**JavaScript Quiz**

**Easy**

1. **What is the output of console.log(1 + 1);?  
   a) 1  
   b) 11  
   c) 2  
   d) undefined**
2. **How do you declare a variable in JavaScript?  
   a) var x;  
   b) x = 10;  
   c) let 10;  
   d) variable x;**
3. **Which method is used to add an element at the end of an array?  
   a) push()  
   b) pop()  
   c) shift()  
   d) unshift()**
4. **Which of the following is a valid JavaScript function?  
   a) function myFunction()  
   b) myFunction()  
   c) function: myFunction  
   d) function myFunction[]**
5. **How do you write a single-line comment in JavaScript?  
   a) <!-- comment -->  
   b) # comment  
   c) /\* comment \*/  
   d) // comment**
6. **What is the result of typeof(5)?  
   a) object  
   b) number  
   c) string  
   d) undefined**
7. **How do you check if a number is an integer in JavaScript?  
   a) isInteger()  
   b) Number.isInteger()  
   c) parseInt()  
   d) integer()**
8. **Which symbol is used for strict equality comparison in JavaScript?  
   a) ==  
   b) ===  
   c) !=  
   d) !==**
9. **How do you define an array in JavaScript?  
   a) var arr = (1, 2, 3);  
   b) arr = [1, 2, 3];  
   c) array = {1, 2, 3};  
   d) let arr = "1,2,3";**
10. **What does the console.log() function do?  
    a) Logs the error  
    b) Logs a variable's type  
    c) Outputs data to the console  
    d) Stops the program**
11. **Which method is used to remove the first element of an array in JavaScript?  
    a) pop()  
    b) shift()  
    c) push()  
    d) unshift()**
12. **How do you define an object in JavaScript?  
    a) var obj = {};  
    b) let obj = [];  
    c) var obj = () {};  
    d) const obj = '';**
13. **What is the result of console.log(5 + '5') in JavaScript?  
    a) 55  
    b) 10  
    c) NaN  
    d) undefined**
14. **What does the parseInt() function do?  
    a) Converts a string to an integer  
    b) Converts a string to a float  
    c) Converts an integer to a string  
    d) Converts an array to a string**
15. **What does the typeof operator do?  
    a) Returns the type of a variable  
    b) Converts a variable to a type  
    c) Declares a variable type  
    d) None of the above**
16. **What is the output of console.log(2 == '2')?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**
17. **How do you add a comment that spans multiple lines in JavaScript?  
    a) /\* comment \*/  
    b) <!-- comment -->  
    c) // comment //  
    d) # comment**
18. **Which method is used to combine two arrays in JavaScript?  
    a) merge()  
    b) join()  
    c) concat()  
    d) push()**
19. **How do you create a constant in JavaScript?  
    a) var x = 10;  
    b) let x = 10;  
    c) constant x = 10;  
    d) const x = 10;**
20. **What is the result of console.log(0 == false)?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**

**Medium**

1. **What does the this keyword refer to in JavaScript?  
   a) The global object  
   b) The function itself  
   c) The parent object  
   d) The current object**
2. **How do you create an anonymous function in JavaScript?  
   a) function() {}  
   b) function name() {}  
   c) function(name) {}  
   d) anonymous() => {}**
3. **What is the output of console.log([] == false);?  
   a) true  
   b) false  
   c) undefined  
   d) NaN**
4. **What does JSON.stringify() do in JavaScript?  
   a) Converts a JSON string to an object  
   b) Converts an object to a JSON string  
   c) Converts a string to JSON  
   d) Converts JSON to an array**
5. **How do you check if an array includes a value in JavaScript?  
   a) array.includes(value)  
   b) array.contains(value)  
   c) array.has(value)  
   d) array.in(value)**
6. **What is the difference between null and undefined in JavaScript?  
   a) null is a primitive value, and undefined is an object  
   b) null represents an uninitialized variable, and undefined represents an unknown object  
   c) null represents the absence of a value, while undefined represents an uninitialized variable  
   d) There is no difference**
7. **How do you convert a string to an integer in JavaScript?  
   a) Number(string)  
   b) parseInt(string)  
   c) parseFloat(string)  
   d) Both a and b**
8. **Which of the following is NOT a JavaScript loop?  
   a) for  
   b) while  
   c) loop  
   d) do...while**
9. **What does the slice() method do in JavaScript?  
   a) Removes an element from the beginning of an array  
   b) Adds an element at the beginning of an array  
   c) Extracts a section of an array  
   d) Reverses the elements of an array**
10. **How do you handle errors in JavaScript?  
    a) try/catch  
    b) throw/catch  
    c) catch/throw  
    d) error/catch**
11. **What is the output of console.log([1] == true)?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**
12. **What is the purpose of setTimeout() in JavaScript?  
    a) Executes a function after a delay  
    b) Sets a timer for a loop  
    c) Sets a delay for an array  
    d) None of the above**
13. **What does the call() method do in JavaScript?  
    a) Executes a function with a specified this value  
    b) Calls a function after a delay  
    c) Calls a function repeatedly  
    d) None of the above**
14. **What is the output of console.log(1 + '1') in JavaScript?  
    a) 11  
    b) 2  
    c) 1  
    d) NaN**
15. **What is the purpose of the map() method in JavaScript?  
    a) It creates a new array with the results of applying a function to each element  
    b) It filters an array  
    c) It reduces an array to a single value  
    d) It adds an element to an array**
16. **Which of the following methods is used to add a new element at the start of an array?  
    a) push()  
    b) unshift()  
    c) shift()  
    d) pop()**
17. **How do you create a promise in JavaScript?  
    a) new Promise()  
    b) Promise.create()  
    c) promise = new  
    d) createPromise()**
18. **What is the purpose of Object.freeze() in JavaScript?  
    a) Makes an object immutable  
    b) Converts an object to a string  
    c) Deletes an object  
    d) Creates a shallow copy of an object**
19. **How do you define a function that returns a promise in JavaScript?  
    a) async function name() {}  
    b) function async name() {}  
    c) name() = async function {}  
    d) async function {}**
20. **What is the output of console.log([] == false) in JavaScript?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**

**Hard**

1. **What is a closure in JavaScript?  
   a) A function that runs immediately after being declared  
   b) A function that retains access to its lexical scope even when invoked outside that scope  
   c) A function used to create objects  
   d) A type of loop**
2. **What is the output of console.log(3 > 2 > 1)?  
   a) true  
   b) false  
   c) undefined  
   d) NaN**
3. **What is the result of console.log(1 + true) in JavaScript?  
   a) 2  
   b) NaN  
   c) 1true  
   d) undefined**
4. **What does the bind() method do in JavaScript?  
   a) Binds a function to a specific context  
   b) Binds an event listener to an element  
   c) Binds a value to a function  
   d) Binds a property to a variable**
5. **How do you handle asynchronous code in JavaScript?  
   a) Using setTimeout()  
   b) Using promises and async/await  
   c) Using try/catch  
   d) None of the above**
6. **What is the difference between == and === in JavaScript?  
   a) == checks for equality of values, === checks for both value and type equality  
   b) == checks for strict equality, === is for loose equality  
   c) == is for string comparisons, === is for number comparisons  
   d) There is no difference**
7. **How do you declare a generator function in JavaScript?  
   a) function\* name() {}  
   b) generator function name() {}  
   c) function name\*() {}  
   d) name() = function\* {}**
8. **What does the setInterval() method do in JavaScript?  
   a) Sets a time limit for a function  
   b) Repeats a function call at specified intervals  
   c) Delays the execution of a function  
   d) Sets a timeout for a function**
9. **What is the purpose of Promise.all() in JavaScript?  
   a) Executes all promises sequentially  
   b) Executes all promises simultaneously  
   c) Returns the result of the first promise  
   d) None of the above**
10. **What is the result of console.log([1] == [1])?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**
11. **What is the output of console.log([] + [])?  
    a) "" (empty string)  
    b) []  
    c) NaN  
    d) undefined**
12. **What does eval() do in JavaScript?  
    a) Executes JavaScript code represented as a string  
    b) Evaluates a mathematical expression  
    c) Evaluates a function  
    d) None of the above**
13. **How can you create a deep copy of an object in JavaScript?  
    a) Object.assign()  
    b) JSON.parse(JSON.stringify())  
    c) Object.clone()  
    d) Array.slice()**
14. **What is the difference between let and var in JavaScript?  
    a) let is block-scoped, var is function-scoped  
    b) let allows redeclaration, var does not  
    c) let is hoisted, var is not  
    d) There is no difference**
15. **How do you create an immutable object in JavaScript?  
    a) Use Object.freeze()  
    b) Use const  
    c) Use const and Object.preventExtensions()  
    d) Both a and c**
16. **What is the output of console.log(typeof NaN)?  
    a) number  
    b) NaN  
    c) undefined  
    d) object**
17. **What is the result of console.log([2] == [2]) in JavaScript?  
    a) true  
    b) false  
    c) undefined  
    d) NaN**
18. **What is the purpose of Promise.race() in JavaScript?  
    a) Executes promises in order  
    b) Returns the result of the first promise to resolve  
    c) Returns the first promise to reject  
    d) None of the above**
19. **What is the result of console.log(!!'false') in JavaScript?  
    a) false  
    b) true  
    c) undefined  
    d) NaN**
20. **What is the purpose of new in JavaScript?  
    a) Creates a new instance of an object  
    b) Creates a new array  
    c) Declares a variable  
    d) None of the above**

**CSS Quiz**

**Easy**

1. **What does CSS stand for?  
   a) Computer Style Sheets  
   b) Cascading Style Sheets  
   c) Creative Style Sheets  
   d) Colorful Style Sheets**
2. **How do you define a CSS rule?  
   a) selector {property: value;}  
   b) property: value {selector;}  
   c) value: property {selector;}  
   d) selector: value {property;}**
3. **Which of the following is used to apply CSS styles to an HTML element?  
   a) class  
   b) id  
   c) style  
   d) All of the above**
4. **Which CSS property is used to change the text color of an element?  
   a) color  
   b) text-color  
   c) font-color  
   d) text-style**
5. **What does the padding property do in CSS?  
   a) Adds space inside an element  
   b) Adds space outside an element  
   c) Changes the border color  
   d) Changes the text color**
6. **How can you center an element horizontally in CSS?  
   a) margin: 0 auto;  
   b) text-align: center;  
   c) align: center;  
   d) center: auto;**
7. **Which property is used to change the background color of an element in CSS?  
   a) color  
   b) bgcolor  
   c) background-color  
   d) bgcolor-color**
8. **How do you select all <p> elements in a CSS file?  
   a) p {}  
   b) .p {}  
   c) #p {}  
   d) p[]**
9. **Which of the following is the correct way to add a CSS comment?  
   a) // This is a comment  
   b) <!-- This is a comment -->  
   c) /\* This is a comment \*/  
   d) # This is a comment**
10. **How do you make a font bold using CSS?  
    a) font-weight: bold;  
    b) font-style: bold;  
    c) text-weight: bold;  
    d) text-style: bold;**
11. **Which property is used to change the font of an element?  
    a) font-family  
    b) font-style  
    c) font-size  
    d) font-weight**
12. **Which CSS property is used to add space between the content and the border of an element?  
    a) margin  
    b) padding  
    c) border-spacing  
    d) gap**
13. **What does the display: block; property do?  
    a) Makes an element appear in its own line  
    b) Makes an element inline  
    c) Hides an element  
    d) Makes the element a flex container**
14. **How do you change the width of an element in CSS?  
    a) width: 100px;  
    b) size: 100px;  
    c) element-width: 100px;  
    d) length: 100px;**
15. **How can you add a border to an element in CSS?  
    a) border: solid 1px black;  
    b) border-style: solid;  
    c) border-width: 1px;  
    d) box-border: solid;**
16. **Which of the following is used to add space between the elements in a flex container?  
    a) space-between  
    b) gap  
    c) padding  
    d) margin**
17. **How do you apply a background image in CSS?  
    a) background: url('image.jpg');  
    b) background-image: url('image.jpg');  
    c) image-background: url('image.jpg');  
    d) background-color: url('image.jpg');**
18. **How can you make a hyperlink bold using CSS?  
    a) a {font-weight: bold;}  
    b) a {text-weight: bold;}  
    c) a {font-style: bold;}  
    d) a {text-decoration: bold;}**
19. **How do you apply styles to an element with an id of "header"?  
    a) #header {}  
    b) .header {}  
    c) header {}  
    d) [id="header"] {}**
20. **What is the correct CSS syntax to select an element with a class of "container"?  
    a) .container {}  
    b) #container {}  
    c) container {}  
    d) [class="container"] {}**

**Medium**

1. **How do you center a block element vertically in CSS?  
   a) margin: 0 auto;  
   b) vertical-align: middle;  
   c) display: flex; align-items: center;  
   d) text-align: center;**
2. **What is the purpose of the z-index property in CSS?  
   a) Controls the stacking order of elements  
   b) Controls the transparency of an element  
   c) Defines the border of an element  
   d) Changes the position of an element**
3. **How do you hide an element in CSS?  
   a) display: none;  
   b) visibility: hidden;  
   c) opacity: 0;  
   d) All of the above**
4. **What does the box-sizing property do in CSS?  
   a) Defines the size of the content box  
   b) Defines the size of the padding  
   c) Includes padding and border in the element's total width and height  
   d) Adds a border to the element**
5. **What is the difference between inline and block elements in CSS?  
   a) Block elements take up the full width, inline elements take up only as much width as needed  
   b) Inline elements can have padding, block elements cannot  
   c) Block elements cannot have margins, inline elements can  
   d) There is no difference**
6. **How do you apply a CSS transition to an element?  
   a) transition: all 0.5s ease;  
   b) transition-effect: 0.5s ease;  
   c) animate: all 0.5s ease;  
   d) effect: 0.5s ease;**
7. **How do you set the height of an element to be equal to the viewport height?  
   a) height: 100vh;  
   b) height: 100%;  
   c) height: auto;  
   d) height: 100px;**
8. **How can you create a flex container in CSS?  
   a) display: flex;  
   b) display: grid;  
   c) display: inline-block;  
   d) display: table;**
9. **What is the @media rule used for in CSS?  
   a) To apply CSS rules based on conditions like screen size or device type  
   b) To create a media player  
   c) To create animations  
   d) To embed audio and video**
10. **Which of the following is the correct way to make a flex container's items wrap?  
    a) flex-wrap: wrap;  
    b) wrap: flex;  
    c) wrap-items: true;  
    d) flex-items: wrap;**
11. **What is the CSS property used to define the space between columns in a grid layout?  
    a) column-gap  
    b) grid-gap  
    c) gap  
    d) space-between**
12. **How can you create a circular shape with CSS?  
    a) border-radius: 50%;  
    b) border-radius: 100%;  
    c) shape: circle;  
    d) border: solid 50%**
13. **How do you create a sticky navbar in CSS?  
    a) position: sticky; top: 0;  
    b) position: fixed; top: 0;  
    c) position: absolute; top: 0;  
    d) position: relative; top: 0;**
14. **What is the CSS property used to create shadow effects on text?  
    a) text-shadow  
    b) shadow-effect  
    c) box-shadow  
    d) element-shadow**
15. **What is the purpose of the flex-direction property in CSS?  
    a) Determines the direction of flex items in a container  
    b) Defines the size of flex items  
    c) Aligns items vertically  
    d) Sets the alignment of the container itself**
16. **How do you add a background image that covers the entire element?  
    a) background-size: cover;  
    b) background-repeat: no-repeat;  
    c) background-size: 100%;  
    d) background-position: center;**
17. **What is the default value of position property in CSS?  
    a) absolute  
    b) relative  
    c) static  
    d) fixed**
18. **How do you create a CSS grid layout with three equal columns?  
    a) grid-template-columns: repeat(3, 1fr);  
    b) grid-template-columns: 33% 33% 33%;  
    c) grid-template-columns: 1fr 1fr 1fr;  
    d) All of the above**
19. **What is the purpose of overflow property in CSS?  
    a) Controls the visibility of content that overflows its container  
    b) Controls the width of a container  
    c) Controls the flow of text inside an element  
    d) None of the above**
20. **How do you align an item to the center of a flex container?  
    a) justify-content: center; align-items: center;  
    b) text-align: center;  
    c) align-content: center;  
    d) margin: auto;**

**Hard**

1. **What is the difference between position: absolute; and position: fixed; in CSS?  
   a) absolute is relative to its nearest positioned ancestor, while fixed is relative to the viewport  
   b) absolute is relative to the viewport, while fixed is relative to the parent element  
   c) absolute makes the element visible, while fixed hides it  
   d) There is no difference**
2. **How do you implement a CSS animation with keyframes?  
   a) @keyframes animation {}  
   b) @animation keyframes {}  
   c) @keyframe animation {}  
   d) @animation {}**
3. **What is the transform property used for in CSS?  
   a) To scale, rotate, or translate elements  
   b) To set the background of an element  
   c) To add padding and margin  
   d) To align elements**
4. **How do you specify that an element should be a flex container with items aligned both horizontally and vertically in the center?  
   a) display: flex; justify-content: center; align-items: center;  
   b) display: grid; justify-items: center; align-items: center;  
   c) display: flex; align-content: center; justify-content: space-between;  
   d) display: block; text-align: center; vertical-align: middle;**
5. **What is the difference between rem and em units in CSS?  
   a) rem is relative to the root font size, em is relative to the parent element's font size  
   b) rem is relative to the parent element, em is relative to the root font size  
   c) rem and em are the same  
   d) There is no difference**
6. **How do you achieve a fluid layout using CSS?  
   a) By using percentage-based widths and max-width  
   b) By setting fixed pixel widths  
   c) By using display: flex; only  
   d) By applying margin: auto; to all elements**
7. **What is the effect of using @import in CSS?  
   a) It imports external stylesheets into the current stylesheet  
   b) It defines a new CSS rule  
   c) It imports JavaScript into CSS  
   d) It creates an animation**
8. **How do you create a responsive design for multiple screen sizes in CSS?  
   a) Using @media queries  
   b) Using flexbox  
   c) Using viewport units  
   d) Both a and c**
9. **How do you make an element responsive to different screen sizes using CSS Grid?  
   a) By using grid-template-columns: repeat(auto-fit, minmax(100px, 1fr));  
   b) By using grid-template-columns: 100px;  
   c) By using display: block;  
   d) By using max-width: 100%;**
10. **How do you prevent text from overflowing its container in CSS?  
    a) overflow: hidden; text-overflow: ellipsis;  
    b) overflow: auto; text-overflow: ellipsis;  
    c) white-space: nowrap;  
    d) max-width: 100%;**
11. **How do you create a parallax scrolling effect using CSS?  
    a) By using background-attachment: fixed;  
    b) By using transform: translateY();  
    c) By using scroll-behavior: smooth;  
    d) By using position: absolute;**
12. **How do you make a grid layout with equal rows in CSS?  
    a) grid-template-rows: repeat(3, 1fr);  
    b) grid-template-rows: 100px;  
    c) grid-template-columns: 1fr;  
    d) grid-template: 1fr 1fr 1fr;**
13. **What is the effect of using opacity: 0; on an element in CSS?  
    a) It hides the element, but it remains clickable  
    b) It makes the element visible with no effect  
    c) It removes the element completely from the document  
    d) It changes the element's visibility to none**
14. **What is a container query in CSS used for?  
    a) To apply styles based on the size of a parent element  
    b) To create responsive typography  
    c) To add animations to an element  
    d) To hide elements on large screens**
15. **How do you change the color of a link when it is clicked in CSS?  
    a) a:active { color: red; }  
    b) a:clicked { color: red; }  
    c) a:hover { color: red; }  
    d) a:focus { color: red; }**
16. **What is the effect of using visibility: hidden; on an element in CSS?  
    a) The element becomes invisible but still occupies space in the layout  
    b) The element is removed from the layout entirely  
    c) The element becomes invisible and occupies no space  
    d) The element becomes opaque**
17. **How do you animate a CSS property?  
    a) @keyframes with animation property  
    b) @animate with transition property  
    c) @keyframes with transition property  
    d) @transition with animation property**
18. **How do you make an image responsive in CSS?  
    a) img { width: 100%; height: auto; }  
    b) img { max-width: 100%; }  
    c) img { width: 100%; }  
    d) Both a and b**
19. **What is the purpose of object-fit: cover; in CSS?  
    a) It scales the image to cover the element's box while maintaining its aspect ratio  
    b) It stretches the image to fit the element's box  
    c) It makes the image transparent  
    d) It removes the background of the image**
20. **How do you use the clamp() function in CSS?  
    a) To set a value that adjusts between a defined minimum and maximum range based on viewport size  
    b) To limit the height of an element  
    c) To set fixed font sizes  
    d) To control the width of a container**

**HTML Quiz**

**Easy**

1. **What does HTML stand for?  
   a) Hyper Text Markup Language  
   b) Home Tool Markup Language  
   c) Hyperlinks and Text Markup Language  
   d) None of the above**
2. **Which HTML tag is used to define an unordered list?  
   a) <ul>  
   b) <ol>  
   c) <li>  
   d) <list>**
3. **Which tag is used to create a hyperlink in HTML?  
   a) <a>  
   b) <link>  
   c) <href>  
   d) <hyperlink>**
4. **What is the correct HTML element for inserting an image?  
   a) <image>  
   b) <img>  
   c) <src>  
   d) <picture>**
5. **What is the purpose of the <title> tag in HTML?  
   a) Defines the title of a document in the browser's title bar or tab  
   b) Sets the text for an element  
   c) Creates a heading  
   d) Defines a tooltip**
6. **Which tag is used to create a line break in HTML?  
   a) <br>  
   b) <hr>  
   c) <break>  
   d) <lb>**
7. **Which attribute is used to specify the source of an image in HTML?  
   a) source  
   b) href  
   c) src  
   d) img-source**
8. **How do you create a table in HTML?  
   a) <table>  
   b) <tabular>  
   c) <tbl>  
   d) <grid>**
9. **What does the <p> tag represent in HTML?  
   a) Paragraph  
   b) Parent element  
   c) Pointer  
   d) Page link**
10. **Which element is used to define the main content of an HTML page?  
    a) <header>  
    b) <main>  
    c) <footer>  
    d) <section>**
11. **How can you make text bold in HTML?  
    a) <bold>  
    b) <strong>  
    c) <b>  
    d) Both b and c**
12. **Which tag is used to define a link in HTML?  
    a) <a>  
    b) <link>  
    c) <href>  
    d) <url>**
13. **How do you create an ordered list in HTML?  
    a) <ol>  
    b) <ul>  
    c) <li>  
    d) <list>**
14. **Which tag is used for creating a form in HTML?  
    a) <form>  
    b) <input>  
    c) <button>  
    d) <field>**
15. **How do you make a comment in HTML?  
    a) <!-- comment -->  
    b) // comment  
    c) /\* comment \*/  
    d) # comment**
16. **Which of the following is used to create a drop-down list in HTML?  
    a) <input>  
    b) <select>  
    c) <dropdown>  
    d) <option>**
17. **Which tag is used to define a header in HTML?  
    a) <h1>  
    b) <head>  
    c) <header>  
    d) <heading>**
18. **What does the alt attribute do for images in HTML?  
    a) Provides a description of the image for accessibility  
    b) Changes the color of the image  
    c) Alters the size of the image  
    d) None of the above**
19. **How do you create a hyperlink that opens in a new tab?  
    a) <a href="url" target="\_blank">  
    b) <a href="url" new-window="true">  
    c) <a href="url" open="true">  
    d) <a href="url" window="new">**
20. **Which tag is used to add a video in HTML?  
    a) <video>  
    b) <media>  
    c) <video-source>  
    d) <movie>**

**Medium**

1. **What is the difference between the <div> and <span> tags?  
   a) <div> is a block-level element, while <span> is an inline element  
   b) <span> is a block-level element, while <div> is an inline element  
   c) They are the same  
   d) None of the above**
2. **How do you add a background color to a webpage in HTML?  
   a) <background-color>  
   b) <body style="background-color: color;">  
   c) <style="background-color: color;">  
   d) <color background="color">**
3. **Which tag is used to define a container for navigation links in HTML?  
   a) <nav>  
   b) <links>  
   c) <navigation>  
   d) <header>**
4. **How do you link an external CSS file in HTML?  
   a) <css href="styles.css">  
   b) <style href="styles.css">  
   c) <link rel="stylesheet" href="styles.css">  
   d) <stylesheet src="styles.css">**
5. **What does the target="\_self" attribute do in a link?  
   a) Opens the link in the same tab  
   b) Opens the link in a new window  
   c) Does not open the link  
   d) Forces the link to open in the background**
6. **What is the purpose of the <meta> tag in HTML?  
   a) Defines the metadata for an HTML document  
   b) Creates a navigation menu  
   c) Defines the main content  
   d) None of the above**
7. **How do you group multiple form elements in HTML?  
   a) <fieldset>  
   b) <form-group>  
   c) <elements>  
   d) <section>**
8. **Which HTML tag is used to display a list of options in a drop-down menu?  
   a) <list>  
   b) <option>  
   c) <select>  
   d) <dropdown>**
9. **How do you define the font size in HTML?  
   a) <font size="14px">  
   b) <style="font-size: 14px;">  
   c) <font-size: 14px>  
   d) CSS property font-size**
10. **Which tag is used to create a table row in HTML?  
    a) <row>  
    b) <tr>  
    c) <td>  
    d) <table-row>**
11. **What attribute is used to specify the URL for an anchor (<a>) tag?  
    a) url  
    b) href  
    c) link  
    d) src**
12. **What is the correct HTML tag for inserting a line break?  
    a) <br>  
    b) <lb>  
    c) <break>  
    d) <line>**
13. **How do you insert an unordered list in HTML?  
    a) <list>  
    b) <ol>  
    c) <ul>  
    d) <ulist>**
14. **What attribute is used to specify a target for an anchor link?  
    a) src  
    b) href  
    c) target  
    d) action**
15. **How do you define a text input field in a form?  
    a) <input type="text">  
    b) <text>  
    c) <input type="textfield">  
    d) <textarea>**
16. **What is the purpose of the <footer> tag in HTML?  
    a) Contains the footer content of the webpage  
    b) Defines the header of the webpage  
    c) Used for form fields  
    d) Defines the body section of a webpage**
17. **How can you define a description list in HTML?  
    a) <dl>  
    b) <ol>  
    c) <ul>  
    d) <list>**
18. **Which tag is used to define a form input element that allows users to select a file?  
    a) <input type="file">  
    b) <file-upload>  
    c) <select-file>  
    d) <input type="button">**
19. **How can you link a JavaScript file to an HTML document?  
    a) <script src="script.js">  
    b) <link href="script.js">  
    c) <js src="script.js">  
    d) <javascript src="script.js">**
20. **What is the purpose of the <link> tag in HTML?  
    a) Links an external file to the HTML document  
    b) Defines the main content of the page  
    c) Defines a clickable link  
    d) Defines a table link**

**Hard**

1. **What is the default value of the position property in HTML?  
   a) absolute  
   b) relative  
   c) static  
   d) fixed**
2. **What is the use of the content property in CSS?  
   a) It sets the content of an element  
   b) It specifies additional content for an element before or after its content  
   c) It changes the background color  
   d) It positions elements**
3. **What is the difference between <section> and <div> in HTML?  
   a) <section> is used for grouping content, <div> is used for structuring content  
   b) <section> is a block-level element, <div> is inline  
   c) <section> is used for semantic meaning, <div> is generic  
   d) They are identical**
4. **How do you create a hidden form field in HTML?  
   a) <input type="hidden">  
   b) <input type="invisible">  
   c) <input type="hidden-field">  
   d) <input type="display:none">**
5. **How do you specify an HTML document to use UTF-8 encoding?  
   a) <meta charset="UTF-8">  
   b) <charset="UTF-8">  
   c) <meta encoding="UTF-8">  
   d) <encoding="UTF-8">**
6. **How do you specify the language of an HTML document?  
   a) <html lang="en">  
   b) <html language="en">  
   c) <language="en">  
   d) <document lang="en">**
7. **What does the async attribute do in the <script> tag?  
   a) It specifies that the script should be executed asynchronously  
   b) It loads the script synchronously  
   c) It prevents the script from running  
   d) It forces the script to execute immediately**
8. **What is the correct syntax for adding a comment in HTML?  
   a) /\* comment \*/  
   b) // comment  
   c) <!-- comment -->  
   d) # comment**
9. **How do you create a hyperlink to an anchor on the same page in HTML?  
   a) <a href="#anchor">  
   b) <a href="anchor">  
   c) <a name="anchor">  
   d) <link href="#anchor">**
10. **What is the colspan attribute used for in HTML?  
    a) To merge two or more columns in a table  
    b) To define the number of rows in a table  
    c) To add a background color to a cell  
    d) To align text within a column**
11. **How can you make a table cell span multiple rows in HTML?  
    a) rowspan attribute  
    b) colspan attribute  
    c) span attribute  
    d) columnspan attribute**
12. **What does the <base> tag in HTML do?  
    a) Specifies the base URL for relative links  
    b) Specifies the background color of a webpage  
    c) Defines the main content of the page  
    d) Defines the default font**
13. **How do you specify the action to be taken when a form is submitted in HTML?  
    a) method attribute  
    b) action attribute  
    c) submit attribute  
    d) type attribute**
14. **What does the defer attribute do in the <script> tag?  
    a) Delays the script execution until the HTML parsing is complete  
    b) Makes the script execute immediately  
    c) Prevents the script from running  
    d) Runs the script before HTML parsing**
15. **What does the <meta name="viewport" content="width=device-width, initial-scale=1.0"> tag do?  
    a) Defines the initial zoom level and width of the page  
    b) Sets the character encoding for the page  
    c) Specifies the background color  
    d) Defines the content-type of the page**
16. **How do you create a tooltip in HTML?  
    a) <title>  
    b) hover event  
    c) data-tooltip attribute  
    d) alt attribute**
17. **What is the aria-label attribute used for?  
    a) Defines an accessible label for screen readers  
    b) Defines the color of the text  
    c) Specifies an element’s class  
    d) Sets the size of an element**
18. **How can you specify the size of a canvas in HTML?  
    a) width and height attributes  
    b) size attribute  
    c) dimension attributes  
    d) canvas-size attribute**
19. **How can you create a responsive table in HTML?  
    a) By using <table-responsive> class  
    b) By using @media queries  
    c) By using max-width on table elements  
    d) All of the above**
20. **What does the rel="noopener noreferrer" attribute do in a link?  
    a) It protects against potential malicious activities in links  
    b) It disables the link from opening in a new tab  
    c) It specifies that the link should open in the same tab  
    d) It prevents the link from being followed**

**React Quiz**

**Easy**

1. **What is React?  
   a) A JavaScript framework  
   b) A JavaScript library  
   c) A programming language  
   d) A database management system**
2. **What is JSX?  
   a) A templating language used in React  
   b) A new version of JavaScript  
   c) A type of CSS  
   d) A data management system**
3. **Which of the following is used to create a React component?  
   a) createReactComponent()  
   b) <Component></Component>  
   c) class Component extends React.Component  
   d) React.Component.create()**
4. **What function is used to render a React component to the DOM?  
   a) React.render()  
   b) renderComponent()  
   c) ReactDOM.render()  
   d) component.render()**
5. **In React, what is the purpose of the render() method?  
   a) To define how the component’s output should look  
   b) To initialize component states  
   c) To fetch data  
   d) To define component styles**
6. **What is the correct syntax to define a functional component in React?  
   a) function MyComponent() { return <div></div>; }  
   b) React.Component MyComponent() { return <div></div>; }  
   c) const MyComponent = () => <div></div>;  
   d) ReactFunction MyComponent() { return <div></div>; }**
7. **Which hook is used to handle side effects in React?  
   a) useEffect()  
   b) useState()  
   c) useContext()  
   d) useRef()**
8. **How do you pass props to a React component?  
   a) Using this.props  
   b) Using this.state  
   c) Using <Component name="value" />  
   d) Using props() function**
9. **How do you update the state of a React component?  
   a) this.state = newState  
   b) this.updateState(newState)  
   c) this.setState(newState)  
   d) setState(newState)**
10. **What does the key prop do in React?  
    a) It uniquely identifies elements in a list for efficient rendering  
    b) It defines the order of elements  
    c) It links props to the state  
    d) It specifies the default value for props**
11. **What is a React Hook?  
    a) A function that allows you to hook into React state and lifecycle features  
    b) A component that allows data binding  
    c) A method to define styles in a component  
    d) A feature for asynchronous operations**
12. **How do you handle user input in React?  
    a) Using useState() to manage input value  
    b) Using handleEvent()  
    c) Using inputValue state  
    d) Using this.inputField reference**
13. **Which function is used to handle events in React?  
    a) handleEvent()  
    b) this.onClick()  
    c) onEvent()  
    d) onClick()**
14. **What is a controlled component in React?  
    a) A component where the state is managed by the parent  
    b) A component with no props  
    c) A component that does not render anything  
    d) A component that uses the useEffect hook**
15. **How do you create a class component in React?  
    a) class Component extends React.Component {}  
    b) React.createClass(Component) {}  
    c) class ReactComponent {}  
    d) React.createComponent()**
16. **Which function do you use to manage state in a functional React component?  
    a) useEffect()  
    b) useState()  
    c) state()  
    d) this.setState()**
17. **How do you conditionally render elements in React?  
    a) Using if statements  
    b) Using the && operator or ternary operator  
    c) Using for loops  
    d) Using return only**
18. **Which of the following hooks allows you to refer to elements or values in a React component?  
    a) useRef()  
    b) useContext()  
    c) useEffect()  
    d) useState()**
19. **How do you create an event handler in React?  
    a) By defining a method within the component class  
    b) By passing a function as a prop  
    c) By adding event listeners manually in the DOM  
    d) By importing event handlers from a library**
20. **What is the main benefit of React’s Virtual DOM?  
    a) Faster updates by minimizing direct manipulation of the real DOM  
    b) Allows for server-side rendering  
    c) Easier styling of components  
    d) Automatic data binding**

**Medium**

1. **What is the difference between useState() and useEffect() hooks?  
   a) useState() manages state, useEffect() manages side effects  
   b) useState() manages side effects, useEffect() manages state  
   c) useState() is for functional components only, useEffect() is for class components only  
   d) Both hooks manage side effects**
2. **How can you pass state between components in React?  
   a) Using props  
   b) Using the context API  
   c) Using a global store like Redux  
   d) All of the above**
3. **What does the useEffect() hook do in React?  
   a) It runs a side effect after every render  
   b) It runs only once after the initial render  
   c) It manages state updates  
   d) It runs before the component is rendered**
4. **Which of the following is a correct way to call a function inside JSX?  
   a) {functionName()}  
   b) functionName[]  
   c) <functionName>  
   d) functionName{}**
5. **How do you update an object inside the state using setState() in React?  
   a) this.setState({ objectName: newObject })  
   b) this.state = newObject  
   c) this.updateState({ objectName: newObject })  
   d) this.setState(newObject)**
6. **What is the purpose of the React.Fragment component?  
   a) To group multiple elements without adding extra nodes to the DOM  
   b) To manage the state of components  
   c) To define reusable components  
   d) To manage CSS styles for components**
7. **How do you handle form submissions in React?  
   a) By using onSubmit() event handler  
   b) By using handleSubmit() function  
   c) By using preventDefault() in event handler  
   d) All of the above**
8. **What is React Router used for?  
   a) Managing component rendering based on the URL  
   b) Styling components  
   c) Handling data fetching  
   d) Managing state of components**
9. **What does the useContext() hook do in React?  
   a) Provides access to the closest value of a context  
   b) Allows a component to subscribe to the state  
   c) Manages component events  
   d) Manages data fetching in components**
10. **How do you handle multiple events in React?  
    a) By using a single handler for each event type  
    b) By using event delegation  
    c) By binding multiple functions to the component  
    d) By using the onClick property multiple times**
11. **What is the purpose of React.memo() in React?  
    a) To memoize a component and prevent unnecessary re-renders  
    b) To manage local state  
    c) To allow components to fetch data  
    d) To manage animations**
12. **What does the useReducer() hook do in React?  
    a) Manages complex state logic in a functional component  
    b) Updates the component’s state asynchronously  
    c) Handles component lifecycle methods  
    d) Maps the state to props**
13. **What is the difference between controlled and uncontrolled components in React?  
    a) Controlled components manage their state with React, uncontrolled components manage their state outside React  
    b) Controlled components are always faster  
    c) Uncontrolled components don’t accept any input  
    d) There is no difference**
14. **How can you prevent a form from refreshing the page in React?  
    a) By using event.preventDefault()  
    b) By using this.preventRefresh()  
    c) By using the no-refresh attribute  
    d) By using return false;**
15. **What is a higher-order component in React?  
    a) A component that takes another component as an argument and returns a new component  
    b) A component that renders only once  
    c) A component that is rendered within another component  
    d) A component that manages state**
16. **How do you pass functions as props in React?  
    a) By declaring the function in the parent and passing it to the child component  
    b) By calling the function inside the child component  
    c) By using this keyword  
    d) By using React.Component methods**
17. **How do you dynamically add classes to a React element?  
    a) By using classNames library  
    b) By passing a dynamic string in className  
    c) By using style attribute  
    d) Both a and b**
18. **How do you fetch data from an API in React?  
    a) Using the fetch() API inside useEffect()  
    b) Using the axios library inside componentDidMount()  
    c) Using the fetch() API inside componentDidMount()  
    d) All of the above**
19. **How do you handle conditional rendering in React?  
    a) Using if/else statements  
    b) Using ternary operators or &&  
    c) Using useState() hook  
    d) Using useEffect() hook**
20. **What is the default state of useState()?  
    a) null  
    b) false  
    c) undefined  
    d) setState**

**Hard**

1. **What is the difference between useEffect() with an empty dependency array and one with specific dependencies?  
   a) An empty array runs once, dependencies run on specific changes  
   b) They both run on every render  
   c) An empty array runs on every render, dependencies run only on the first render  
   d) They are identical**
2. **How do you prevent a component from re-rendering in React?  
   a) By using shouldComponentUpdate() in class components  
   b) By using React.memo() in functional components  
   c) By using PureComponent in class components  
   d) All of the above**
3. **What does the useLayoutEffect() hook do in React?  
   a) It runs synchronously after all DOM mutations  
   b) It runs asynchronously before rendering  
   c) It fetches data before the component renders  
   d) It does not exist in React**
4. **How do you pass data between sibling components in React?  
   a) Using context  
   b) Using props via a common parent component  
   c) Using Redux or other state management libraries  
   d) All of the above**
5. **What is the purpose of getDerivedStateFromProps() in React?  
   a) To update state based on props before every render  
   b) To initialize state from props  
   c) To avoid using componentDidUpdate()  
   d) To fetch new props for a component**
6. **How does React handle event binding in class components?  
   a) By automatically binding event handlers to this  
   b) By using bind() method  
   c) By using handleEvent() method  
   d) By binding events inside componentDidMount()**
7. **How do you define a default prop in React?  
   a) By using defaultProps  
   b) By using this.props.default  
   c) By using props={default: value}  
   d) By using useEffect()**
8. **What is contextType used for in React?  
   a) To consume context inside class components  
   b) To provide context inside functional components  
   c) To prevent context updates  
   d) To manage global state**
9. **What are "render props" in React?  
   a) A pattern for sharing code between components using a function that returns a component  
   b) A pattern for managing component lifecycle  
   c) A way of styling components  
   d) A technique for passing state between components**
10. **How do you implement code splitting in React?  
    a) By using React.lazy() and Suspense  
    b) By importing all components dynamically  
    c) By using useEffect() with dynamic imports  
    d) By using webpack configuration**
11. **How do you avoid prop drilling in React?  
    a) By using context API  
    b) By using Redux  
    c) By passing props through multiple components  
    d) Both a and b**
12. **How do you manage global state in React without Redux?  
    a) By using React's context API  
    b) By using useState() hook globally  
    c) By passing state through props  
    d) By using componentDidUpdate()**
13. **How does React handle async rendering?  
    a) By using React.lazy()  
    b) By using Suspense for async components  
    c) By suspending rendering until data is fetched  
    d) Both b and c**
14. **How do you create custom hooks in React?  
    a) By using useState() and useEffect() together in a function  
    b) By using this.customHook() method  
    c) By creating new React components  
    d) By extending React.Component**
15. **What is the difference between componentWillUnmount() and useEffect() cleanup in React?  
    a) componentWillUnmount() is used in class components, while cleanup is used in useEffect()  
    b) componentWillUnmount() is synchronous, while cleanup is asynchronous  
    c) They are identical  
    d) useEffect() cleanup is used only in functional components**
16. **What does React.StrictMode do in React?  
    a) It highlights potential problems in the application without affecting the production build  
    b) It prevents code from being minified  
    c) It optimizes the rendering process  
    d) It disables lifecycle methods**
17. **How do you optimize performance in React?  
    a) By using React.memo() and PureComponent  
    b) By lazy loading components with React.lazy()  
    c) By minimizing re-renders and using memoization techniques  
    d) All of the above**
18. **What is the purpose of useImperativeHandle() in React?  
    a) To expose a component’s instance methods to its parent  
    b) To pass props to a component  
    c) To create a custom hook  
    d) To avoid re-rendering**
19. **What is the difference between a "function component" and "class component" in React?  
    a) Function components are simpler and don’t require lifecycle methods  
    b) Class components use hooks, while function components don’t  
    c) Class components can’t have state  
    d) Function components are less performant than class components**
20. **How do you handle errors in React using Error Boundaries?  
    a) By creating a class component that implements componentDidCatch()  
    b) By using try-catch blocks around JSX  
    c) By using useErrorBoundary() hook  
    d) By wrapping the component in React.Fragment**

**Testing (Jest, Mocha, Unit Testing, Integration Testing) Quiz**

**Easy**

1. **What is Jest primarily used for?  
   a) Styling web pages  
   b) Testing JavaScript code  
   c) Managing databases  
   d) Building websites**
2. **Which command is used to run tests in Jest?  
   a) npm run test  
   b) npm test  
   c) jest run  
   d) test run**
3. **What does Mocha help developers with?  
   a) Database management  
   b) Writing JavaScript applications  
   c) Running unit and integration tests  
   d) Styling web applications**
4. **Which of the following is the correct way to describe a test case in Jest?  
   a) test('Description', () => {})  
   b) describe('Description', () => {})  
   c) assert('Description', () => {})  
   d) check('Description', () => {})**
5. **What is a unit test?  
   a) Testing a database  
   b) Testing the entire application  
   c) Testing a single function or unit of code  
   d) Testing user interfaces**
6. **Which function is used to make assertions in Jest tests?  
   a) assert()  
   b) expect()  
   c) verify()  
   d) check()**
7. **In Mocha, which function is used to define a test?  
   a) test()  
   b) expect()  
   c) it()  
   d) describe()**
8. **How do you install Jest in a project?  
   a) npm install jest  
   b) npm add jest  
   c) npm install mocha  
   d) npm install test**
9. **What is integration testing?  
   a) Testing individual components in isolation  
   b) Testing the integration of multiple components  
   c) Testing the style of a webpage  
   d) Testing database performance**
10. **Which of the following is a common testing method for API endpoints?  
    a) Unit testing  
    b) Smoke testing  
    c) Integration testing  
    d) End-to-end testing**
11. **In Jest, what is the function beforeEach() used for?  
    a) Setting up tests before each test runs  
    b) Cleaning up after each test runs  
    c) Skipping the current test  
    d) Running the test multiple times**
12. **In Mocha, what is the function before() used for?  
    a) It runs once before all tests  
    b) It runs after each test  
    c) It runs before each test  
    d) It runs after all tests**
13. **What is the function describe() used for in Jest or Mocha?  
    a) To set up the testing environment  
    b) To group related tests together  
    c) To make assertions  
    d) To clean up after tests**
14. **How do you test asynchronous code in Jest?  
    a) Using done()  
    b) Using await and async  
    c) Using setTimeout()  
    d) Using setInterval()**
15. **Which assertion is correct in Jest to check if a value equals another value?  
    a) expect(value).toEqual(value)  
    b) expect(value).toBe(value)  
    c) assert(value).equals(value)  
    d) expect(value).equals(value)**
16. **In Mocha, how do you handle asynchronous tests?  
    a) By using the async/await pattern  
    b) By passing a done callback to the test function  
    c) By returning a Promise  
    d) All of the above**
17. **Which of the following is an example of a "mock" in Jest?  
    a) jest.fn()  
    b) jest.spyOn()  
    c) jest.mock()  
    d) All of the above**
18. **What is the purpose of afterEach() in Jest?  
    a) It runs once after each test case  
    b) It runs before each test case  
    c) It resets test data  
    d) It initializes variables before tests**
19. **What is the correct syntax to skip a test in Jest?  
    a) test.skip()  
    b) it.skip()  
    c) describe.skip()  
    d) test.skip('description', () => {})**
20. **How can you test an error thrown by a function in Jest?  
    a) By using expect(function).toThrow()  
    b) By using assert.error()  
    c) By using it.raises()  
    d) By using assert.throws()**

**Medium**

1. **What is the difference between unit tests and integration tests?  
   a) Unit tests check the entire system, while integration tests check individual units  
   b) Unit tests check individual units of code, while integration tests check interactions between units  
   c) Unit tests use mocks, while integration tests use real systems  
   d) There is no difference between the two**
2. **How do you test a function that involves promises in Jest?  
   a) expect(asyncFunction).toBeResolved()  
   b) expect(asyncFunction).toHaveReturned()  
   c) expect(asyncFunction).resolves.toBe(value)  
   d) expect(asyncFunction).toBePending()**
3. **What is a "spy" in Jest?  
   a) A function that executes asynchronously  
   b) A function used to replace an existing function with a mocked version  
   c) A function that checks for errors  
   d) A function used for data logging**
4. **What is the done callback used for in Mocha?  
   a) To finish the test asynchronously  
   b) To stop a test from running  
   c) To declare the test as passed  
   d) To simulate network requests**
5. **What is the difference between expect().toBe() and expect().toEqual() in Jest?  
   a) toBe() checks strict equality, while toEqual() checks deep equality  
   b) toEqual() checks strict equality, while toBe() checks deep equality  
   c) There is no difference between the two  
   d) toBe() is used for asynchronous tests**
6. **How do you mock a module in Jest?  
   a) jest.mock('module-name')  
   b) jest.mockModule('module-name')  
   c) mock('module-name')  
   d) jest.importMock('module-name')**
7. **What is the purpose of the assert module in Node.js testing?  
   a) To execute asynchronous tests  
   b) To write a report for the tests  
   c) To perform assertions for testing conditions  
   d) To run the tests**
8. **In Mocha, how do you run a specific test or test suite?  
   a) mocha path-to-test  
   b) mocha test() {}  
   c) mocha -run test  
   d) mocha --only path-to-test**
9. **How can you make sure that a mock function is called a certain number of times in Jest?  
   a) expect(mockFunction).toHaveBeenCalledTimes(3)  
   b) expect(mockFunction).toHaveBeenCalledOnce()  
   c) expect(mockFunction).toHaveTimes(3)  
   d) expect(mockFunction).toHaveCallCount(3)**
10. **What is the syntax for testing HTTP requests using Jest and Supertest?  
    a) request(app).get('/path').expect(200)  
    b) request('/path').get(app).expect(200)  
    c) app.get('/path').request().expect(200)  
    d) expect(request(app).get('/path')).toEqual(200)**
11. **How do you handle exceptions in Jest tests?  
    a) expect(function).toThrowError()  
    b) expect(function).toThrow()  
    c) expect(function).toBeThrown()  
    d) expect(function).toThrowException()**
12. **In Jest, how do you test the value returned by a Promise?  
    a) expect(promise).resolves.toEqual(value)  
    b) expect(promise).toEqual(value)  
    c) expect(promise).resolves.equal(value)  
    d) expect(promise).toResolve(value)**
13. **Which of the following testing types uses the actual components of a system working together?  
    a) Unit testing  
    b) Integration testing  
    c) Smoke testing  
    d) Functional testing**
14. **How do you run tests on multiple conditions in Jest?  
    a) Using test.each()  
    b) Using describe.each()  
    c) Using test.parallel()  
    d) Using runTests()**
15. **In Mocha, which of these methods is used for defining a test suite?  
    a) describe()  
    b) suite()  
    c) it()  
    d) testSuite()**
16. **What is the correct syntax to test async code with Jest using async/await?  
    a) expect(await asyncFunction()).resolves.toBe(value)  
    b) expect(asyncFunction()).toBe(value)  
    c) expect(asyncFunction).toResolve(value)  
    d) expect(await asyncFunction).toEqual(value)**
17. **What is the main difference between integration tests and functional tests?  
    a) Integration tests test entire systems, while functional tests test individual functions  
    b) Integration tests test interaction between system components, while functional tests test a function's behavior  
    c) Integration tests run on the server side, while functional tests run on the client side  
    d) There is no difference**
18. **How do you perform assertions on a JSON response in Jest?  
    a) expect(response).toHaveJson(value)  
    b) expect(response.json()).toEqual(value)  
    c) expect(response).toEqualJson(value)  
    d) expect(response).json(value)**
19. **How do you test that a function is not called in Jest?  
    a) expect(mockFunction).toHaveBeenCalled()  
    b) expect(mockFunction).not.toHaveBeenCalled()  
    c) expect(mockFunction).toNotHaveBeenCalled()  
    d) expect(mockFunction).never.toHaveBeenCalled()**
20. **What does the after() function do in Mocha?  
    a) It runs after each individual test  
    b) It runs before each test  
    c) It runs once after all tests in a suite have completed  
    d) It runs after each test suite completes**

**Hard**

1. **How do you test multiple assertions on the same mock function in Jest?  
   a) expect(mockFunction).toHaveBeenCalledTimes(2).toHaveBeenCalledWith(arg)  
   b) expect(mockFunction).toCall(2).toCallWith(arg)  
   c) expect(mockFunction).toHaveBeenCalledAtLeast(2).toHaveBeenCalledWith(arg)  
   d) expect(mockFunction).toHaveTimes(2)**
2. **How can you handle time-dependent tests in Jest?  
   a) Using jest.runAllTimers()  
   b) Using jest.advanceTimersByTime()  
   c) Using jest.useFakeTimers()  
   d) All of the above**
3. **What is the purpose of jest.fn()?  
   a) To create a mock function  
   b) To spy on function calls  
   c) To replace a function with a mock version  
   d) All of the above**
4. **How do you test the HTTP request and response in Mocha using Supertest?  
   a) request(app).get('/path').expect(200).then(response => {...})  
   b) app.request().get('/path').expect(200).send(response => {...})  
   c) request(app).post('/path').expect(200)  
   d) request().expect(response).toHaveStatus(200)**
5. **In Jest, what does mockImplementationOnce() do?  
   a) Mocks a function only once with a specified implementation  
   b) Mocks a function with multiple implementations  
   c) Mocks a function forever  
   d) Replaces a function's implementation completely**
6. **How can you simulate an HTTP request failure in Jest tests?  
   a) By using jest.mock() to simulate the failure  
   b) By mocking an API with mockRejectedValue()  
   c) By using throwError()  
   d) By setting up a test that triggers a network error**
7. **What is the purpose of using jest.clearAllMocks()?  
   a) To reset all mocks between tests  
   b) To clear test results  
   c) To stop mocking specific functions  
   d) To remove all test files**
8. **How do you define a custom matcher in Jest?  
   a) expect.extend({ toBeMock: () => {} })  
   b) expect.custom({ toBeMock: () => {} })  
   c) jest.defineCustomMatcher({ toBeMock: () => {} })  
   d) expect.addMatcher({ toBeMock: () => {} })**
9. **How do you handle a function that takes a callback in Mocha?  
   a) By using done callback to signal completion  
   b) By using async/await pattern  
   c) By returning a Promise  
   d) All of the above**
10. **What is the default test behavior when using jest --coverage?  
    a) It tests only code covered by tests  
    b) It tests the entire codebase regardless of test coverage  
    c) It skips over untested code  
    d) It adds extra code for testing**
11. **How do you run tests asynchronously in Mocha?  
    a) By using Promises  
    b) By passing a done callback  
    c) By using async/await  
    d) All of the above**
12. **What is the difference between beforeEach() and before() in Mocha?  
    a) beforeEach() runs before each test case, while before() runs once before the test suite  
    b) Both run before each test case  
    c) beforeEach() runs after each test case, while before() runs once before the test suite  
    d) There is no difference**
13. **How can you skip a specific test in Jest?  
    a) test.skip('Description', () => {})  
    b) test.ignore('Description', () => {})  
    c) describe.skip('Description')  
    d) skipTest('Description', () => {})**
14. **What is the function of jest.resetAllMocks()?  
    a) To reset the implementation of all mocks  
    b) To reset the state of all mocks to their initial values  
    c) To reset all tests  
    d) To mock all functions in the test**
15. **In Mocha, which function is used for cleaning up after tests?  
    a) afterEach()  
    b) beforeEach()  
    c) after()  
    d) before()**
16. **How can you mock a specific module in Jest?  
    a) jest.mock('./moduleName')  
    b) mock('./moduleName')  
    c) jest.spyOn('./moduleName')  
    d) jest.fake('./moduleName')**
17. **What is the purpose of jest.clearAllTimers()?  
    a) To reset all mock timers  
    b) To clear all test case results  
    c) To stop any ongoing async operations  
    d) To cancel test execution**
18. **How do you handle performance testing with Jest?  
    a) Jest doesn't handle performance testing  
    b) By using jest.runPerformanceTests()  
    c) By using jest.measure()  
    d) By setting up custom performance tests**
19. **How do you ensure a mock function is called with specific arguments in Jest?  
    a) expect(mockFunction).toHaveBeenCalledWith(arg1, arg2)  
    b) expect(mockFunction).toCallWith(arg1, arg2)  
    c) expect(mockFunction).toEqualWith(arg1, arg2)  
    d) expect(mockFunction).toHaveArguments(arg1, arg2)**
20. **How do you assert that a mock function was called after a specific time in Jest?  
    a) expect(mockFunction).toHaveBeenCalledAfter(time)  
    b) expect(mockFunction).toHaveBeenCalledWithin(time)  
    c) expect(mockFunction).toHaveBeenCalledTimesAtLeast(time)  
    d) expect(mockFunction).toHaveBeenCalledTimesAfter(time)**

**Express, Node.js (Frontend/Backend) Quiz**

**Easy**

1. **What is Node.js primarily used for?  
   a) Building databases  
   b) Building interactive websites  
   c) Running JavaScript on the server  
   d) Running JavaScript in the browser**
2. **What is Express.js?  
   a) A database management system  
   b) A JavaScript runtime environment  
   c) A web application framework for Node.js  
   d) A CSS framework**
3. **How do you install Express in a Node.js project?  
   a) npm install express  
   b) npm add express  
   c) npm express install  
   d) node install express**
4. **What is the default HTTP port that an Express app listens to?  
   a) 8080  
   b) 3000  
   c) 5000  
   d) 8088**
5. **In Express, which method is used to create a simple HTTP server?  
   a) app.listen()  
   b) app.start()  
   c) server.create()  
   d) app.run()**
6. **Which of the following is a valid Express route?  
   a) app.get('/home', function(req, res) {...})  
   b) app.home('/home', function(req, res) {...})  
   c) app.route('/home', function(req, res) {...})  
   d) app.create('/home', function(req, res) {...})**
7. **What is req in Express?  
   a) The request object  
   b) The response object  
   c) The router object  
   d) The middleware object**
8. **How do you handle HTTP POST requests in Express?  
   a) app.get()  
   b) app.post()  
   c) app.put()  
   d) app.delete()**
9. **How do you serve static files in Express?  
   a) app.use(express.static('public'))  
   b) app.static('public')  
   c) app.serve('public')  
   d) app.set('static', 'public')**
10. **What is middleware in Express?  
    a) A function that handles HTTP requests  
    b) A function that handles data storage  
    c) A function that manages routes  
    d) A function that runs before sending a response**
11. **In Express, which function is used to send a JSON response?  
    a) res.json()  
    b) res.sendJSON()  
    c) res.response()  
    d) res.data()**
12. **What is Node.js?  
    a) A database management system  
    b) A web server  
    c) A JavaScript runtime built on Chrome’s V8 engine  
    d) A task automation tool**
13. **How do you require a module in Node.js?  
    a) require('module')  
    b) import module from 'module'  
    c) include module  
    d) load module()**
14. **What is the role of package.json in Node.js?  
    a) To store database credentials  
    b) To configure the Node.js runtime environment  
    c) To define metadata about a project, including dependencies  
    d) To store environment variables**
15. **How do you install dependencies in a Node.js project?  
    a) npm add  
    b) npm install  
    c) node install  
    d) npm fetch**
16. **Which Node.js module is used to work with the file system?  
    a) fs  
    b) file  
    c) directory  
    d) storage**
17. **What is a callback in Node.js?  
    a) A function that is executed asynchronously after a task is completed  
    b) A function that handles database queries  
    c) A function that starts the application  
    d) A function that triggers events**
18. **How do you create an HTTP server in Node.js?  
    a) http.createServer()  
    b) server.create()  
    c) http.server()  
    d) server.listen()**
19. **Which of the following is used to parse incoming request bodies in Express?  
    a) express.bodyParser()  
    b) express.json()  
    c) express.urlencoded()  
    d) express.requestParser()**
20. **How do you define a route to handle a GET request in Express?  
    a) app.get('/path', function(req, res) {...})  
    b) app.post('/path', function(req, res) {...})  
    c) app.route('/path', function(req, res) {...})  
    d) app.use('/path', function(req, res) {...})**

**Medium**

1. **What is the purpose of next() in Express middleware?  
   a) To continue to the next middleware or route handler  
   b) To stop the current request  
   c) To send a response to the client  
   d) To restart the server**
2. **What does res.status(200) do in an Express route handler?  
   a) Sets the HTTP status code to 200 (OK)  
   b) Sets the response content type  
   c) Sets the response body  
   d) Sends a JSON response**
3. **What is the difference between app.get() and app.post() in Express?  
   a) app.get() is used for GET requests, app.post() is used for POST requests  
   b) app.get() is used for server-side rendering, app.post() is for API calls  
   c) app.get() handles data submission, app.post() handles data retrieval  
   d) There is no difference**
4. **How do you define a route parameter in Express?  
   a) app.get('/user/:id', function(req, res) {...})  
   b) app.get('/user{id}', function(req, res) {...})  
   c) app.get('/user(/id)', function(req, res) {...})  
   d) app.get('/user/\*id', function(req, res) {...})**
5. **In Node.js, what is the require() function used for?  
   a) To define a new module  
   b) To import a built-in or external module  
   c) To invoke an external API  
   d) To export a module**
6. **Which Node.js function is used to read a file asynchronously?  
   a) fs.readFile()  
   b) fs.read()  
   c) fs.openFile()  
   d) fs.getFile()**
7. **How do you handle errors in Express middleware?  
   a) By passing the error to the next middleware with next(error)  
   b) By throwing an error and stopping the server  
   c) By logging the error only  
   d) By returning an error message in the response**
8. **How can you handle different HTTP methods on the same route in Express?  
   a) Use app.all()  
   b) Use app.method()  
   c) Use app.route()  
   d) Use app.use()**
9. **What is the role of the express.Router() in Express?  
   a) To define routes and handle requests  
   b) To create a database connection  
   c) To manage HTTP status codes  
   d) To define global middleware**
10. **Which method is used to send data to a client using JSON in Express?  
    a) res.sendJSON()  
    b) res.send()  
    c) res.json()  
    d) res.output()**
11. **What is the purpose of the app.use() method in Express?  
    a) To define a middleware function for all routes  
    b) To define a custom route handler  
    c) To send a response to the client  
    d) To parse incoming request bodies**
12. **How do you create a REST API in Express?  
    a) By defining routes and handling HTTP requests for specific resources  
    b) By creating a database schema  
    c) By generating static HTML pages  
    d) By setting up authentication middleware**
13. **What is a promise in Node.js?  
    a) A callback function  
    b) An object that represents the eventual completion or failure of an asynchronous operation  
    c) A type of data structure  
    d) A process for handling errors**
14. **How do you handle authentication in Express?  
    a) Using JSON Web Tokens (JWT)  
    b) Using middleware functions  
    c) Using sessions  
    d) All of the above**
15. **What is express.json() used for in Express?  
    a) To parse JSON request bodies  
    b) To send JSON data to the client  
    c) To validate JSON schema  
    d) To log incoming JSON data**
16. **How do you handle query parameters in Express?  
    a) req.query  
    b) req.body  
    c) req.params  
    d) req.headers**
17. **What is CORS, and why is it important in Node.js?  
    a) Cross-Origin Request Sharing, allows servers to specify who can access their resources  
    b) Cross-Origin Request Service, helps in authentication  
    c) Content Oriented Response System, helps in routing  
    d) Centralized Origin Routing System, used for managing routes**
18. **What does app.set('view engine', 'ejs') do in an Express app?  
    a) Sets the template engine to EJS  
    b) Sets the routing behavior  
    c) Defines the database type  
    d) Configures middleware**
19. **How do you handle URL-encoded data in Express?  
    a) express.urlencoded()  
    b) express.bodyParser()  
    c) express.json()  
    d) express.dataParser()**
20. **How do you set up a route that accepts both GET and POST methods in Express?  
    a) app.all('/path', ...)  
    b) app.route('/path').get().post()  
    c) app.any('/path', ...)  
    d) app.use('/path', ...)**

**Hard**

1. **What is the purpose of app.param() in Express?  
   a) To define parameters for specific routes  
   b) To handle route parameters in a middleware  
   c) To extract query parameters  
   d) To modify the request body**
2. **What is the difference between req.body and req.query in Express?  
   a) req.body contains form data, req.query contains query parameters  
   b) req.body contains JSON data, req.query contains cookies  
   c) req.body is used for URL parameters, req.query is used for POST data  
   d) There is no difference**
3. **How do you enable cookie parsing in Express?  
   a) app.use(cookieParser())  
   b) app.use(cookieParse())  
   c) app.use(parseCookies())  
   d) app.use(parseCookieData())**
4. **How do you integrate MongoDB with an Express application?  
   a) By using the mongoose library  
   b) By using the mongodb module  
   c) By writing raw SQL queries  
   d) By setting up a NoSQL database server**
5. **How can you prevent route hijacking in Express?  
   a) By using proper authorization and authentication middleware  
   b) By restricting access to specific IP addresses  
   c) By checking request methods  
   d) By encoding all route parameters**
6. **What is the role of cluster module in Node.js?  
   a) To create child processes for better CPU utilization  
   b) To manage database connections  
   c) To handle web server routing  
   d) To parse incoming requests**
7. **What does the app.listen() method do in Express?  
   a) Starts the Express server and listens on a specific port  
   b) Defines the route to handle HTTP requests  
   c) Sets up middleware for all requests  
   d) Handles static file serving**
8. **How do you handle rate limiting in Express?  
   a) By using third-party middleware like express-rate-limit  
   b) By manually setting timeouts  
   c) By using HTTP status codes  
   d) By logging failed attempts**
9. **How do you send an error response with status code 500 in Express?  
   a) res.status(500).send('Internal Server Error')  
   b) res.sendError(500)  
   c) res.error(500)  
   d) res.status(500).json({ error: 'Internal Error' })**
10. **How do you protect your Express app from SQL injection attacks?  
    a) By using parameterized queries with libraries like mongoose  
    b) By encoding all inputs  
    c) By using GET requests only  
    d) By sanitizing all outputs**
11. **How do you handle file uploads in Express?  
    a) By using middleware like multer  
    b) By writing custom file handling functions  
    c) By using fs to handle uploads  
    d) By sending files as base64 strings**
12. **What is the purpose of async and await in Node.js/Express?  
    a) To handle asynchronous operations in a synchronous manner  
    b) To make synchronous code asynchronous  
    c) To handle HTTP requests asynchronously  
    d) To parse query parameters**
13. **How do you handle multiple sub-routes in an Express app?  
    a) By using express.Router()  
    b) By creating separate Express instances  
    c) By defining multiple app.use() calls  
    d) By chaining route handlers**
14. **How do you prevent security vulnerabilities like Cross-Site Scripting (XSS) in Express?  
    a) By sanitizing user inputs  
    b) By using HTTPS  
    c) By limiting allowed request methods  
    d) By logging all input**
15. **What does mongoose.connect() do in an Express app?  
    a) Establishes a connection to a MongoDB database  
    b) Creates a new database  
    c) Configures the Express server  
    d) Creates a new model schema**
16. **How do you configure error handling in Express?  
    a) By creating an error-handling middleware with next(err)  
    b) By using a catch block  
    c) By defining a global try/catch function  
    d) By writing a custom error() function**
17. **How do you integrate JWT authentication in an Express application?  
    a) By using the jsonwebtoken library to sign and verify tokens  
    b) By using cookies for storing session information  
    c) By defining user routes for login and registration  
    d) By using the express-session middleware**
18. **How do you implement WebSockets in a Node.js application with Express?  
    a) By using the ws library  
    b) By using socket.io  
    c) By using HTTP Upgrade header  
    d) All of the above**
19. **How do you implement caching in Express for API responses?  
    a) By using express-redis-cache or similar middleware  
    b) By manually caching in memory  
    c) By using the cache-control header  
    d) All of the above**
20. **What is the purpose of app.set('env', 'production') in Express?  
    a) To set the application environment to production  
    b) To configure production-level error handling  
    c) To enable certain optimizations for production  
    d) All of the above**

**JavaScript Methods Quiz**

**Easy**

1. **What does the toUpperCase() method do in JavaScript?  
   a) Converts a string to lowercase  
   b) Converts a string to uppercase  
   c) Reverses the string  
   d) Trims the string**
2. **Which method is used to add one or more elements to the end of an array?  
   a) push()  
   b) pop()  
   c) shift()  
   d) unshift()**
3. **What does the indexOf() method do in JavaScript?  
   a) Searches for an element and returns its index  
   b) Finds the last element of an array  
   c) Adds a new element to the array  
   d) Removes an element from the array**
4. **Which method is used to remove the first element of an array?  
   a) shift()  
   b) unshift()  
   c) push()  
   d) pop()**
5. **What does the split() method do in JavaScript?  
   a) Joins an array into a string  
   b) Splits a string into an array  
   c) Converts an array into an object  
   d) Converts a string to uppercase**
6. **Which of the following methods is used to sort an array alphabetically?  
   a) sort()  
   b) reverse()  
   c) join()  
   d) concat()**
7. **How do you create a new object in JavaScript?  
   a) let obj = new Object()  
   b) let obj = {}  
   c) let obj = Object.create()  
   d) All of the above**
8. **Which method is used to find the length of a string in JavaScript?  
   a) string.length()  
   b) string.size()  
   c) string.length  
   d) string.count()**
9. **How can you access the last element of an array in JavaScript?  
   a) arr[0]  
   b) arr[arr.length - 1]  
   c) arr[1]  
   d) arr.end()**
10. **What does the concat() method do?  
    a) Combines two or more strings  
    b) Merges two arrays  
    c) Adds an element to the beginning of an array  
    d) All of the above**
11. **Which method can be used to determine if a string contains a certain substring?  
    a) contains()  
    b) has()  
    c) includes()  
    d) find()**
12. **What does the join() method do in an array?  
    a) Joins two arrays together  
    b) Joins all array elements into a string  
    c) Removes the last array element  
    d) Removes the first array element**
13. **Which method is used to remove the last element of an array?  
    a) shift()  
    b) pop()  
    c) unshift()  
    d) push()**
14. **How do you replace a character in a string in JavaScript?  
    a) replace()  
    b) substring()  
    c) slice()  
    d) split()**
15. **What does the slice() method do on an array?  
    a) Adds an element to the array  
    b) Returns a shallow copy of a portion of the array  
    c) Merges two arrays  
    d) Sorts the array in ascending order**
16. **Which method is used to test whether a string matches a regular expression?  
    a) test()  
    b) match()  
    c) search()  
    d) find()**
17. **Which method is used to remove all white space from both ends of a string?  
    a) trim()  
    b) replace()  
    c) strip()  
    d) clean()**
18. **How can you check if an element is an array in JavaScript?  
    a) Array.isArray()  
    b) Array.check()  
    c) isArray()  
    d) array.is()**
19. **Which method is used to round a number to the nearest integer?  
    a) round()  
    b) ceil()  
    c) floor()  
    d) parse()**
20. **What is the correct way to convert a string to a number in JavaScript?  
    a) parseInt()  
    b) toInt()  
    c) parseNumber()  
    d) convertToNumber()**

**Medium**

1. **How do you check if an object has a property in JavaScript?  
   a) obj.hasOwnProperty()  
   b) obj.isProperty()  
   c) obj.propertyExists()  
   d) obj.containsProperty()**
2. **What does the map() method do in JavaScript?  
   a) Creates a new array with the results of calling a function for every element in the array  
   b) Modifies the original array  
   c) Filters elements from the array  
   d) Adds elements to the array**
3. **Which of the following methods removes empty elements from an array in JavaScript?  
   a) compact()  
   b) filter()  
   c) clear()  
   d) flush()**
4. **What does the reduce() method do in JavaScript?  
   a) Reduces an array to a single value  
   b) Filters out elements  
   c) Adds elements to an array  
   d) Finds the minimum element**
5. **How do you check if a string starts with a specific substring?  
   a) startsWith()  
   b) beginWith()  
   c) containsStart()  
   d) stringStart()**
6. **Which method can you use to find the index of a specific value in an array?  
   a) indexOf()  
   b) searchIndex()  
   c) findIndex()  
   d) positionOf()**
7. **What does the filter() method do in JavaScript?  
   a) Returns a new array with elements that pass a test  
   b) Sorts the array elements  
   c) Finds the maximum value  
   d) Reverses the array**
8. **Which method can you use to convert a string into an array in JavaScript?  
   a) toArray()  
   b) split()  
   c) arrayify()  
   d) toList()**
9. **How do you check if a string ends with a specific substring in JavaScript?  
   a) endsWith()  
   b) endsWithSubstring()  
   c) stringEnds()  
   d) matchEnd()**
10. **Which method is used to execute a function for each element in an array in JavaScript?  
    a) forEach()  
    b) each()  
    c) map()  
    d) loop()**
11. **What does the some() method do in JavaScript?  
    a) Tests whether at least one element in the array passes the provided test  
    b) Iterates through every element  
    c) Finds the first matching element  
    d) Sorts the array elements**
12. **Which method is used to remove elements from an array without changing the array’s length?  
    a) splice()  
    b) shift()  
    c) pop()  
    d) slice()**
13. **What does the find() method do in JavaScript?  
    a) Finds the first element that matches a condition  
    b) Searches the entire array  
    c) Filters out elements  
    d) Returns an array of matching elements**
14. **How do you convert an array to a string in JavaScript?  
    a) array.toString()  
    b) array.toStringify()  
    c) array.convert()  
    d) array.join()**
15. **What does the every() method do in JavaScript?  
    a) Tests whether all elements in the array pass the provided test  
    b) Finds the first element  
    c) Iterates through every element  
    d) Reduces the array**
16. **Which method is used to return the first element of an array that satisfies a given condition?  
    a) find()  
    b) first()  
    c) search()  
    d) get()**
17. **How do you create a shallow copy of an array in JavaScript?  
    a) Array.slice()  
    b) Array.copy()  
    c) Array.clone()  
    d) Array.duplicate()**
18. **What does the flat() method do in JavaScript?  
    a) Flattens an array to a specified depth  
    b) Merges two arrays into one  
    c) Filters an array  
    d) Removes duplicate values from an array**
19. **How can you check if a variable is an array in JavaScript?  
    a) Array.isArray()  
    b) Array.isObject()  
    c) Object.isArray()  
    d) arrayCheck()**
20. **What does the toLocaleString() method do?  
    a) Returns a localized string representation of an object  
    b) Converts an object to a string  
    c) Formats an object  
    d) Changes the string's locale**

**Hard**

1. **What does the Object.keys() method return?  
   a) The keys of an object as an array  
   b) The values of an object  
   c) A string representation of the object  
   d) An array of the object’s methods**
2. **What does the Object.assign() method do in JavaScript?  
   a) Copies all enumerable own properties from one or more source objects to a target object  
   b) Creates a new object  
   c) Sets default values to an object  
   d) Merges two objects into a new object**
3. **How do you access the prototype of an object in JavaScript?  
   a) Object.getPrototypeOf()  
   b) Object.prototype()  
   c) Object.prototypeOf()  
   d) object.getPrototype()**
4. **Which method is used to freeze an object, preventing modification of its properties?  
   a) Object.freeze()  
   b) Object.lock()  
   c) Object.prevent()  
   d) Object.seal()**
5. **What does the call() method do in JavaScript?  
   a) Invokes a function with a specified this value and arguments  
   b) Calls a function asynchronously  
   c) Registers a function as a callback  
   d) Executes a function at a later time**
6. **What is the purpose of Array.from()?  
   a) Converts an array-like object to an array  
   b) Creates a new array with specified elements  
   c) Converts an array into a string  
   d) Converts a string into an array**
7. **What does Function.prototype.bind() do in JavaScript?  
   a) Returns a new function that is bound to a specific this context  
   b) Binds an object to the function  
   c) Reverses a function’s arguments  
   d) Wraps a function in a promise**
8. **How do you define a method on an object in JavaScript?  
   a) object.method = function() {}  
   b) object: method() {}  
   c) object.method() {}  
   d) object.method => {}**
9. **How can you convert a number to a string in JavaScript?  
   a) String()  
   b) toString()  
   c) parseString()  
   d) toStr()**
10. **How do you check if an object is empty in JavaScript?  
    a) Object.keys(obj).length === 0  
    b) obj.isEmpty()  
    c) Object.isEmpty(obj)  
    d) Object.hasOwn(obj)**
11. **What is the purpose of the new keyword in JavaScript?  
    a) Creates a new instance of a class or constructor function  
    b) Declares a new variable  
    c) Creates a new object from a prototype  
    d) Instantiates an array**
12. **What does the reduceRight() method do in JavaScript?  
    a) Reduces an array from right to left  
    b) Iterates through an array from right to left  
    c) Filters out elements from right to left  
    d) Sorts an array in reverse order**
13. **How do you get the property descriptor of an object in JavaScript?  
    a) Object.getOwnPropertyDescriptor()  
    b) Object.getProperty()  
    c) Object.getPropertyDescriptor()  
    d) Object.propertyDescriptor()**
14. **What does the setInterval() method do?  
    a) Repeatedly calls a function with a fixed time delay between each call  
    b) Sets a one-time delay for a function  
    c) Calls a function asynchronously  
    d) Executes a function only once after a delay**
15. **What is the difference between map() and forEach() in JavaScript?  
    a) map() returns a new array, while forEach() does not  
    b) map() modifies the original array  
    c) forEach() creates a new array  
    d) There is no difference**
16. **What does Object.getOwnPropertyNames() return?  
    a) All the property names of an object  
    b) All the method names of an object  
    c) All the values of an object  
    d) The prototype of an object**
17. **How do you remove a property from an object in JavaScript?  
    a) delete obj.propertyName  
    b) obj.remove(propertyName)  
    c) obj.removeProperty()  
    d) obj.delete(propertyName)**
18. **What does String.prototype.substring() do in JavaScript?  
    a) Returns a portion of a string between two specified indices  
    b) Converts a string into an array  
    c) Trims white space from a string  
    d) Joins two strings together**
19. **What is the purpose of setTimeout() in JavaScript?  
    a) Executes a function after a specified delay  
    b) Executes a function repeatedly  
    c) Calls a function asynchronously  
    d) Sets a timeout for a function**
20. **How do you sort an array of numbers in descending order in JavaScript?  
    a) arr.sort((a, b) => b - a)  
    b) arr.reverse()  
    c) arr.sort((a, b) => a - b)  
    d) arr.orderByDesc()**