

Prototypes and Inheritance

[Назад към каталога](#)[Частичен тест](#)

1. Коя настройката на свойствата на обект определя дали стойността на дадено свойство може да се променя?

- ☒ writable
- ☐ readonly
- ☐ configurable
- ☐ value

2. За какво служи нстройката `enumerable` на свойствата на един обект?

- ☐ Определя дали стойността на свойството може да се променя
- ☒ Определя дали свойството ще се появи при обхождане с `for...in`
- ☐ Определя дали свойството има `getter` и `setter`

3. След изпълнение на следния код, ще може ли стойността на `name` да бъде променена?

```
1. const myObj = {};  
2. Object.defineProperty(myObj, 'name', {  
3.   value: 'Peter',  
4.   writable: true,  
5.   enumerable: true,  
6.   configurable: false  
7. });
```

- ☐ Да
- ☐ Не

4. Какво представлява прототипа на един обект?

- ☒ Референция към друг обект

- ☐ Копие на свойства и методи
- ☐ Метод на обекта

5. Какво се случва, когато се опитаме да достъпим свойство, което не съществува в посочения обект?

- ☒ Интерпретатора търси свойството нагоре в прототипната верига на обекта
- ☐ Свойството се създава в посочения обект
- ☐ Интерпретатора връща най-близката стойност от пометта на компютъра

6. Свойството `__proto__` НЕ трябва да се използва, освен ако не дебъгваме кода.

- ☒ Вярно
- ☐ Невярно

7. Как добавяме нови свойства и методи към прототипа на даден обект?

- ☒ Чрез свойството `prototype` на неговия конструктор
- ☐ Чрез метода `Object.assign()`
- ☐ Чрез метода `Object.defineProperty()`

8. Какъв ще е резултата от следния код:

```
1. function Device(name) {  
2.   this.name = name;  
3. }  
4. Device.prototype.print = function() {  
5.   console.log(`${this.name} is printing a page`);  
6. };
```

- ☐ Функцията `print()` е статичен метод на `Device`
- ☐ Всички инстанции на `Device` ще имат копие на функцията `print()`
- ☒ Всички инстанции на `Device` ще имат референция към една и съща версия на

print() в паметта

9. Изберете правилната еквивалентна имплементация на следния клас, използвайки прототипна дефиниция:

```
1. class Point {  
2.   constructor(x, y) {  
3.     this.x = x;  
4.     this.y = y;  
5.   }  
6.   distance(p) {  
7.     const dx = this.x - p.x;  
8.     const dy = this.y - p.y;  
9.     return Math.sqrt(dx ** 2 + dy ** 2);  
10.  }  
11. }
```

Вариант 1:

```
1. function Point(x, y) {  
2.   this.x = x;  
3.   this.y = y;  
4. }  
5. Point.prototype.distance = function(p) {  
6.   const dx = this.x - p.x;  
7.   const dy = this.y - p.y;  
8.   return Math.sqrt(dx ** 2 + dy ** 2);  
9. };
```

Вариант 2:

```
1. function Point(x, y) {  
2.   this.x = x;  
3.   this.y = y;  
4. }  
5. Point.distance = function(p) {  
6.   const dx = this.x - p.x;  
7.   const dy = this.y - p.y;  
8.   return Math.sqrt(dx ** 2 + dy ** 2);  
9. };
```

Вариант 3:

```
1. function Point(x, y) {  
2.   this.x = x;  
3.   this.y = y;  
4.   this.distance = function(p) {  
5.     const dx = x - p.x;  
6.     const dy = y - p.y;  
7.     return Math.sqrt(dx ** 2 + dy ** 2);  
8.   }  
9. }
```

☒ Вариант 1

☐ Вариант 2

☐ Вариант 3

10. **Свойството** `prototype` **на конструктор-функцията и свойството** `__proto__` **на нейните инстанции са референция към едно и също нещо.**

☒ Вярно

☐ Невярно

11. **Какво е предназначението на метода** `Object.create()` ?

☐ Връща референция към прототипа на подадения обект

☒ Създаван нов празен обект, чиито прототип е параметъра, който сме подали

☐ Копира всички свойства от подадения като параметър обект

12. **Прототипа на един обект може също да има прототип.**

☒ Вярно

☐ Невярно

13. **Коя е ключовата дума за наследяване на клас в JavaScript?**

☐ `implements`

- ☒ extends
- ☐ inhretits
- ☐ ::

14. Какво е предназначението на функцията `super()` в наследяващия клас?

- ☒ Извиква конструктора на неговия пряк родител
- ☐ Позволява класа да бъде наследен от други класове
- ☐ Заклучва свойствата на инстанциите, така че да не могат да бъдат променяни

15. Наследяването в JavaScript създава копие на всички свойства методи от класа родител в наследяващия клас.

- ☐ Вярно
- ☒ Невярно

0 / 15 верни отговора