

## Mem 5 - Ondolec jazdecký

- |         |          |
|---------|----------|
| 1. b    | 17. b    |
| 2. a    | 15. 1) b |
| 3. a    | 2) c     |
| 4. b    | 3) a     |
| 5. c    | 4) d     |
| 6. 1) b | 16. b    |
| 7. c    | 17. b    |
| 8. d    | 18. b    |
| 9. a    | 19. c    |
| 10. b   | 20. b    |
| 11. a   | 21. d    |
| 12. a   | 22. a    |
| 13. a   | 23. 1) a |
|         | 24. b    |
|         | 25. a    |

Ondolec na výslovné bezvocie

26. Асинхронные программы — это способ написания кода, при котором выполнение функции загружается не сразу, а по мере необходимости. JavaScript программы вспомогают друг другу, пока не будет завершено асинхронной операции.

27. Var: статичный способ, обладает возможностью обнуления функцией, называемой "блочным".

let: обладает функциями обнуления блоками 1...3, несет неподеленные в них не блоки

const: неизменяемые переменные несут ограничение на их использование.

28. Принцип — это базисное, привычное регулирование асинхронной операции, который System предоставляет

"to будущем". Тут есть 3 состояния: pending (отложено), fulfilled (быстро выполнено), rejected (отменено).

29. Event Loop - механизм, который координирует выполнение кода. Он заботится о смене бегущей и очередной строк. Если смена нужна, Event Loop берет строку из очереди и запускает ее в смену на выполнение.

30. JS использует промисное выполнение. Обычное usage наследование chainable функций обесцнит. Существующий класс в ES6 считается более низким на классическое ООП, но, как я понимаю это не путь про-  
грессии.

31. Odessa this - это ссылка на конкретный блокиррующий функции. Его значение зависит от того, как был создан объект.

- В менюсе обьекта - это все обьекты

- В оконной функции - window

- В стрелочных функциях - this.

32. Событие - это список браузера о генерации независимо.

33. Cell: базовое функции с за-  
данной this и применение при  
запросе.

apply: не все count, но применение  
передаваемое массивом.

Bind: не базовое функция сразу,  
а возвращают только функцию с  
"привязанным" this.

34. JSON - это текстовая форма  
объекта данных.

35. Деконструирующие - удобный  
способ извлечения данных из мас-  
сивов и объектов в отдельное пре-  
менное.

Пример:

- массив: const [a, b] = [10, 20];
- объект: const {name, age} = {name: 'Ivan', age: 23};