  
   D. Chuỗi  
   → Đáp án: C**
6. **Node nào chứa nội dung văn bản?  
   A. text node  
   B. attribute node  
   C. element node  
   D. document node  
   → Đáp án: A**
7. **Thuộc tính innerHTML dùng để làm gì?  
   A. Tạo node mới  
   B. Gán style  
   C. Truy cập nội dung HTML bên trong một phần tử  
   D. Truy cập thuộc tính  
   → Đáp án: C**
8. **Giá trị của nodeName của text node là gì?  
   A. text  
   B. node  
   C. "#text"  
   D. innerText  
   → Đáp án: C**
9. **nodeValue của element node là:  
   A. Tên thẻ  
   B. Nội dung văn bản  
   C. undefined  
   D. ID của thẻ  
   → Đáp án: C**
10. **Có bao nhiêu loại node khác nhau trong DOM?  
    A. 8  
    B. 10  
    C. 12  
    D. 14  
    → Đáp án: C**

**🔸 PHẦN 2: TRUY CẬP VÀ THAO TÁC DOM**

1. **Phương thức getElementById("id") trả về:  
   A. Một danh sách node  
   B. Một node cụ thể có id tương ứng  
   C. Một chuỗi  
   D. Một mảng  
   → Đáp án: B**
2. **getElementsByTagName("tag") trả về:  
   A. Một chuỗi HTML  
   B. Một đối tượng NodeList  
   C. Một mảng JSON  
   D. Một đối tượng DOMString  
   → Đáp án: B**
3. **appendChild() dùng để:  
   A. Xóa node  
   B. Lấy nội dung  
   C. Thêm node con  
   D. Tìm node cha  
   → Đáp án: C**
4. **Để xóa node con khỏi node cha, dùng:  
   A. deleteNode()  
   B. removeElement()  
   C. removeChild()  
   D. deleteChild()  
   → Đáp án: C**
5. **Phương thức tạo phần tử mới:  
   A. createNode()  
   B. newElement()  
   C. document.createElement("tag")  
   D. tag.createElement()  
   → Đáp án: C**
6. **Phương thức chèn node vào trước một node con khác:  
   A. appendBefore()  
   B. insertBefore()  
   C. insertAfter()  
   D. prependChild()  
   → Đáp án: B**
7. **Thuộc tính childNodes trả về:  
   A. Một danh sách các node con  
   B. ID của node  
   C. Giá trị text  
   D. Tên node  
   → Đáp án: A**
8. **Thuộc tính firstChild trả về:  
   A. Node cha đầu tiên  
   B. Node con đầu tiên  
   C. Node kế tiếp  
   D. Node cuối cùng  
   → Đáp án: B**
9. **replaceChild() dùng để:  
   A. Xóa một node  
   B. Thay thế một node con bằng một node mới  
   C. Thêm node con  
   D. Lấy giá trị thuộc tính  
   → Đáp án: B**
10. **Phương thức trả về giá trị của thuộc tính là:  
    A. getAttribute()  
    B. valueOf()  
    C. attr()  
    D. getAttr()  
    → Đáp án: A**

**🔸 PHẦN 3: SCRIPTING VÀ SỰ KIỆN**

1. **Thuộc tính className dùng để:  
   A. Gán ID  
   B. Lấy nội dung HTML  
   C. Gán tên lớp CSS  
   D. Tạo node mới  
   → Đáp án: C**
2. **classList.add("className") dùng để:  
   A. Gỡ bỏ lớp  
   B. Kiểm tra lớp  
   C. Thêm lớp  
   D. Gán nội dung  
   → Đáp án: C**
3. **Cú pháp xử lý sự kiện trực tiếp trong HTML:  
   A. onclick = "functionName()"  
   B. onEvent = "javascript()"  
   C. addEvent("click")  
   D. ontrigger = "function"  
   → Đáp án: A**
4. **Phương thức dùng để gọi một hàm sau khoảng thời gian:  
   A. setTimer()  
   B. setInterval()  
   C. setTimeout()  
   D. startTimeout()  
   → Đáp án: C**
5. **Để hủy setTimeout, dùng:  
   A. stopTimeout()  
   B. clearTimeout()  
   C. cancelTimeout()  
   D. killTimeout()  
   → Đáp án: B**

**🔸 PHẦN 4: HTML COLLECTION VÀ TRUY XUẤT ĐẶC BIỆT**

1. **HTMLCollection không phải là:  
   A. Một kiểu mảng  
   B. Một object  
   C. Một DOM element  
   D. Một đối tượng có thể duyệt  
   → Đáp án: A**
2. **document.forms trả về:  
   A. Danh sách các thẻ form  
   B. Danh sách các input  
   C. Một HTML node  
   D. Danh sách các class  
   → Đáp án: A**
3. **document.images trả về:  
   A. Mảng các hình ảnh  
   B. Danh sách thẻ img  
   C. Một chuỗi src  
   D. Thẻ style  
   → Đáp án: B**
4. **document.links bao gồm:  
   A. Tất cả các <a> có href  
   B. Các liên kết hình ảnh  
   C. Mọi URL trên trang  
   D. Tất cả node script  
   → Đáp án: A**
5. **getElementsByName("x") trả về:  
   A. Một chuỗi  
   B. Node duy nhất  
   C. NodeList của phần tử có thuộc tính name=x  
   D. Một object JSON  
   → Đáp án: C**

**PHẦN 5: DOM NÂNG CAO & SCRIPTING**

1. **Phương thức hasChildNodes() trả về:  
   A. Số lượng node con  
   B. true nếu có ít nhất một node con  
   C. Một mảng node con  
   D. Tên các node con  
   → Đáp án: B**
2. **Thuộc tính nextSibling dùng để:  
   A. Lấy node cha  
   B. Lấy node con tiếp theo  
   C. Lấy node kế tiếp cùng cấp  
   D. Lấy node đầu tiên  
   → Đáp án: C**
3. **Để kiểm tra phần tử có thuộc tính cụ thể hay không, dùng:  
   A. getAttr()  
   B. checkAttribute()  
   C. hasAttribute()  
   D. existsAttribute()  
   → Đáp án: C**
4. **setAttribute("class", "blue") sẽ:  
   A. Tạo node class  
   B. Gán class có tên "blue"  
   C. Gán id là "blue"  
   D. Gán nội dung văn bản  
   → Đáp án: B**
5. **Để lặp qua NodeList, cách nào sau đây đúng nhất?  
   A. Dùng vòng lặp for in  
   B. Dùng map()  
   C. Dùng for với chỉ số  
   D. Không thể lặp  
   → Đáp án: C**
6. **Phương thức cloneNode(true) thực hiện:  
   A. Di chuyển node  
   B. Nhân bản node và tất cả node con  
   C. Nhân bản nội dung văn bản  
   D. Xóa node  
   → Đáp án: B**
7. **Khi gọi element.style.color = "red"; điều gì xảy ra?  
   A. Không có gì  
   B. Lỗi xảy ra  
   C. Màu văn bản thành đỏ  
   D. Thêm class có tên "red"  
   → Đáp án: C**
8. **Để lấy giá trị của một input, sử dụng:  
   A. input.value  
   B. input.text  
   C. input.content  
   D. input.getAttribute("text")  
   → Đáp án: A**
9. **textContent khác innerHTML ở chỗ:  
   A. textContent bao gồm HTML  
   B. textContent chỉ lấy nội dung thuần văn bản  
   C. Không có sự khác biệt  
   D. textContent dùng cho node cha  
   → Đáp án: B**
10. **Sự kiện onload xảy ra khi:  
    A. Trình duyệt tải xong trang  
    B. Nhấn chuột vào phần tử  
    C. Trang bị đóng  
    D. Thẻ script được gọi  
    → Đáp án: A**

**🔸 PHẦN 6: XỬ LÝ SỰ KIỆN NÂNG CAO**

1. **Để gắn nhiều sự kiện cho cùng một phần tử, nên dùng:  
   A. onclick = function()  
   B. addEventListener()  
   C. trigger()  
   D. setEvent()  
   → Đáp án: B**
2. **Tham số thứ hai của addEventListener("click", function) là gì?  
   A. Kiểu dữ liệu  
   B. Tên sự kiện  
   C. Hàm xử lý sự kiện  
   D. Giá trị mặc định  
   → Đáp án: C**
3. **Để ngăn hành vi mặc định của sự kiện, dùng:  
   A. event.returnValue = false  
   B. prevent()  
   C. event.preventDefault()  
   D. stopAction()  
   → Đáp án: C**
4. **Sự kiện change xảy ra khi nào?  
   A. Khi người dùng nhấn chuột  
   B. Khi nội dung phần tử thay đổi và mất focus  
   C. Khi người dùng rê chuột  
   D. Khi trang được reload  
   → Đáp án: B**
5. **Sự kiện keyup xảy ra khi:  
   A. Nhấn một phím bất kỳ  
   B. Nhấn giữ phím  
   C. Thả phím ra  
   D. Đưa chuột vào input  
   → Đáp án: C**
6. **Để dừng lan truyền sự kiện lên cha, dùng:  
   A. event.preventDefault()  
   B. event.stopPropagation()  
   C. cancelBubble()  
   D. stopBubble()  
   → Đáp án: B**
7. **Đối tượng event thường được truyền vào xử lý sự kiện để:  
   A. Tạo DOM mới  
   B. Lấy thông tin về sự kiện  
   C. Thêm CSS  
   D. Reset trang  
   → Đáp án: B**
8. **Sự kiện mouseover khác mouseenter ở điểm nào?  
   A. mouseover không lan xuống con  
   B. mouseenter không kích hoạt khi vào con  
   C. Chúng giống nhau  
   D. mouseover chỉ áp dụng với thẻ a  
   → Đáp án: B**
9. **Sự kiện input xảy ra khi:  
   A. Khi nhấn enter trong input  
   B. Khi nhấn submit  
   C. Khi người dùng thay đổi giá trị bất kỳ  
   D. Khi con trỏ rời khỏi input  
   → Đáp án: C**
10. **DOMContentLoaded là sự kiện xảy ra khi:  
    A. Toàn bộ tài nguyên (ảnh, JS, v.v.) đã tải xong  
    B. Toàn bộ DOM đã sẵn sàng, không chờ ảnh  
    C. Trình duyệt đóng  
    D. Lỗi xảy ra  
    → Đáp án: B**

**PHẦN 7: TRỌNG TÂM KIẾN THỨC DOM**

1. **DOM dùng để:  
   A. Hiển thị CSS  
   B. Quản lý cơ sở dữ liệu  
   C. Thao tác cấu trúc, nội dung và style của HTML  
   D. Thiết lập server  
   → Đáp án: C**
2. **document.getElementById("demo") sẽ trả về:  
   A. Chuỗi chứa ID  
   B. Một phần tử HTML có ID là "demo"  
   C. Một danh sách phần tử  
   D. Một giá trị số  
   → Đáp án: B**
3. **Khi sử dụng innerHTML, bạn nên cẩn thận vì:  
   A. Nó làm thay đổi nội dung CSS  
   B. Nó có thể gây lỗi trình duyệt  
   C. Nó có thể gây lỗ hổng bảo mật (XSS)  
   D. Nó không tương thích với HTML5  
   → Đáp án: C**
4. **DOM giúp kết nối giữa:  
   A. CSS và hình ảnh  
   B. HTML và server  
   C. HTML và JavaScript  
   D. JavaScript và JSON  
   → Đáp án: C**
5. **Để thay đổi nội dung văn bản của một thẻ <p>, dùng:  
   A. p.innerHTML  
   B. p.textContent  
   C. p.innerText  
   D. Cả B và C  
   → Đáp án: D**
6. **Tất cả các phần tử trong HTML có thể được truy cập bằng:  
   A. NodeList  
   B. DOM API  
   C. JSON  
   D. Java  
   → Đáp án: B**
7. **Sự khác biệt giữa textContent và innerHTML là:  
   A. innerHTML chỉ lấy văn bản  
   B. textContent bao gồm cả HTML  
   C. textContent không phân tích HTML  
   D. innerHTML không lấy thuộc tính  
   → Đáp án: C**
8. **Để thêm một phần tử mới vào cuối danh sách, ta dùng:  
   A. appendChild()  
   B. insertBefore()  
   C. setChild()  
   D. addNode()  
   → Đáp án: A**
9. **Phương thức querySelector() cho phép:  
   A. Lấy nhiều phần tử  
   B. Truy cập phần tử đầu tiên phù hợp với CSS selector  
   C. Tạo phần tử mới  
   D. Gán class  
   → Đáp án: B**
10. **DOM phân biệt rõ giữa:  
    A. HTML và CSS  
    B. Phần tử và thuộc tính  
    C. node cha và node con  
    D. Tất cả đáp án trên  
    → Đáp án: D**

**🔸 PHẦN 8: CỦNG CỐ KIẾN THỨC ỨNG DỤNG**

1. **document.body là cách để truy cập:  
   A. Toàn bộ tài liệu  
   B. Phần tử <body>  
   C. Thẻ đầu tiên  
   D. Tên trang  
   → Đáp án: B**
2. **Kết quả của document.childNodes[0] thường là:  
   A. <head>  
   B. #text  
   C. <!DOCTYPE> hoặc html  
   D. <body>  
   → Đáp án: C**
3. **Để kiểm tra loại node, dùng thuộc tính:  
   A. nodeType  
   B. nodeName  
   C. nodeCheck  
   D. typeOfNode  
   → Đáp án: A**
4. **Node có nodeType = 1 là:  
   A. Text node  
   B. Document node  
   C. Element node  
   D. Comment node  
   → Đáp án: C**
5. **getElementsByClassName("menu") trả về:  
   A. Một phần tử  
   B. Một chuỗi  
   C. Một HTMLCollection chứa các phần tử có class "menu"  
   D. Một JSON object  
   → Đáp án: C**
6. **Để kiểm tra phần tử có tồn tại trước khi thao tác, nên dùng:  
   A. try/catch  
   B. if (element != null)  
   C. exists(element)  
   D. checkNode()  
   → Đáp án: B**
7. **Lấy giá trị thuộc tính href trong thẻ <a>:  
   A. link.textContent  
   B. link.href  
   C. link.getValue()  
   D. link.getAttribute("src")  
   → Đáp án: B**
8. **Để thay thế phần tử A bằng phần tử B, dùng:  
   A. A.replaceChild(B)  
   B. parent.replaceChild(B, A)  
   C. document.changeNode(A, B)  
   D. A.innerHTML = B  
   → Đáp án: B**
9. **for (let item of collection) áp dụng được với:  
   A. HTMLCollection  
   B. NodeList  
   C. Array  
   D. Tất cả đáp án trên  
   → Đáp án: D**
10. **DOM là phần quan trọng trong mô hình:  
    A. MVC  
    B. AJAX  
    C. BOM  
    D. Lập trình web phía client  
    → Đáp án: D**

**AJAX**

**⚙️ Phần I: Tổng quan về AJAX**

1. **AJAX là viết tắt của cụm từ nào?  
   A. Advanced Java and XML  
   B. Asynchronous JavaScript and XML ✅  
   C. Asynchronous JSON and XML  
   D. Active Java and XML  
   ➤ *Giải thích:* AJAX là viết tắt của "Asynchronous JavaScript and XML", là kỹ thuật xây dựng ứng dụng web không cần tải lại toàn trang.**
2. **AJAX KHÔNG phải là gì?  
   A. Một ngôn ngữ lập trình ✅  
   B. Một kỹ thuật  
   C. Một cách tiếp cận  
   D. Một công cụ hỗ trợ web  
   ➤ *Giải thích:* AJAX không phải là ngôn ngữ lập trình, mà là sự kết hợp các công nghệ web có sẵn.**
3. **Lợi ích chính của AJAX là gì?  
   A. Tăng tốc độ tải toàn bộ trang  
   B. Tự động lưu trạng thái trang  
   C. Cập nhật nội dung mà không cần reload ✅  
   D. Mã hóa toàn bộ dữ liệu  
   ➤ *Giải thích:* AJAX cho phép cập nhật từng phần mà không cần reload trang.**
4. **Công nghệ nào KHÔNG liên quan đến AJAX?  
   A. HTML DOM  
   B. CSS  
   C. Java  
   D. XMLHttpRequest ✅  
   ➤ *Giải thích:* Java không phải thành phần trực tiếp của AJAX, XMLHttpRequest mới là thành phần cốt lõi.**
5. **AJAX hoạt động tốt nhất khi nào?  
   A. Dữ liệu nhỏ và thay đổi thường xuyên ✅  
   B. Khi tải toàn bộ website  
   C. Khi không có JavaScript  
   D. Khi không có kết nối internet  
   ➤ *Giải thích:* AJAX tối ưu khi cần cập nhật dữ liệu động một phần nhỏ.**

**🔧 Phần II: XMLHttpRequest Object**

1. **Đối tượng cốt lõi của AJAX là gì?  
   A. JavaObject  
   B. XMLHttpRequest ✅  
   C. AjaxDOM  
   D. RequestHandler  
   ➤ *Giải thích:* XMLHttpRequest dùng để gửi/nhận dữ liệu không đồng bộ với server.**
2. **Lệnh để tạo XMLHttpRequest?  
   A. new Ajax()  
   B. XMLHttpRequest.create()  
   C. new XMLHttpRequest() ✅  
   D. getXMLRequest()  
   ➤ *Giải thích:* Cú pháp khởi tạo XMLHttpRequest: var xhttp = new XMLHttpRequest();**
3. **Thuộc tính readyState = 4 nghĩa là gì?  
   A. Chưa khởi tạo  
   B. Đã gửi  
   C. Đang nhận dữ liệu  
   D. Hoàn tất ✅  
   ➤ *Giải thích:* Giá trị 4 của readyState nghĩa là đã hoàn tất nhận dữ liệu từ server.**
4. **Phương thức .open() KHÔNG yêu cầu tham số nào?  
   A. method  
   B. url  
   C. callback ✅  
   D. async  
   ➤ *Giải thích:* .open(method, url, async) không cần callback.**
5. **Phương thức gửi dữ liệu đến server là?  
   A. load()  
   B. send() ✅  
   C. execute()  
   D. request()  
   ➤ *Giải thích:* .send() dùng để gửi yêu cầu đến server.**

**🛰️ Phần III: Chế độ đồng bộ & bất đồng bộ**

1. **Chế độ mặc định của AJAX là gì?  
   A. Đồng bộ  
   B. Bất đồng bộ ✅  
   ➤ *Giải thích:* async = true là mặc định => không chặn luồng JavaScript.**
2. **Chế độ đồng bộ KHÔNG được khuyến khích vì?  
   A. Chạy sai  
   B. Chặn giao diện người dùng ✅  
   C. Không gửi được dữ liệu  
   D. Không bảo mật  
   ➤ *Giải thích:* Đồng bộ khiến trình duyệt “đơ” nếu server phản hồi chậm.**
3. **Hàm dùng để hủy yêu cầu AJAX đang chạy?  
   A. stop()  
   B. end()  
   C. abort() ✅  
   ➤ *Giải thích:* xhr.abort(); dùng để hủy yêu cầu chưa hoàn tất.**

**📡 Phần IV: Gửi & nhận dữ liệu**

1. **Dữ liệu phản hồi dạng văn bản lưu ở đâu?  
   A. responseData  
   B. responseText ✅  
   ➤ *Giải thích:* responseText chứa dữ liệu trả về dạng chuỗi.**
2. **Khi phản hồi là XML, sử dụng thuộc tính nào?  
   A. responseText  
   B. responseJSON  
   C. responseXML ✅  
   ➤ *Giải thích:* responseXML chứa đối tượng XML DOM nếu phản hồi là XML.**
3. **Trạng thái HTTP trả về 200 nghĩa là gì?  
   A. Thành công ✅  
   B. Lỗi  
   ➤ *Giải thích:* Mã trạng thái 200 nghĩa là server đã xử lý thành công.**
4. **Dùng phương thức nào để đặt header cho request?  
   A. setHeader()  
   B. setRequestHeader() ✅  
   ➤ *Giải thích:* .setRequestHeader("Content-type", "...") thiết lập loại dữ liệu gửi.**
5. **Dữ liệu từ form thường gửi bằng phương thức nào?  
   A. GET  
   B. POST ✅  
   ➤ *Giải thích:* POST phù hợp gửi dữ liệu từ form, nhất là khi nhiều và có ký tự đặc biệt.**
6. **Dữ liệu gửi bằng GET nằm ở đâu?  
   A. Trong header  
   B. Trong body  
   C. Trong URL ✅  
   ➤ *Giải thích:* Dữ liệu GET gắn kèm sau dấu ? của URL.**
7. **Khi nào nên dùng POST thay vì GET?  
   A. Khi dữ liệu lớn ✅  
   B. Khi cần cache  
   ➤ *Giải thích:* POST không giới hạn độ dài dữ liệu và an toàn hơn.**

**🔄 Phần V: Kết nối AJAX với PHP & MySQL**

1. **PHP script sẽ làm gì trong mô hình AJAX?  
   A. Gửi giao diện  
   B. Nhận request, truy vấn MySQL, trả kết quả ✅  
   ➤ *Giải thích:* PHP nhận yêu cầu, xử lý cơ sở dữ liệu, trả dữ liệu lại cho client.**
2. **Kết quả trả về từ PHP có thể ở định dạng nào?  
   A. Plain Text  
   B. JSON  
   C. XML  
   D. Tất cả ✅  
   ➤ *Giải thích:* PHP có thể xuất ra nhiều định dạng tùy yêu cầu.**
3. **Dữ liệu JSON được xử lý trên JavaScript bằng?  
   A. JSON.parse() ✅  
   B. eval()  
   ➤ *Giải thích:* JSON.parse() chuyển chuỗi JSON thành đối tượng.**
4. **FormData object dùng để làm gì?  
   A. Tạo giao diện  
   B. Gửi dữ liệu biểu mẫu ✅  
   ➤ *Giải thích:* FormData dùng để thu thập và gửi dữ liệu từ form.**
5. **Hàm encodeURIComponent dùng để?  
   A. Mã hóa dữ liệu query string ✅  
   ➤ *Giải thích:* Tránh lỗi khi gửi các ký tự đặc biệt qua URL.**

**🧠 Phần VI: Các ví dụ thực hành**

1. **Định dạng chuỗi query GET?  
   A. name=value&name2=value2 ✅  
   ➤ *Giải thích:* Chuỗi nối sau URL, ví dụ: ...?name=John&age=30.**
2. **Thêm dữ liệu POST vào send() thế nào?  
   A. Truyền vào trực tiếp chuỗi dữ liệu ✅  
   ➤ *Giải thích:* xhr.send("name=John&age=30")**
3. **Để lấy toàn bộ header phản hồi dùng hàm?  
   A. getAllHeaders  
   B. getAllResponseHeaders ✅  
   ➤ *Giải thích:* Trả về toàn bộ các trường header.**
4. **AJAX hỗ trợ các tệp nào từ server?  
   A. txt  
   B. php  
   C. xml  
   D. Tất cả ✅  
   ➤ *Giải thích:* AJAX có thể giao tiếp với bất kỳ tệp server-side nào.**
5. **AJAX có hỗ trợ truy cập khác miền (cross-domain)?  
   A. Có  
   B. Không ✅  
   ➤ *Giải thích:* Trình duyệt chặn cross-domain để đảm bảo bảo mật.**

**🔍 Phần VII: Mã AJAX chi tiết**

1. **Đâu là cách đúng để xử lý sự kiện thay đổi trạng thái yêu cầu AJAX?  
   A. xhr.whenReady = function()  
   B. xhr.onreadystatechange = function() ✅  
   ➤ *Giải thích:* Sử dụng onreadystatechange để xử lý khi trạng thái thay đổi.**
2. **Thuộc tính nào cho biết trạng thái hiện tại của yêu cầu?  
   A. xhr.status  
   B. xhr.state  
   C. xhr.readyState ✅  
   ➤ *Giải thích:* readyState có 5 mức từ 0 đến 4.**
3. **Trong xử lý AJAX, kiểm tra thành công khi nào?  
   A. readyState == 4 && status == 200 ✅  
   ➤ *Giải thích:* readyState=4 hoàn tất và status=200 nghĩa là OK.**
4. **Sự kiện onload trong XMLHttpRequest dùng khi nào?  
   A. Khi server phản hồi hoàn tất ✅  
   ➤ *Giải thích:* onload thay thế onreadystatechange trong trình duyệt hiện đại.**
5. **AJAX xử lý dữ liệu JSON như thế nào?  
   A. Bằng JSON.stringify()  
   B. Bằng JSON.parse() ✅  
   ➤ *Giải thích:* JSON.parse() chuyển chuỗi JSON thành đối tượng.**
6. **AJAX giúp tăng trải nghiệm người dùng bằng cách nào?  
   A. Tăng tốc độ server  
   B. Tải lại toàn bộ trang  
   C. Cập nhật một phần trang nhanh chóng ✅  
   ➤ *Giải thích:* Giúp nội dung thay đổi mượt mà, không gián đoạn.**

**🧩 Phần VIII: Kết nối với PHP nâng cao**

1. **Làm sao để gửi dữ liệu từ client đến PHP bằng POST?  
   A. Gửi qua URL  
   B. Gửi trong send() ✅  
   ➤ *Giải thích:* POST gửi dữ liệu qua body: xhr.send("key=value").**
2. **PHP nhận dữ liệu POST từ AJAX bằng biến nào?  
   A. $\_GET  
   B. $\_POST ✅  
   ➤ *Giải thích:* Với phương thức POST, dùng biến $\_POST để truy cập dữ liệu.**
3. **Để phản hồi dữ liệu JSON từ PHP, dùng hàm nào?  
   A. echo()  
   B. json\_encode() ✅  
   ➤ *Giải thích:* json\_encode() chuyển mảng hoặc đối tượng thành JSON.**
4. **Cần thiết lập header gì khi trả JSON từ PHP?  
   A. text/html  
   B. application/json ✅  
   ➤ *Giải thích:* header("Content-Type: application/json") thông báo dữ liệu là JSON.**

**🛠️ Phần IX: Xử lý lỗi và bảo mật**

1. **Nếu server trả về lỗi 404, có nghĩa là gì?  
   A. Sai cú pháp  
   B. Không tìm thấy tài nguyên ✅  
   ➤ *Giải thích:* Mã lỗi HTTP 404 là "Not Found".**
2. **Thuộc tính status dùng để kiểm tra gì?  
   A. Trạng thái xử lý  
   B. Mã HTTP trả về ✅  
   ➤ *Giải thích:* xhr.status trả về mã HTTP như 200, 404, 500.**
3. **AJAX không thể gọi sang domain khác nếu không có gì?  
   A. Header Access-Control-Allow-Origin ✅  
   ➤ *Giải thích:* CORS yêu cầu header này để cho phép truy cập khác miền.**
4. **AJAX KHÔNG nên dùng để xử lý thao tác nào?  
   A. Đăng nhập  
   B. Đăng xuất  
   C. Cập nhật tiền trong tài khoản ngân hàng ✅  
   ➤ *Giải thích:* Những thao tác nhạy cảm cần xác thực kỹ, tránh lộ qua AJAX.**
5. **Giải pháp bảo mật thường dùng trong AJAX là?  
   A. Base64 encode  
   B. Captcha  
   C. Token CSRF ✅  
   ➤ *Giải thích:* Token CSRF chống tấn công giả mạo yêu cầu.**

**🌐 Phần X: JSON và XML**

1. **Ưu điểm chính của JSON so với XML?  
   A. Ít bảo mật hơn  
   B. Cấu trúc đơn giản, dễ phân tích ✅  
   ➤ *Giải thích:* JSON nhẹ, dễ đọc/viết bằng JavaScript.**
2. **Hàm nào trong PHP phân tích JSON?  
   A. json\_parse()  
   B. json\_decode() ✅  
   ➤ *Giải thích:* json\_decode() chuyển JSON thành mảng hoặc đối tượng PHP.**
3. **Để hiển thị dữ liệu JSON trong HTML, cần gì?  
   A. Chuyển thành DOM element ✅  
   ➤ *Giải thích:* Dữ liệu cần được xử lý rồi đưa vào nội dung HTML.**
4. **AJAX xử lý XML bằng đối tượng nào?  
   A. JSONDocument  
   B. DOMParser  
   C. responseXML ✅  
   ➤ *Giải thích:* responseXML là đối tượng DOM có thể thao tác bằng JavaScript.**
5. **Để truy cập một thẻ <name> trong XML dùng gì?  
   A. xml.name  
   B. getElementsByTagName("name") ✅  
   ➤ *Giải thích:* Truy cập bằng DOM như HTML.**

**💻 Phần XI: AJAX nâng cao**

1. **AJAX polling là gì?  
   A. Gửi 1 request  
   B. Gửi nhiều request định kỳ để lấy dữ liệu ✅  
   ➤ *Giải thích:* Polling = thăm dò server liên tục.**
2. **Thay vì polling, nên dùng kỹ thuật gì hiện đại hơn?  
   A. Long Polling  
   B. WebSocket ✅  
   ➤ *Giải thích:* WebSocket cho kết nối 2 chiều, hiệu quả hơn.**
3. **Khi dùng jQuery AJAX, phương thức phổ biến là gì?  
   A. $.send()  
   B. $.ajax() ✅  
   ➤ *Giải thích:* jQuery cung cấp hàm $.ajax() rất linh hoạt.**
4. **Thuộc tính async: false trong $.ajax nghĩa là gì?  
   A. Gửi đồng bộ ✅  
   ➤ *Giải thích:* async: false làm yêu cầu chạy đồng bộ — nên tránh.**
5. **Hàm xử lý lỗi trong $.ajax là?  
   A. onError  
   B. error ✅  
   ➤ *Giải thích:* error: function(xhr, status, error) { ... }**

**🧪 Phần XII: Thực tiễn và tối ưu**

1. **Thay vì innerHTML, nên dùng gì để cập nhật dữ liệu an toàn hơn?  
   A. document.write()  
   B. textContent ✅  
   ➤ *Giải thích:* textContent tránh chèn HTML nguy hiểm.**
2. **Tại sao nên dùng encodeURIComponent() khi gửi dữ liệu GET?  
   A. Mã hóa tránh lỗi ký tự đặc biệt ✅  
   ➤ *Giải thích:* Tránh lỗi khi có dấu cách, &, %, ...**
3. **AJAX thường dùng để làm tính năng nào?  
   A. In tài liệu  
   B. Hiển thị nội dung động như gợi ý tìm kiếm ✅  
   ➤ *Giải thích:* AJAX hỗ trợ gợi ý khi người dùng nhập liệu.**
4. **AJAX có thể kết hợp với framework frontend nào?  
   A. React  
   B. Angular  
   C. Vue  
   D. Tất cả ✅  
   ➤ *Giải thích:* AJAX là kỹ thuật nền có thể dùng trong bất kỳ framework nào.**
5. **Tại sao AJAX quan trọng với ứng dụng web hiện đại?  
   A. Giảm chi phí hosting  
   B. Tăng SEO  
   C. Tạo trải nghiệm người dùng mượt mà, giống ứng dụng desktop ✅  
   ➤ *Giải thích:* AJAX giúp trang không phải reload, tăng UX.**

**🧠 Phần XIII: Thực hành và xử lý dữ liệu**

1. **AJAX có thể xử lý phản hồi dưới những dạng nào?  
   A. Văn bản (text)  
   B. XML  
   C. JSON  
   D. Tất cả đều đúng ✅  
   ➤ *Giải thích:* responseText, responseXML và parse JSON đều được hỗ trợ.**
2. **AJAX giúp cải thiện UX bằng cách nào?  
   A. Giảm tài nguyên server  
   B. Giao tiếp không đồng bộ giúp tương tác mượt ✅  
   ➤ *Giải thích:* AJAX không làm tải lại toàn bộ trang => UX tốt hơn.**
3. **Cách tối ưu hiệu suất khi dùng AJAX nhiều lần là?  
   A. Giảm số lần gửi request ✅  
   ➤ *Giải thích:* Kết hợp yêu cầu hoặc dùng cache giúp giảm tải server.**
4. **AJAX thường được dùng để xây dựng gì?  
   A. Ứng dụng thời gian thực (chat, thông báo) ✅  
   ➤ *Giải thích:* AJAX giúp cập nhật nội dung liên tục mà không cần reload.**
5. **Vấn đề thường gặp khi làm việc với AJAX là gì?  
   A. Không xử lý được lỗi server ✅  
   ➤ *Giải thích:* AJAX cần có cơ chế phát hiện lỗi trả về từ backend.**

**🧪 Phần XIV: Tối ưu hóa và thực tiễn**

1. **Cách tốt nhất để kiểm tra dữ liệu trả về từ AJAX là?  
   A. In ra console để debug ✅  
   ➤ *Giải thích:* Dùng console.log() giúp kiểm tra dữ liệu nhận được.**
2. **Trong JavaScript, XMLHttpRequest là đối tượng gì?  
   A. Lớp xử lý DOM  
   B. Đối tượng HTTP client ✅  
   ➤ *Giải thích:* Nó là đối tượng dùng để gửi request đến server.**
3. **AJAX KHÔNG thể thao tác trực tiếp với?  
   A. LocalStorage  
   B. Database ✅  
   ➤ *Giải thích:* AJAX chỉ gửi dữ liệu – việc truy xuất database nằm ở phía server (PHP, Node,...)**
4. **AJAX không nên lạm dụng vì?  
   A. Tốn RAM  
   B. Gây khó khăn cho SEO ✅  
   ➤ *Giải thích:* Bot tìm kiếm không xử lý tốt nội dung tải động.**
5. **Giải pháp khắc phục hạn chế SEO của AJAX?  
   A. SSR (Server Side Rendering) ✅  
   ➤ *Giải thích:* Kết hợp AJAX với SSR giúp tăng khả năng SEO.**

**💬 Phần XV: AJAX và sự kiện người dùng**

1. **AJAX thường được kích hoạt bởi sự kiện nào?  
   A. load  
   B. click ✅  
   ➤ *Giải thích:* AJAX thường gọi khi người dùng nhấn nút, chọn dropdown...**
2. **Sự kiện change có thể được dùng để gọi AJAX khi nào?  
   A. Khi nhấn nút  
   B. Khi người dùng chọn giá trị từ input ✅  
   ➤ *Giải thích:* change rất hữu ích khi dùng form kết hợp AJAX.**
3. **Cần làm gì sau khi nhận dữ liệu từ AJAX?  
   A. Chuyển sang JSON  
   B. Gán vào HTML tương ứng ✅  
   ➤ *Giải thích:* Nhận được dữ liệu => cập nhật nội dung HTML.**
4. **AJAX hỗ trợ kỹ thuật nhập liệu nào?  
   A. Auto-complete ✅  
   ➤ *Giải thích:* Gợi ý tự động khi người dùng gõ từ khóa là ứng dụng phổ biến của AJAX.**
5. **Tại sao AJAX phải gán Content-type khi POST?  
   A. Để trình duyệt hiểu kiểu dữ liệu ✅  
   ➤ *Giải thích:* application/x-www-form-urlencoded là phổ biến khi gửi POST.**

**🔄 Phần XVI: Xử lý nhiều yêu cầu và đồng bộ**

1. **Lỗi thường gặp khi gửi nhiều AJAX liên tục?  
   A. Trùng request, quá tải server ✅  
   ➤ *Giải thích:* Cần debounce hoặc throttle yêu cầu.**
2. **Giải pháp tránh gửi trùng AJAX là gì?  
   A. Dùng setTimeout  
   B. Kiểm tra trạng thái trước khi gửi ✅  
   ➤ *Giải thích:* Đảm bảo request trước hoàn tất rồi mới gửi tiếp.**
3. **AJAX có thể sử dụng Promise không?  
   A. Có ✅  
   ➤ *Giải thích:* XMLHttpRequest không có sẵn Promise, nhưng ta có thể bọc lại.**
4. **AJAX có thể dùng async/await không?  
   A. Có, nếu được Promise hóa ✅  
   ➤ *Giải thích:* async/await cần Promise, thường dùng với fetch.**
5. **AJAX có thể dừng giữa chừng không?  
   A. Có, dùng xhr.abort() ✅  
   ➤ *Giải thích:* xhr.abort() dừng yêu cầu đang gửi.**

**⚙️ Phần XVII: Fetch API – Giải pháp thay thế AJAX hiện đại**

1. **Fetch API là gì?  
   A. Công cụ tạo DOM  
   B. Thay thế hiện đại cho XMLHttpRequest ✅  
   ➤ *Giải thích:* Fetch API đơn giản hơn, hỗ trợ Promise.**
2. **Cú pháp trả về kết quả JSON trong Fetch là?  
   A. .then(res => res.json()) ✅  
   ➤ *Giải thích:* res.json() trả về Promise chứa dữ liệu JSON.**
3. **Phản hồi lỗi trong Fetch API phải xử lý thế nào?  
   A. .catch() ✅  
   ➤ *Giải thích:* .catch() xử lý lỗi mạng, còn lỗi HTTP phải kiểm tra thủ công res.ok.**
4. **Fetch trả về lỗi HTTP (404), có vào catch không?  
   A. Không ✅  
   ➤ *Giải thích:* Phải tự kiểm tra response.ok hoặc status.**
5. **Fetch có hỗ trợ POST không?  
   A. Có, thông qua { method: "POST", body: ... } ✅  
   ➤ *Giải thích:* Tùy chỉnh method và headers dễ dàng với Fetch.**

**📚 Phần XVIII: Tổng hợp ứng dụng AJAX**

1. **AJAX giúp cải thiện hiệu năng bằng cách nào?  
   A. Giảm số lượng trang tải ✅  
   ➤ *Giải thích:* Chỉ thay đổi vùng cần thiết, giảm số lần load trang.**
2. **Kết hợp AJAX với database giúp gì?  
   A. Truy vấn dữ liệu theo thời gian thực ✅  
   ➤ *Giải thích:* AJAX gửi yêu cầu đến server, server truy xuất database và trả kết quả.**
3. **AJAX thường được dùng trong hệ thống nào?  
   A. Blog  
   B. Trang thương mại điện tử  
   C. Ứng dụng email  
   D. Tất cả ✅  
   ➤ *Giải thích:* AJAX phù hợp cho mọi loại web hiện đại.**
4. **Một nguy cơ khi dùng AJAX là gì?  
   A. Lộ dữ liệu nhạy cảm nếu không mã hóa ✅  
   ➤ *Giải thích:* AJAX gửi/nhận dữ liệu => nên bảo mật bằng HTTPS.**
5. **AJAX KHÔNG nên dùng để thay thế hoàn toàn?  
   A. Hệ thống routing  
   B. SEO nội dung ✅  
   ➤ *Giải thích:* Nội dung AJAX không được index tốt bởi máy tìm kiếm.**

**🎯 Phần XIX: Tình huống thực tế và kiểm tra**

1. **Nếu AJAX không phản hồi gì, bước đầu kiểm tra là gì?  
   A. Xem lại đường dẫn URL ✅  
   ➤ *Giải thích:* Đường dẫn sai là lỗi phổ biến.**
2. **AJAX gửi sai dữ liệu có thể do đâu?  
   A. Không encode dữ liệu ✅  
   ➤ *Giải thích:* Ký tự đặc biệt gây lỗi nếu không mã hóa.**
3. **AJAX hỗ trợ thao tác CRUD như thế nào?  
   A. Dùng GET, POST, PUT, DELETE ✅  
   ➤ *Giải thích:* Giao tiếp RESTful phổ biến trong AJAX.**
4. **Khi xhr.responseText là rỗng, có thể là vì?  
   A. Server chưa phản hồi ✅  
   ➤ *Giải thích:* Có thể request bị lỗi hoặc chưa tới server.**
5. **AJAX ảnh hưởng gì đến lịch sử trình duyệt?  
   A. Không tạo mục trong lịch sử ✅  
   ➤ *Giải thích:* Vì không reload trang, nên không thay đổi URL.**
6. **Để AJAX hỗ trợ back/forward trình duyệt, ta cần?  
   A. HTML5 History API ✅  
   ➤ *Giải thích:* history.pushState() cho phép quản lý URL không reload.**
7. **AJAX thường kết hợp với thư viện nào để đơn giản hóa?  
   A. jQuery ✅  
   ➤ *Giải thích:* jQuery giúp gọi AJAX ngắn gọn, dễ dùng.**
8. **AJAX có thể thực hiện upload file không?  
   A. Có, dùng FormData ✅  
   ➤ *Giải thích:* FormData cho phép gửi file và dữ liệu khác.**
9. **AJAX có thể dùng trong SPA (Single Page Application) không?  
   A. Có ✅  
   ➤ *Giải thích:* AJAX là lõi quan trọng của SPA.**
10. **Từ AJAX, muốn kết nối real-time thì nâng cấp lên công nghệ nào?  
    A. WebSocket ✅  
    ➤ *Giải thích:* WebSocket giữ kết nối liên tục, phù hợp ứng dụng thời gian thực.**

**Node.js**

**1–10: Kiến thức cơ bản về Express**

1. Express là gì?
   * A. Một loại cơ sở dữ liệu
   * B. Một trình duyệt
   * ⭐ C. Một framework cho Node.js
   * D. Một phần mềm dựng hình
2. Express được viết bằng ngôn ngữ nào?
   * A. Java
   * ⭐ B. JavaScript
   * C. Python
   * D. C++
3. Để cài đặt Express, bạn dùng lệnh nào?
   * A. install express
   * ⭐ B. npm install express --save
   * C. express install
   * D. node install express
4. Lệnh app.listen(3000) trong Express dùng để làm gì?
   * A. Kết nối cơ sở dữ liệu
   * B. Tạo route
   * ⭐ C. Khởi động server lắng nghe cổng 3000
   * D. Chạy middleware
5. Tập tin package.json lưu thông tin gì?
   * A. Nội dung HTML
   * B. Lịch sử trình duyệt
   * C. Cấu trúc thư mục
   * ⭐ D. Thông tin gói và phụ thuộc trong Node.js
6. Câu lệnh nào tạo ứng dụng Express cơ bản?
   * ⭐ A. const app = express();
   * B. express(app)
   * C. create express();
   * D. express.start();
7. Express hỗ trợ định tuyến theo bao nhiêu phương thức HTTP cơ bản?
   * A. 2
   * B. 3
   * ⭐ C. 4 trở lên (GET, POST, PUT, DELETE...)
   * D. Chỉ GET và POST
8. Route trong Express được tạo với hàm nào?
   * A. route()
   * B. send()
   * ⭐ C. app.get(), app.post(),...
   * D. express.route()
9. Middleware là gì?
   * A. Một kiểu cơ sở dữ liệu
   * B. Một hàm định tuyến
   * ⭐ C. Một hàm xử lý trước khi tới route chính
   * D. Một thư viện riêng
10. Middleware luôn cần gọi hàm nào để chuyển tiếp?
    * A. forward()
    * B. nextRoute()
    * ⭐ C. next()
    * D. continue()

**11–20: Routing và Request Handling**

1. Route GET nhận request từ:
   * A. Client gửi POST
   * ⭐ B. Client gửi GET
   * C. Server khác
   * D. Middleware
2. Route POST chủ yếu dùng để:
   * ⭐ A. Nhận dữ liệu gửi từ form
   * B. Hiển thị trang
   * C. Tạo cookie
   * D. Xóa dữ liệu
3. Câu lệnh gửi phản hồi văn bản về client là:
   * A. res.get()
   * B. res.route()
   * ⭐ C. res.send()
   * D. res.write()
4. res.sendFile() dùng để:
   * A. Gửi text
   * ⭐ B. Gửi file về client
   * C. Nhận file
   * D. Hiển thị console log
5. req.params chứa gì?
   * ⭐ A. Tham số từ URL
   * B. Biến toàn cục
   * C. Tham số middleware
   * D. Tham số cấu hình Express
6. req.query được dùng khi nào?
   * ⭐ A. Client gửi query string trên URL
   * B. Trong file CSS
   * C. Khi kết nối cơ sở dữ liệu
   * D. Trong POST request
7. Route /user/:id sẽ bắt giá trị nào?
   * A. user
   * B. :id là tên hàm
   * ⭐ C. Một giá trị động từ URL
   * D. Tên middleware
8. app.use() thường được dùng để:
   * A. Tạo route
   * ⭐ B. Dùng middleware
   * C. Tạo view
   * D. Gọi file JS
9. Một middleware có thể thao tác với đối tượng nào?
   * A. File
   * ⭐ B. req, res
   * C. ejs
   * D. URL
10. res.status(404).send() dùng để:
    * ⭐ A. Gửi lỗi 404
    * B. Xác thực người dùng
    * C. Redirect
    * D. Kết nối API

**21–30: Middleware nâng cao và Body Parser**

1. Middleware phải gọi next() để:
   * A. Trả response
   * ⭐ B. Chuyển sang middleware/route tiếp theo
   * C. Reset server
   * D. Dừng ứng dụng
2. Thư viện nào dùng để xử lý dữ liệu POST?
   * A. express-body
   * B. request
   * ⭐ C. body-parser
   * D. postman
3. body-parser.urlencoded() xử lý dữ liệu:
   * A. JSON
   * ⭐ B. Form HTML
   * C. File PDF
   * D. Email
4. Dòng extended: false nghĩa là gì?
   * A. Không cho phép gửi dữ liệu
   * B. Dùng cơ sở dữ liệu
   * ⭐ C. Sử dụng thư viện querystring
   * D. Bắt buộc dùng token
5. req.body chứa dữ liệu từ:
   * A. GET request
   * ⭐ B. POST request
   * C. Middleware
   * D. res object
6. Middleware có thể đặt ở vị trí nào trong code?
   * A. Cuối cùng
   * ⭐ B. Trước định tuyến
   * C. Sau res.send()
   * D. Trong file .ejs
7. Middleware custom do ai viết?
   * A. Express core
   * B. NPM
   * ⭐ C. Lập trình viên
   * D. Server
8. Middleware có bao nhiêu tham số?
   * A. 1
   * B. 2
   * ⭐ C. 3 (req, res, next)
   * D. 4
9. res.redirect('/home') dùng để:
   * ⭐ A. Điều hướng tới URL khác
   * B. Tạo route
   * C. Xử lý lỗi
   * D. Gửi file
10. Middleware xử lý lỗi có thêm tham số nào?
    * A. msg
    * B. error
    * ⭐ C. err (đầu tiên)
    * D. res.end()

**31–40: Template Engine và EJS**

1. Template Engine dùng để:
   * A. Gửi POST request
   * ⭐ B. Tạo HTML động
   * C. Ghi log
   * D. Kết nối SQL
2. Express hỗ trợ engine nào sau đây?
   * A. Vue
   * B. React
   * ⭐ C. EJS
   * D. Flask
3. Câu lệnh cài EJS:
   * ⭐ A. npm install ejs
   * B. install ejs --html
   * C. node ejs
   * D. ejs setup
4. Cấu hình EJS trong Express:
   * A. app.engine('ejs')
   * ⭐ B. app.set('view engine', 'ejs')
   * C. express.view('ejs')
   * D. app.view('ejs')
5. Hàm res.render('home') dùng để:
   * ⭐ A. Hiển thị giao diện home.ejs
   * B. Tạo route mới
   * C. Gửi HTML thô
   * D. Xử lý POST
6. Trong file EJS, để chèn biến ta dùng:
   * A. {{ title }}
   * ⭐ B. <%= title %>
   * C. :title
   * D. { title }
7. Biến được truyền từ route vào view thông qua:
   * A. req.body
   * ⭐ B. Đối số thứ hai của res.render()
   * C. query
   * D. fs.readFile
8. Template Engine giúp:
   * A. Gửi API
   * B. Định tuyến
   * ⭐ C. Tạo nội dung HTML theo dữ liệu
   * D. Xác thực JWT
9. Tên thư mục chứa file EJS mặc định là:
   * A. static
   * ⭐ B. views
   * C. html
   * D. public
10. File EJS có phần mở rộng là:
    * A. .html
    * B. .jsx
    * ⭐ C. .ejs
    * D. .tpl

**41–50: Tổng hợp ứng dụng**

1. Express kết hợp tốt với:
   * A. MongoDB
   * B. MySQL
   * C. Redis
   * ⭐ D. Tất cả các cơ sở dữ liệu trên
2. Express dùng trong loại ứng dụng nào?
   * A. Desktop
   * B. Mobile app
   * ⭐ C. Web Server/API
   * D. Game Engine
3. Để xử lý lỗi toàn cục, ta nên:
   * A. Tạo route riêng
   * ⭐ B. Dùng middleware lỗi
   * C. Gửi alert
   * D. Log ra console
4. Middleware express.static() dùng để:
   * ⭐ A. Phục vụ file tĩnh (CSS, JS, img)
   * B. Chạy SQL
   * C. Tạo route
   * D. Nạp view
5. Route /about và /about/ có giống nhau không?
   * A. Luôn giống
   * B. Không bao giờ giống
   * ⭐ C. Có thể khác tùy cấu hình
   * D. Chỉ giống khi dùng GET
6. express.json() dùng cho:
   * A. Gửi dữ liệu GET
   * ⭐ B. Middleware xử lý JSON body
   * C. Tạo cookie
   * D. Gửi file
7. Express hoạt động dựa trên:
   * A. Blocking IO
   * ⭐ B. Non-blocking IO
   * C. Giao diện đồ họa
   * D. Websocket mặc định
8. Express được cài qua:
   * A. yarn
   * B. curl
   * C. brew
   * ⭐ D. npm
9. Express có được tích hợp trong Node.js mặc định không?
   * A. Có
   * ⭐ B. Không, phải cài riêng
   * C. Tùy OS
   * D. Chỉ có trong bản 18+
10. Express thường dùng trong mô hình:
    * A. MVC
    * B. MVVM
    * C. MVP
    * ⭐ D. MVC (với view engine)

Nếu bạn muốn mình tạo thêm file Word, PDF hoặc định dạng cho quiz app/Google Forms thì mình có thể hỗ trợ luôn!

4o

**You said:**

tạo cho tôi thêm 50 câu trắc nghiệm nữa

**ChatGPT said:**

Dưới đây là **50 câu hỏi trắc nghiệm tiếp theo** (từ câu 51 đến 100), tiếp tục mở rộng từ nội dung chương 8 về **Express.js trong Node.js**. Các câu hỏi đi sâu hơn vào ứng dụng thực tiễn, chi tiết API, và xử lý tình huống. Mỗi câu gồm 4 lựa chọn và đáp án đúng được đánh dấu ⭐.

**51–60: Routing nâng cao và Dynamic Routes**

1. Trong route /product/:id, giá trị :id được truy xuất qua:

* A. req.body.id
* B. req.query.id
* ⭐ C. req.params.id
* D. req.id

1. Route động /user/:id/profile sẽ khớp với URL nào?

* A. /user/profile/123
* ⭐ B. /user/123/profile
* C. /user/:profile/123
* D. /profile/user/123

1. Bạn muốn định nghĩa một route bắt mọi phương thức HTTP, bạn dùng:

* A. app.get()
* B. app.post()
* ⭐ C. app.all()
* D. app.any()

1. Câu lệnh app.route('/book').get(...).post(...) có ý nghĩa gì?

* A. Định nghĩa 2 URL khác nhau
* ⭐ B. Gộp nhiều phương thức trên cùng một URL
* C. Tạo nhiều middleware
* D. Dùng cho router tĩnh

1. Để định nghĩa tiền tố route như /api/\*, ta dùng:

* A. app.get('/api/\*')
* B. app.api('\*')
* ⭐ C. app.use('/api', router)
* D. app.prefix('/api')

1. app.use('/admin', adminRouter) có tác dụng gì?

* A. Phân quyền admin
* B. Đăng nhập admin
* ⭐ C. Dùng các route của adminRouter dưới đường /admin
* D. Gọi API bên ngoài

1. Router trong Express được tạo bằng:

* A. new Route()
* B. express.use()
* ⭐ C. express.Router()
* D. route.create()

1. router.get('/', handler) là cách định nghĩa:

* A. Middleware
* B. Route global
* ⭐ C. Route trong router con
* D. Template route

1. Để gắn router con vào app chính, ta dùng:

* A. router.attach(app)
* ⭐ B. app.use('/path', router)
* C. router.export()
* D. app.route(router)

1. Trong router con, để export, bạn dùng:

* A. module.export(router)
* B. exports.router
* ⭐ C. module.exports = router
* D. route.export()

**61–70: Middleware nâng cao**

1. Middleware có thể kiểm soát:

* A. HTML
* ⭐ B. Quy trình xử lý request
* C. DOM
* D. Session chỉ

1. Middleware có thể chặn route bằng cách:

* A. Gọi app.close()
* B. Dừng res.render
* ⭐ C. Không gọi next()
* D. Xóa req

1. Express có sẵn middleware nào sau đây?

* A. express.log()
* ⭐ B. express.static()
* C. express.session()
* D. express.route()

1. Middleware xử lý lỗi có 4 tham số nào?

* A. req, res, err, next
* ⭐ B. err, req, res, next
* C. res, err, req, next
* D. next, res, err, req

1. Để xử lý mọi lỗi chưa được xử lý, bạn nên:

* A. console.log lỗi
* ⭐ B. Viết middleware lỗi ở cuối tất cả route
* C. Tạo file log
* D. Tắt server

1. app.use(express.static('public')) nghĩa là gì?

* A. Tạo route tĩnh
* ⭐ B. Phục vụ file trong thư mục public
* C. Bảo vệ file
* D. Nạp HTML từ server khác

1. Nếu không có middleware xử lý lỗi, lỗi xảy ra sẽ:

* A. Bị bỏ qua
* ⭐ B. Làm server crash
* C. In ra HTML lỗi
* D. Gửi lỗi 200

1. Một middleware custom thường có vị trí:

* A. Sau tất cả route
* B. Trong file .ejs
* ⭐ C. Trước hoặc giữa các route
* D. Trong public/

1. Express hỗ trợ bao nhiêu middleware cùng lúc?

* A. 1
* B. 2
* ⭐ C. Không giới hạn
* D. Tối đa 10

1. Câu lệnh gọi nhiều middleware cho 1 route:

* A. app.get('/', m1 + m2)
* B. app.route('/', m1, m2)
* ⭐ C. app.get('/', m1, m2, handler)
* D. route('/', [m1, m2])

**71–80: Template và Giao diện**

1. File EJS có thể chứa:

* A. CSS
* B. JavaScript
* ⭐ C. JavaScript + HTML
* D. Chỉ text

1. Để hiển thị danh sách biến trong EJS, bạn dùng:

* A. forEach()
* ⭐ B. <% arr.forEach(...) %>
* C. {{ each }}
* D. <% arr.list() %>

1. Trong EJS, để viết logic điều kiện:

* A. if() trong JS
* ⭐ B. <% if (...) { %> ... <% } %>
* C. {{ if ... }}
* D. <: if ... >

1. Để tạo layout chung với EJS, bạn cần:

* A. HTML cơ bản
* B. Dùng body-parser
* ⭐ C. Dùng thư viện ejs-mate hoặc thủ công include
* D. Tạo nhiều views

1. res.render('view', { title: 'Hello' }) sẽ truyền biến nào?

* A. req.title
* B. res.title
* ⭐ C. title
* D. view.title

1. EJS thường đặt trong thư mục:

* A. static/
* ⭐ B. views/
* C. html/
* D. pages/

1. <%= %> trong EJS dùng để:

* A. Viết JS
* ⭐ B. Hiển thị dữ liệu
* C. Gọi route
* D. Tạo form

1. Để nhúng file EJS con vào file cha, dùng:

* A. <script>
* ⭐ B. <%- include('header') %>
* C. import
* D. require()

1. Nếu thiếu dữ liệu khi render, EJS sẽ:

* ⭐ A. Lỗi hoặc hiển thị trống
* B. Hiển thị mặc định
* C. Bỏ qua lỗi
* D. Hiển thị “undefined”

1. Thay đổi biến trong EJS phải làm ở:

* A. File .ejs
* ⭐ B. Trong route (truyền vào res.render)
* C. body-parser
* D. app.js

**81–90: Express Best Practices & Use Cases**

1. Express thường dùng nhất với:

* A. Frontend game
* ⭐ B. RESTful API backend
* C. Desktop GUI
* D. Email client

1. Express có thể mở rộng qua:

* ⭐ A. Middleware
* B. jQuery
* C. Browser plugin
* D. HTML tags

1. Để cấu trúc app lớn, nên:

* A. Viết tất cả trong app.js
* ⭐ B. Tách route, controller, model
* C. Gộp vào một route chính
* D. Dùng alert()

1. Express có hỗ trợ async/await không?

* ⭐ A. Có
* B. Không
* C. Chỉ trong middleware
* D. Tùy trình duyệt

1. Mô hình phổ biến cho app Express:

* A. Singleton
* ⭐ B. MVC
* C. Monolith
* D. PWA

1. Express có hỗ trợ WebSocket không?

* A. Không
* ⭐ B. Có, qua thư viện socket.io
* C. Có sẵn
* D. Qua HTML

1. Express xử lý JSON bằng:

* A. xml-parser
* ⭐ B. express.json()
* C. form-parser
* D. json-server

1. Express có thể dùng với TypeScript không?

* A. Không
* ⭐ B. Có, cần cấu hình
* C. Tùy phiên bản
* D. Chỉ với React

1. Để giữ trạng thái người dùng, Express dùng:

* A. log file
* B. cookies
* ⭐ C. sessions hoặc JWT
* D. GET param

1. Để xử lý form upload file, nên dùng:

* A. fs
* ⭐ B. multer
* C. ejs
* D. body-parser

**91–100: Bảo mật & Hiệu năng**

1. Middleware helmet dùng để:

* A. Tạo router
* ⭐ B. Tăng bảo mật HTTP headers
* C. Mã hóa dữ liệu
* D. Nén ảnh

1. Để chống spam request, nên dùng:

* A. helmet
* ⭐ B. express-rate-limit
* C. logger
* D. static()

1. Để mã hóa dữ liệu nhạy cảm, dùng:

* ⭐ A. bcrypt
* B. express
* C. helmet
* D. rate-limit

1. Để kiểm thử API Express, bạn dùng:

* A. Chrome
* B. Angular
* ⭐ C. Postman
* D. CSS

1. Express có log request không?

* A. Không
* ⭐ B. Có, dùng morgan
* C. Tùy route
* D. Cần server khác

1. Để tăng tốc độ nạp dữ liệu, bạn nên:

* A. Lưu tất cả vào DB
* ⭐ B. Dùng cache (Redis, memory)
* C. Dùng console.log
* D. Tạo nhiều router

1. Để xác thực người dùng, thường dùng:

* ⭐ A. passport.js
* B. ejs
* C. react
* D. socket.io

1. Khi deploy Express, cần quan tâm:

* A. Chỉ frontend
* ⭐ B. Bảo mật, port, cấu hình môi trường
* C. Cài winrar
* D. Dùng React

1. Express có thể chạy song song nhiều process bằng:

* A. thread()
* ⭐ B. cluster module
* C. setTimeout
* D. async()

1. Để đọc biến môi trường .env, dùng:

* A. ejs
* B. process.run()
* ⭐ C. dotenv
* D. fs.readFile

**Nội dung trọng tâm node.js**

**1. Giới thiệu Express.js**

* **Express.js** là một **web framework tối giản và linh hoạt** dành cho Node.js.
* Hỗ trợ xây dựng **ứng dụng web và di động** nhanh chóng, mạnh mẽ.
* Cung cấp các tính năng như:
  + **Templating (kết xuất HTML động)**
  + **Xử lý tệp tĩnh**
  + **Kết nối cơ sở dữ liệu (SQL & NoSQL)**​.

**✅ 2. Khởi tạo ứng dụng với Express**

* Cài đặt: npm install express
* Tạo app:

js

CopyEdit

const express = require('express');

const app = express();

app.listen(8000);

* Phản hồi “Hello World” khi truy cập trang chủ, còn lại trả về lỗi 404​.

**✅ 3. Routing (Định tuyến)**

* Xử lý HTTP methods: GET, POST, PUT, DELETE thông qua:

js

CopyEdit

app.get('/path', (req, res) => { ... });

* req là đối tượng yêu cầu, res là đối tượng phản hồi​.

**✅ 4. Middleware trong Express**

* **Middleware** là các hàm trung gian xử lý yêu cầu trước khi phản hồi.
* Ví dụ về middleware tích hợp:
  + express.static('public'): phục vụ tệp tĩnh như ảnh, CSS, JS.
  + body-parser: xử lý dữ liệu POST từ form.
  + cookie-parser, express-session: xử lý cookie và session​​.

**✅ 5. Template Engine - EJS**

* **EJS (Embedded JavaScript)** cho phép nhúng logic JavaScript vào HTML.
* Cài đặt: npm install ejs
* Cấu hình Express dùng EJS:

js

CopyEdit

app.set('view engine', 'ejs');

* Kết xuất HTML với dữ liệu:

js

CopyEdit

res.render('index', { name: 'John' });

* Dùng partials để chia nhỏ view (ví dụ: header.ejs, footer.ejs)​

**Quy Trình thiết kế web**

**I. CÁC KHÁI NIỆM CƠ BẢN**

1. **Trang web là gì?  
   A. Một phần mềm  
   B. Một loại cơ sở dữ liệu  
   C. Một trang nội dung trả về HTML  
   D. Một địa chỉ IP  
   Đáp án: C**
2. **Website là:  
   A. Một mạng xã hội  
   B. Một trình duyệt  
   C. Tập hợp các trang web có nội dung thống nhất  
   D. Một hệ điều hành  
   Đáp án: C**
3. **WWW là viết tắt của:  
   A. World Web Wide  
   B. Web World Wide  
   C. World Wide Web  
   D. Web Wide World  
   Đáp án: C**
4. **Web tĩnh thường sử dụng ngôn ngữ:  
   A. PHP  
   B. JavaScript  
   C. HTML  
   D. Python  
   Đáp án: C**
5. **Đặc điểm của web động là:  
   A. Tương tác yếu  
   B. Chỉ dùng HTML  
   C. Khó phát triển hơn web tĩnh  
   D. Không cần lập trình  
   Đáp án: C**

**II. CÁC GIAI ĐOẠN PHÁT TRIỂN WEBSITE**

1. **Giai đoạn đầu tiên trong quy trình phát triển web là:  
   A. Xây dựng  
   B. Thiết kế  
   C. Đặc tả  
   D. Phân tích  
   Đáp án: C**
2. **Giai đoạn nào xác định các chức năng và kỹ thuật cần sử dụng?  
   A. Triển khai  
   B. Phân tích  
   C. Bảo trì  
   D. Kiểm thử  
   Đáp án: B**
3. **Giai đoạn thiết kế bao gồm:  
   A. Mua tên miền  
   B. Tạo cơ sở dữ liệu  
   C. Sơ đồ cấu trúc, giao diện, CSDL  
   D. Tạo mã nguồn  
   Đáp án: C**
4. **Mục đích của kiểm thử website là:  
   A. Tăng tốc độ website  
   B. Quảng bá website  
   C. Phát hiện lỗi và kiểm tra tính tương thích  
   D. Viết thêm chức năng  
   Đáp án: C**
5. **Giai đoạn triển khai bao gồm việc:  
   A. Thiết kế giao diện  
   B. Phân tích hệ thống  
   C. Đưa website lên Internet  
   D. Tạo mockup  
   Đáp án: C**

**III. ĐẶC TẢ & PHÂN TÍCH**

1. **Câu hỏi không thuộc giai đoạn đặc tả là:  
   A. Ai dùng web?  
   B. Web để làm gì?  
   C. Kỹ thuật cần sử dụng?  
   D. Đối thủ là ai?  
   Đáp án: D**
2. **Trong giai đoạn phân tích, yếu tố nào sau đây được xem xét?  
   A. Màu sắc giao diện  
   B. Kỹ thuật sử dụng  
   C. Mức độ an toàn  
   D. Chi phí phát triển  
   Đáp án: B**
3. **Việc xác định “thứ tự các nội dung” thuộc giai đoạn nào?  
   A. Thiết kế  
   B. Đặc tả  
   C. Phân tích  
   D. Kiểm thử  
   Đáp án: C**

**IV. THIẾT KẾ WEBSITE**

1. **Giao diện website bao gồm:  
   A. Chỉ trang chính  
   B. Trang chính và trang phụ  
   C. Chỉ trang liên hệ  
   D. Trang chủ và footer  
   Đáp án: B**
2. **Cách trình bày nội dung nên:  
   A. Rối mắt và đa sắc  
   B. Đồng nhất và đa dạng  
   C. Tự phát  
   D. Dùng nhiều video  
   Đáp án: B**
3. **Điều hướng tuyến tính là:  
   A. Trang chủ trỏ đến tất cả các trang  
   B. Mỗi trang liên kết tuần tự với trang tiếp theo  
   C. Sử dụng ảnh làm bản đồ  
   D. Không có điều hướng  
   Đáp án: B**
4. **Điều hướng ánh xạ ảnh nghĩa là:  
   A. Sử dụng icon  
   B. Dùng hình ảnh có các vùng nhấp chuột  
   C. Menu dropdown  
   D. Giao diện tối giản  
   Đáp án: B**
5. **Siêu liên kết nên được thiết kế như thế nào?  
   A. Ẩn để tránh người dùng nhấn  
   B. Dễ nhìn và rõ ràng  
   C. Có hoạt ảnh phức tạp  
   D. Chỉ xuất hiện khi rê chuột  
   Đáp án: B**

**V. XÂY DỰNG & KIỂM THỬ**

1. **Một việc cần làm trong xây dựng website là:  
   A. Viết báo cáo thống kê  
   B. Cấu trúc thư mục  
   C. Quảng cáo website  
   D. Mua máy chủ  
   Đáp án: B**
2. **Kiểm thử nên thực hiện trên:  
   A. Một trình duyệt duy nhất  
   B. Mạng nội bộ  
   C. Nhiều trình duyệt và loại mạng khác nhau  
   D. Điện thoại di động  
   Đáp án: C**

**VI. TRIỂN KHAI WEBSITE**

1. **Tên miền có thể được sử dụng thông qua cách nào?  
   A. Đăng ký  
   B. Tự tạo  
   C. Copy từ web khác  
   D. Không cần thiết  
   Đáp án: A**
2. **Hosting miễn phí thường có hạn chế về:  
   A. Dung lượng và tính năng  
   B. Số lượng trang  
   C. Khả năng chia sẻ  
   D. Thiết kế  
   Đáp án: A**
3. **Máy chủ Windows hỗ trợ:  
   A. PHP, JSP  
   B. PHP, ASP  
   C. JSP, Perl  
   D. Python, HTML  
   Đáp án: B**
4. **Máy chủ Linux rẻ hơn do:  
   A. Không có bảo mật  
   B. Không hỗ trợ SQL  
   C. Dễ dàng tùy biến  
   D. Phí bản quyền thấp hơn  
   Đáp án: D**

**VII. DUY TRÌ VÀ PHÁT TRIỂN WEBSITE**

1. **Web tĩnh được cập nhật qua:  
   A. Giao diện quản trị  
   B. Cơ sở dữ liệu  
   C. Web browser hoặc FTP  
   D. Email  
   Đáp án: C**
2. **Web động cập nhật thông tin qua:  
   A. FTP  
   B. Form cập nhật CSDL  
   C. File Excel  
   D. Google Drive  
   Đáp án: B**
3. **Một cách nâng tầm phát triển website là:  
   A. Giảm chức năng  
   B. Dùng màu sặc sỡ  
   C. Tự động hóa chức năng  
   D. Xoá liên kết  
   Đáp án: C**

**VIII. MARKETING WEBSITE**

1. **Cách marketing nào dưới đây là hiệu quả?  
   A. Trao đổi banner  
   B. Spam diễn đàn  
   C. Viết blog ngẫu nhiên  
   D. Không cần quảng bá  
   Đáp án: A**
2. **Một công cụ đăng ký website trên thế giới là:  
   A. Youtube  
   B. Twitter  
   C. Google  
   D. Zalo  
   Đáp án: C**
3. **Similarweb dùng để:  
   A. Thiết kế website  
   B. Xếp hạng traffic  
   C. Mua tên miền  
   D. Tạo nội dung  
   Đáp án: B**

**IX. PHÁT TRIỂN, MỞ RỘNG, CẬP NHẬT WEBSITE**

1. **Một yếu tố giúp tăng trải nghiệm người dùng là:  
   A. Menu điều hướng rõ ràng  
   B. Tải chậm để xem kỹ  
   C. Nền tối và chữ mờ  
   D. Nhiều cửa sổ bật lên  
   Đáp án: A**
2. **Khi cập nhật website, cần chú ý đến:  
   A. Thời tiết  
   B. Sự ổn định của cấu trúc  
   C. Tên miền mới  
   D. Tăng màu sắc  
   Đáp án: B**
3. **Một website chuẩn SEO cần có:  
   A. Hình ảnh lớn  
   B. Nội dung tối ưu từ khóa  
   C. Video tự động chạy  
   D. File PDF đính kèm  
   Đáp án: B**
4. **Khi nâng cấp website, không nên:  
   A. Tối ưu hiệu năng  
   B. Bỏ các chức năng cần thiết  
   C. Cải thiện giao diện  
   D. Đảm bảo bảo mật  
   Đáp án: B**
5. **Bảo trì website định kỳ nhằm:  
   A. Giảm chi phí  
   B. Bỏ bớt dữ liệu  
   C. Phát hiện và sửa lỗi sớm  
   D. Tăng thời gian chết  
   Đáp án: C**

**X. QUẢN TRỊ & BẢO MẬT WEBSITE**

1. **Một trong các rủi ro lớn nhất với website là:  
   A. Hết hạn tên miền  
   B. Đổi ngôn ngữ  
   C. Tấn công mạng  
   D. Giao diện xấu  
   Đáp án: C**
2. **Firewall (tường lửa) dùng để:  
   A. Cản bot tìm kiếm  
   B. Chống xâm nhập trái phép  
   C. Tăng tốc độ tải  
   D. Phân quyền người dùng  
   Đáp án: B**
3. **Hệ thống phân quyền giúp:  
   A. Giảm tính bảo mật  
   B. Phân chia dữ liệu  
   C. Giới hạn quyền truy cập của người dùng  
   D. Làm website hấp dẫn hơn  
   Đáp án: C**
4. **Sao lưu dữ liệu website nên được thực hiện:  
   A. Một lần/năm  
   B. Khi có sự cố  
   C. Định kỳ  
   D. Chỉ khi cập nhật phiên bản  
   Đáp án: C**
5. **Công cụ mã hóa thường dùng để:  
   A. Nén ảnh  
   B. Bảo vệ dữ liệu khi truyền  
   C. Tăng tốc độ xử lý  
   D. Tự động đăng bài  
   Đáp án: B**

**XI. ĐÁNH GIÁ & PHÂN TÍCH HIỆU SUẤT WEBSITE**

1. **Một tiêu chí đánh giá hiệu quả website:  
   A. Thời gian tải trang  
   B. Số lượng ảnh nền  
   C. Số lượng lập trình viên  
   D. Màu sắc chủ đạo  
   Đáp án: A**
2. **Google Analytics giúp:  
   A. Tạo giao diện web  
   B. Theo dõi lượng truy cập  
   C. Lập trình trang web  
   D. Kiểm tra lỗi chính tả  
   Đáp án: B**
3. **Bounce rate cao nghĩa là:  
   A. Nhiều người truy cập lâu  
   B. Người dùng rời ngay sau khi vào trang  
   C. Website được yêu thích  
   D. Tốc độ tải nhanh  
   Đáp án: B**
4. **Thời gian tải trang lý tưởng nên dưới:  
   A. 10 giây  
   B. 6 giây  
   C. 3 giây  
   D. 15 giây  
   Đáp án: C**
5. **Tối ưu ảnh trên website giúp:  
   A. Giảm chất lượng giao diện  
   B. Tăng tốc độ tải  
   C. Làm ảnh đẹp hơn  
   D. Tăng chi phí hosting  
   Đáp án: B**

**XII. CÂU HỎI TỔNG HỢP NÂNG CAO**

1. **Việc tạo sơ đồ cấu trúc trang giúp:  
   A. Làm đẹp web  
   B. Hỗ trợ SEO và định hướng phát triển  
   C. Giảm dữ liệu  
   D. Tăng tính nghệ thuật  
   Đáp án: B**
2. **Thiết kế responsive nghĩa là:  
   A. Web chỉ chạy trên điện thoại  
   B. Web tự điều chỉnh theo thiết bị  
   C. Web có hiệu ứng động  
   D. Web dùng PHP  
   Đáp án: B**
3. **Một chiến lược phát triển lâu dài cho website là:  
   A. Mở rộng nội dung, dịch vụ và tương tác  
   B. Giữ nguyên sau khi xây xong  
   C. Giảm bài viết  
   D. Dùng kỹ thuật cũ  
   Đáp án: A**
4. **Phản hồi người dùng giúp:  
   A. Gây nhiễu thông tin  
   B. Tăng độ tương tác và cải tiến dịch vụ  
   C. Giảm tốc độ  
   D. Bảo mật website  
   Đáp án: B**
5. **Tại sao cần đánh giá lại website định kỳ?  
   A. Do nội dung bị mất  
   B. Để phát hiện lỗi, nâng cấp kịp thời  
   C. Vì thay tên miền  
   D. Vì hình ảnh xấu  
   Đáp án: B**

**XIII. CÁC KHÁI NIỆM & PHÂN TÍCH BỔ SUNG**

1. **Một website được đánh giá là thân thiện với người dùng khi:  
   A. Nhiều nội dung trên một trang  
   B. Dễ điều hướng, giao diện rõ ràng  
   C. Ít màu sắc  
   D. Chứa nhiều quảng cáo  
   Đáp án: B**
2. **Phân tích chức năng website giúp:  
   A. Xác định màu nền  
   B. Chọn ngôn ngữ lập trình  
   C. Biết rõ yêu cầu hệ thống và người dùng  
   D. Tạo tên miền  
   Đáp án: C**
3. **Trong phân tích kỹ thuật, ta không cần xét đến:  
   A. Ngôn ngữ lập trình  
   B. Kiểu máy tính người dùng  
   C. Công nghệ cơ sở dữ liệu  
   D. Tính tương thích trình duyệt  
   Đáp án: B**
4. **Web động có ưu điểm lớn nhất là:  
   A. Tốc độ nhanh  
   B. Giao diện đơn giản  
   C. Có khả năng tương tác và cập nhật dễ  
   D. Không cần bảo trì  
   Đáp án: C**
5. **Một website tốt nên có sơ đồ điều hướng để:  
   A. Người dùng dễ tìm thông tin  
   B. Tăng lượng mã nguồn  
   C. Giảm chi phí  
   D. Thay thế menu chính  
   Đáp án: A**

**XIV. THIẾT KẾ GIAO DIỆN & ĐIỀU HƯỚNG**

1. **Một nguyên tắc trong thiết kế giao diện web:  
   A. Dùng càng nhiều màu càng tốt  
   B. Ưu tiên tính thẩm mỹ hơn nội dung  
   C. Giữ bố cục rõ ràng và thống nhất  
   D. Dùng hiệu ứng động liên tục  
   Đáp án: C**
2. **Thiết kế nhất quán về giao diện sẽ giúp:  
   A. Giảm trải nghiệm người dùng  
   B. Làm web nhàm chán  
   C. Tạo cảm giác chuyên nghiệp, dễ sử dụng  
   D. Tăng chi phí thiết kế  
   Đáp án: C**
3. **Sử dụng hệ thống phân cấp thông tin giúp:  
   A. Tạo độ khó cho người dùng  
   B. Dễ quản lý nội dung  
   C. Rối cấu trúc web  
   D. Loại bỏ các liên kết  
   Đáp án: B**
4. **Điều hướng phân cấp là:  
   A. Mỗi trang liên kết trực tiếp với trang chủ  
   B. Có cấu trúc cha – con giữa các trang  
   C. Dùng ảnh làm bản đồ  
   D. Chỉ có một trang chính  
   Đáp án: B**
5. **Thanh điều hướng ngang thường đặt ở:  
   A. Cuối trang  
   B. Bên trái màn hình  
   C. Trên đầu trang  
   D. Trong ảnh nền  
   Đáp án: C**

**XV. XÂY DỰNG VÀ KIỂM THỬ**

1. **HTML dùng để:  
   A. Tạo logic xử lý  
   B. Kiểm tra bảo mật  
   C. Xây dựng cấu trúc nội dung  
   D. Tối ưu hóa máy chủ  
   Đáp án: C**
2. **CSS giúp:  
   A. Lưu trữ dữ liệu  
   B. Thiết kế giao diện và định dạng trang  
   C. Thêm chức năng động  
   D. Tạo liên kết  
   Đáp án: B**
3. **JavaScript thường được dùng để:  
   A. Viết giao diện tĩnh  
   B. Tăng tương tác người dùng  
   C. Tạo tên miền  
   D. Thiết kế CSDL  
   Đáp án: B**
4. **Trong quá trình xây dựng website, file nào chứa nội dung chính?  
   A. .exe  
   B. .html  
   C. .xls  
   D. .jpg  
   Đáp án: B**
5. **Việc kiểm thử trên nhiều thiết bị giúp:  
   A. Tăng tính tương thích và ổn định  
   B. Tăng chi phí không cần thiết  
   C. Làm rối đội ngũ thiết kế  
   D. Chỉ cần thiết cho web động  
   Đáp án: A**

**XVI. TRIỂN KHAI & HOẠT ĐỘNG WEBSITE**

1. **Tại sao cần chọn tên miền dễ nhớ?  
   A. Giúp tăng thứ hạng tìm kiếm  
   B. Dễ quảng bá và ghi nhớ  
   C. Rẻ hơn  
   D. Không bị trùng IP  
   Đáp án: B**
2. **FTP dùng để:  
   A. Thiết kế website  
   B. Chạy thử offline  
   C. Chuyển file lên hosting  
   D. Phân tích SEO  
   Đáp án: C**
3. **Khi chọn dịch vụ hosting, cần quan tâm:  
   A. Số lượng nhân viên  
   B. Vị trí địa lý  
   C. Dung lượng, băng thông, tốc độ  
   D. Màu logo  
   Đáp án: C**
4. **Công cụ giúp người không biết lập trình vẫn cập nhật website là:  
   A. Notepad  
   B. Control Panel  
   C. CMS (như WordPress)  
   D. Photoshop  
   Đáp án: C**
5. **Sau khi triển khai, bước quan trọng tiếp theo là:  
   A. Quên nó đi  
   B. Bảo trì và cập nhật thường xuyên  
   C. Chỉnh sửa giao diện  
   D. Thay đổi tên miền  
   Đáp án: B**