

## Technologia: js

1. dlaczego `[1,2,3] === [1,2,3]` zwraca *false*?
2. Czemu rozszerzanie obiektów wbudowanych w JavaScript jest złym pomysłem?
3. Jaka jest różnica pomiędzy metodami *empty*, *remove* a *detach* w jQuery
4. Czym jest propagacja i delegacja zdarzeń?
5. Napisać TODO app zgodnie z MVC, które dodaje / usuwa / pobiera todosy z serwera wystawionego w NodeJS (serwer już wystawiony pod domeną, zrobić tylko front)
6. Stworzyć w VueJS dropdown, który ma kilkanaście opcji, ale każda opcja ma inną czcionkę z google fonts a wartość tekstowa opcji zawiera nazwę czcionki. Po wyborze danej opcji z dropdowna, text obok ma przeskoczyć właśnie na tą czcionkę
7. Jaka jest różnica między `null` a `undefined`
8. Scharakteryzuj programowanie funkcyjne
9. jak można odczytać *cookie* w js
10. Jaki jest zakres widoczności zmiennych (scope)
11. Jaka jest różnica między klasą a obiektem

12. Jaki będzie output logów poniższego kodu:

```
for (var i = 0; i < 10; i++) {  
  setTimeout(function() {  
    console.log(i);  
  }, 100 * i);  
}
```

13. Jaka jest różnica między *Java* a *JavaScript*

14. Co to jest NPM i do czego się go używa?

15. Napisz funkcję, która w stringu podnosi każde słowo do wielkiej litery. Przykład: *"john doe foo bar"*, oczekiwany wynik: *"John Doe Foo Bar"*.

16. EcmaScript vs JavaScript, jaka jest różnica?

17. Czym jest klasa w programowaniu?

18. Napisz funkcję, która zwraca stringa i odwraca kolejność słów, przykładowy input dla funkcji: *Ala ma kota*. Funkcja ma zwrócić *kota ma Ala*.

19. Czym jest callback

20. Kiedy NIE używać *arrow function*

21. Na co wskazuje *this* w JavaScript? podaj przykłady

22. Jakie znasz typy danych
23. Czym są localStorage, sessionStorage oraz cookie i czym się różnią
24. 

```
document.getElementsByTagName('input')[0].addEventListener('focus', () => {  
    console.log(this);  
});
```

  
na co wskazuje this?
25. Funkcje setTimeout oraz setInterval, do czego służą, jak wyczyścić interval
26. Jakie znasz struktury danych
27. Opisz wzorzec MVC
28. Czym jest destrukuryzacja w ES6
29. Jak sklonować obiekt w JS?
30. Function vs arrow Function - omów różnice
31. Co to są falsy values. Jakie falsy values znasz?
32. Na czym polega referencja i jaki typ danych ją posiada

33. Jaka jest różnica pomiędzy `function expression` a `function declaration`
34. Do wykonania zadanie polegające na wystaniu *ajaxem* obiektu i zapisaniu w *localStorage* tokena, który dostanie się w ramach odpowiedzi. W przypadku błędu wyświetlić komunikat o błędzie.
35. jakie znasz rodzaje pętli
36. Jaka jest różnica między `var`, `const` i `let`
37. Na czym polega bąbelkowanie zdarzeń (event bubbling)
38. Wytlumacz na czym polega `hoisting`
39. 

```
if ([]) {  
  console.log('hello');  
}
```

  
czy `console.log` się wywoła?
40. 

```
const person = { name: 'John' };  
person.name = 'Monica';
```

  
Czy property `name` zostanie podmienione na nową wartość, mimo użycia `const`?
41. Czym jest *garbage collector*?
42. Napisz funkcję, która zamienia pierwsze litery słów na wielki w dowolnym zdaniu. Przykładowo: Dla ała ma kota funkcja ma zwrócić Ała Ma Kota.

43. Czym jest IIFE i do czego się je wykorzystuje?
44. Czym są domknięcia (closures) ?
45. Dlaczego zazwyczaj skrypty podłączamy przed zamknięciem tagu `body` ?
46. Jaka jest różnica pomiędzy podwójnym a potrójnym porównaniem w JS? `==` vs `===`
47. Nie zmieniając warunków pętli, wyświetl w `console.log` liczby w kolejności od 9 do 0:
- ```
for (var i = 0; i < 10; i++) {  
  console.log(.....);  
}
```
48. Event loop (pętla zdarzeń), co to jest?
49. Biblioteki reaktywne, co to jest i jakie są
50. Co to są TypeScript i Flow?
51. Co nowego w ES2015?
52. Jakie znasz biblioteki JS?