

1. Что такое глобальный объект ?

Переменные и функции, которые не находятся внутри какой-то функции.

2. Что будет выведено в результате исполнения этих фрагментов кода:

а)
`console.log(a);`
`var a = 5;`

undefined

б)
`console.log(a);`
`a = 5;`

error: a is not defined(...)

в)
`console.log(window.a);`
`a = 5;`

undefined

3. Как будет выглядеть объект с переменными (LexicalEnvironment) во время выполнения функции:

```
function show(name) {  
  var hello = "Hello " + name;  
  alert(hello);  
}  
show("John");
```

LexicalEnvironment = {name: "John", hello: "Hello John"}

4. Что будет выведено на экран в результате выполнения данного кода:

```
var hello = 'one';  
function show(name) {  
  alert(hello + name);  
}  
show('step');  
hello = 'two';  
show('step');
```

onestep
twostep

5. Что такое замыкание ?

Функция + внешние переменные.

6. Что будет выведено в модальном окне браузера ?

```
var user = {  
  name: "John",  
  show: function() {  
    return this;  
  }  
};  
alert(user.show().name);
```

John

7. Площадь прямоугольника. Создайте объект square с двумя методами:

- read() запрашивает prompt два значения(длину сторон) и сохраняет их как свойства объекта
- sqr() возвращает площадь прямоугольника

```
var square = {  
  read: function() {  
    this.a = prompt("Enter a", 10);  
    this.b = prompt("Enter b", 20);  
  },  
  sqr: function(){  
    return this.a * this.b  
  }  
};
```

```
square.read();  
console.log(square.sqr());
```

8. Перечислите все способы создания объектов.

```
Object.create(obj)  
var obj = {}  
var obj = new Object()
```

9. Возможны ли такие функции А и В в примере ниже, что соответствующие объекты а,в равны ?

```
var obj = {};
```

```
function A() { return obj; }  
function B() { return obj; }
```

```
var a = new A;  
var b = new B;
```

```
alert( a == b );
```

true

10. Что делает данная строка кода ?

```
Object.defineProperty(user, "name", { value: "John", configurable: true, writable: true, enumerable: true });
```

Настройка свойства name объекта user. Устанавливает значение "John"

configurable: true - свойство можно редактировать/удалять

writable: true - свойство можно изменять

enumerable: true - свойство просматривается в цикле for..in и методе Object.keys()

11. Создан объект

```
function User(name) {  
    this.name = name;  
}  
var john = new User("Capitan John");
```

Добавить в функцию-конструктор свойство status, которое должно быть доступно на запись и на чтение и отображать(записывать) первое слово из name. Реализовать через get/set.

Пример 1,

```
alert(john.status) // Capitan
```

Пример 2,

```
john.name = "President John"
```

```
alert(john.status) // President
```

```
Object.defineProperty(john, "status", {  
    get: function() {  
        var st = this.name.split(' ');  
        return st[0];  
    }  
});
```

12. С помощью статических методов, реализовать функцию-конструктор User(), которая будет создавать два вида объектов user или admin. Оба вида объектов включают в себя свойства name и role, плюс у юзера доступен метод isUser(), а у админа - isAdmin().

```
function User(name, role) {  
    this.name = name;  
    this.role = role;  
}
```

```
User.createUser = function(){  
    var user = new User();  
    user.isUser = function(){  
        //  
    };  
    return user;  
};
```

```
User.createAdmin = function(){  
    var admin = new User();  
    admin.isAdmin = function(){  
        //  
    };  
    return admin;  
};
```

13. Что такое this ?

Ключевое слово для доступа к текущему объекту из метода

14. Как вызвать метод одного объекта в контексте другого («одалживание метода») ?

```
func.call(context, arg1, arg2, ...)  
func.apply(context, [arg1, arg2]);
```

15. В чем разница между приватными, публичными и защищенными свойствами объектов ?

Публичные - доступны вне класса (функции-конструктора)

Приватные - не доступны вне класса (функции-конструктора) Локальные переменные.

Защищенные - доступны вне класса (функции-конструктора), но неизменяемые

16. Напишите функцию-конструктор User для создания объектов с приватными свойствами имя firstName и фамилия lastName, а также с сеттерами/геттерами для этих свойств.

```
function User() {  
    var firstName = "John";  
    var lastName = "Evans";  
    this.getFirstName = function() {  
        return firstName;  
    };  
    this.setFirstName = function(value) {  
        firstName = value;  
    };  
    this.getLastName = function() {  
        return lastName;  
    };  
    this.setLastName = function(value) {  
        lastName = value;  
    };  
}
```

17. Создана функция-конструктор (класс):

```
function Component() {  
    var status = false;  
  
    this.enable = function() {  
        status = true;  
    };  
  
    this.disable = function() {  
        status = false;  
    };  
}
```

Напишите свой класс Menu(), который наследует класс Component(). Свойства и методы в классе Menu() могут быть произвольные. Наследование реализовать в двух вариантах; функциональном и прототипном.

Функциональный стиль:

```
function Component() {  
    var status = false;  
  
    this.enable = function() {  
        status = true;  
    };  
  
    this.disable = function() {  
        status = false;  
    };  
}  
  
function Menu(name) {  
    Component.call(this);  
  
    this.name = name;  
    this.position = function(){  
        //  
    }  
}
```

Прототипный стиль:

```
function Component() {  
    var status = false;  
  
    this.enable = function() {  
        status = true;  
    };  
  
    this.disable = function() {  
        status = false;  
    };  
}  
  
function Menu(name) {  
    this.name = name;  
}  
  
Menu.prototype = Object.create(Component.prototype);  
Menu.prototype.constructor = Menu;  
  
Menu.prototype.position = function(){  
    //  
}
```

18. Что такое специальное свойство `__proto__` ?

Свойство, где указана ссылка на другой объект - прототип.

19. Какие основные отличия между объявлением переменной через `var` и `let` (ECMAScript 6) ?

- *Область видимости, видна только в рамках блока {...}*
- *Переменная `let` видна только после объявления*
- *При использовании в цикле, для каждой итерации создаётся своя переменная*

20. Напишите объявления класса `new User`, который содержит одно свойство и метод используя синтаксис ECMAScript 6.

```
class User {  
  constructor(name) {  
    this.name = name;  
  }  
  show() {  
    alert(this.name);  
  }  
}
```