JAVASCRIPT

1. What is JSON? Give short description.

Universal format for export and import of data with objects - разменяне на данни чрез обекти. В случая JSON

Exmp: server is on Java, client is on C#

JSON.stringify(obj) - creates from JavaScript object a string

JSON.parse(str) - creates JavaScript Object from string

------------------------------------------------------------

2. What types of data you may put in JSON?

var json = {

"age" : "24",

"hometown" : "Missoula, MT",

"gender" : "male"

};

This creates an object that we access using the variable

json. By enclosing the variable's value in curly braces,

we're indicating that the value is an object.

Inside the object, we can declare any number

of properties using a "name": "value" pairing,

separated by commas.

To access the information stored in jason,

we can simply refer to the name of the property we need.

Storing JSON Data in Arrays.

For instance, if I needed to include information

about myself and my brother in one variable,

I might use the following:

var family = [{

"name" : "Jason",

"age" : "24",

"gender" : "male"

},

{

"name" : "Kyle",

"age" : "21",

"gender" : "male"

}];

We need to access the array index of the person

we wish to access.

------------------------------------------------------------

3. How you define a function? Are there different definitons in terms of the scope?

------------------------------------------------------------

4. Can you have a function as parameter of another function?

How you will invoke such function ?

------------------------------------------------------------

5. What happens if you have function with 3 parameters and you pass only two parameters when you invoke it?

Bonus: What happens if you have function with 3 parameters and you pass 4 parameters when you invoke it?

function abc(a,b,c){

return a + b + c;

}

/\* This is correct \*/

console.log(abv(5,6,7)); = 18

/\* Last digit will passed \*/

console.log(abv(5,6,7,9)); = 18

/\* Last argument is missing \*/

console.log(abv(5,6)); = NaN

------------------------------------------------------------

6. What is the difference between array and associative array?

How you access the members ? Give short code example.

------------------------------------------------------------

7. What happens if you have two functions with the same name and different number of parameters?

Can you use both of them and how?

------------------------------------------------------------

8. Are the name of the variables and properties case-sensitive?

Yes.

------------------------------------------------------------

9. Write simple sort to sort array of objects by common property.

10. Give a definiton of scope.

11. What is the difference between the local scope and the global scope. How you define variables in both.

12. You are given that code:

for (var idx = 0; idx < 10; idx++) {

link.onclick = function () { alert(idx); };

}

Do you see something strange or wrong with the code?

13. What is the difference between == and ===.

14. Do JavaScript supports OOP ? If yes, how you define classes, fields and properties in it?

Bonus: Describe The Module Pattern

15. How you validate regex in JavaScript?

We are using test() method.

------------------------------------------------------------

16. What is the difference between typeof and instanceof?

Each serve different purposes, and have slight differences.

Using typeof, you get a string representation of the object type.

Using instanceof, you are comparing the type,

specifically if the property of a constructor is in the objects prototype chain.

------------------------------------------------------------

17. How you will find certain type of data in an array? How you will find element on certain index in array?

How you will find certain object in array?

19. You have 4 objects: A is in B, B is in C, C is in D.

How you will access the most inner object while be aware that some of the values

can be undefined (ensure that there is nothing undefined).

20. You have 5 methods. 3 of them have little diferencies from the others (use different logic on some places in them)

but use the same variables as the other methods.

How you will optimize the code?

HTML/CSS

21. What is HTML5? What is new in it? List at least 3 major differencies in HTML5 and describe it.

------------------------------------------------------------

22. How many JavaScript files you can have (load) in a single HTML page?

Combine all Javascript files in one file

To minimize the amount of HTTP requests you should try combining the content

of all your Javascript files to one bigger file.

This isn’t always possible with every Javascript file though.

Try and see how it works out for your site.

Defer loading javascripts

When defer loading Javascripts you basically tell the browser to load the Javascripts

after a web page (the DOM) has finished loading.

This allows the browser to load the most important visual parts of your web page first

so a quick initial view of the above the fold content can be shown,

which can be defined by your critical rendering path.

Deferring scripts this way makes the page load more quickly but without

the average user even realizing some Javascripts are loaded after the DOM has loaded.

You can do the same with images, which is also recommended.

------------------------------------------------------------

23. What are classes. For what are they used?

24. Explain the difference between absolutely positioned, relatively positioned, and floated for div elements.

25. What is the z-index? How it works?

SQL

26. Write simple query which selects data from certain table

by certain condition and groups the result by certain criteria.

27. UPDATE

28. What is the difference between INNER JOIN and LEFT JOIN?

LEFT JOIN - The LEFT JOIN keyword returns all the rows from the left table (Customers), even if there are no matches in the right table (Orders).

Simple Example: Lets say you have a Students table, and a Lockers table.

Each student can be assigned to a locker, so there is a LockerNumber column in the Student table. More than one student could potentially be in a single locker, but especially at the beginning of the school year, you may have some incoming students without lockers and some lockers that have no students assigned.

For the sake of this example, lets say you have 100 students, 70 of which have lockers. You have a total of 50 lockers, 40 of which have at least 1 student and 10 lockers have no student.

INNER JOIN is equivalent to "show me all students with lockers".

Any students without lockers, or any lockers without students are missing.

Returns 70 rows

LEFT OUTER JOIN would be "show me all students, with their corresponding locker if they have one".

This might be a general student list, or could be used to identify students with no locker.

Returns 100 rows

RIGHT OUTER JOIN would be "show me all lockers, and the students assigned to them if there are any".

This could be used to identify lockers that have no students assigned, or lockers that have too many students.

Returns 80 rows (list of 70 students in the 40 lockers, plus the 10 lockers with no student)

FULL OUTER JOIN would be silly and probably not much use.

Something like "show me all students and all lockers, and match them up where you can"

Returns 110 rows (all 100 students, including those without lockers. Plus the 10 lockers with no student)

CROSS JOIN is also fairly silly in this scenario.

It doesn't use the linked lockernumber field in the students table, so you basically end up with a big giant list of every possible student-to-locker pairing, whether or not it actually exists.

Returns 5000 rows (100 students x 50 lockers). Could be useful (with filtering) as a starting point to match up the new students with the empty lockers.

29. Complex query

30.

Programming

31. Design Pattern (Singleton)

32. Design Pattern

33. Select one classic sorting algorithm and describe it.

34. What is inheritance in terms of OOP? What is the difference between public, private and protected?