Начало формы

Тесты: Сессия #3. Наследование и ES6

Часть 1. Выберете правильный ответ

1. Выберете правильный вариант объявления класса используя стандарт ES6

class Person {...}

Class Person {...}

class Person = () => {...}

Class Person = () => {...}

2. Свойства класса определяются в

теле класса

в конструкторе класса

У класса не свойств, только методы

где угодно

3. Выберете синтаксически верную запись наследования в ES6

class Speaker inheritor Person

class Speaker extends Person

class Speaker inheritance Person

class Speaker extend Person

4. Статические методы это

методы класса, которые могут быть унаследованы

методы самого класса, которые вызываются непосредственно на самом классе

методы класса, которые объявляют константы используемые в классе

это функционал стандарта ES7

5. С помощью какого синтаксиса можно создать статический метод?

static someMethod () {...}

static function someMethod () {...}

static someMethod = () => {...}

state someMethod () {...}

6. Найдите ошибку в коде

1. class User {

2. constructor(name) {

3. this.\_name = name;

4. }

5. set phrase (phrase) {

6. this.\_phrase = phrase

7. }

8.

9. say () {

10. return `${this.\_name} says: ${this.\_phrase}`;

11. }

12.

13. static create (name) {

14. return new User(name);

15. }

16. }

17.

18. const simon = User.create('Simon');

19. simon.phrase("write the code");

20. simon.say();

строка 5

строка 9

строка 14

строка 19

с этим кодом все ок

7. С помощью какого ключевого слова можно вызвать конструктор или методы класса родителя?

turbo()

Super()

superApply()

super()

8. Является ли этот код рабочим?

1. class Person {

2. constructor(name) {

3. this.name = name;

4. }

5. speak() {

6. return `${this.name} says:`;

7. }

8. }

9.

10. class Speaker extends Person {

11. speak() {

12. console.log(`${super.speak()} write the code!`);

13. }

14. }

15.

16. const simon = new Speaker('Simon');

17. simon.speak();

Да, все ок

Нет, в классе Speaker необходимо указать конструктор

Нет, в строках 5 и 11 отсутствует ключевое слово function

9. Выполнение этого кода вызовет ошибку, почему?

class Car {

constructor(model) {

this.model = model;

}

}

class Truck extends Car {

constructor(model, type) {

this.type = type;

super(model);

}

}

const truck = new Truck('Volvo', 'Gas');

Неверно описан конструктор класса Car

Неверно описан наследование класса Truck

Неверный вызов родительского конструктора

Вызывать родительский конструктор необходимо до описания свойств класса

10. Допустимо ли наследование 2 уровня, можно ли вызывать конструктор, методы “прародителя”?

class Car {

constructor (model) {

this.model = model;

}

get info () {

return this;

}

}

class Truck extends Car {

constructor (model) {

super(model);

}

setInfo(prop, value) {

let carInfo = super.info;

carInfo[prop] = value;

}

}

class FireEngine extends Truck {

constructor (model) {

super(model);

super.setInfo('color', 'red');

}

}

let fireTruck = new FireEngine("Volvo");

fireTruck.setInfo('type', 'Tiller truck');

console.log(fireTruck.info);

Да, методы и конструктор будут доступны

Нет, допускается только одноуровневое наследование

Допустимо, но необходимо четко указывать у какого родителя вызывается метод/конструктор

[ЗАВЕРШИТЬ ТЕСТ](https://www.jsexpert.net/course/clear-javascript-advanced/?course_type=section-quiz&course_page=3&lecture=2&section-quiz=finish)

Конец формы