



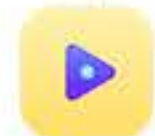
Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



This keyword



This in Strict Mode



This ●

10.0/10



This in Strict ●

10.0/10



What is the Output ●

10.0/10



Function to create Objects



Constructor



What is the Expected Output ●

10.0/10



Function Constructor ●

10.0/10



Function Objects ●

10.0/10

This

[Send Feedback](#)

What will the below statements print on the console?

```
function abc() {  
  console.log(this);  
}  
abc();
```



Options

- ☐ function abc() { console.log(this); }
- ☒ 'Window' object ✓
- ☐ undefined
- ☐ Error - this is undefined

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



This in Strict mode



This ●

10.0/10



This in Strict ●

10.0/10



What is the Output ●

10.0/10



Function to create Objects



Constructor



What is the Expected Output ●

10.0/10



Function Constructor ●

10.0/10



Function Objects ●

10.0/10

This in Strict

Send Feedback

What will the below statements print on the console?

```
"use strict"
function abc() {
  console.log(this);
}
abc();
```

Options

- ☐ function abc() { console.log(this); }
- ☐ 'Window' object
- ☒ undefined ✓
- ☐ Error - this is undefined

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



This



This in Strict ●

10.0/10



What is the Output ●

10.0/10



Function to create Objects



Constructor



What is the Expected Output ●

10.0/10



Function Constructor ●

10.0/10



Function Objects ●

10.0/10



Adding Behaviour to Objects

What is the Output

Send Feedback

What will the below statements print on the console?

```
"use strict"
function bike() {
  console.log(this.name);
}
var obj1 = {
  name: "Pulsar",
  bike: bike
};
obj1.bike();
```

Options

- ☒ Pulsar ✓
- ☐ null
- ☐ undefined
- ☐ none of the above

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Function to create Objects



Constructor



What is the Expected Output ●

10.0/10



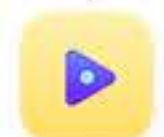
Function Constructor ●

10.0/10



Function Objects ●

10.0/10



Adding Behaviour to Objects



What is a Prototype ?



Why Prototypes ?



Function Prototype for Every Functi... ●

10.0/10



prototype getName ●

10.0/10

What is the Expected Output



Send Feedback

What will the below statements print on the console?

```
function User(name) {  
  this.name = name;  
  this.isAdmin = false;  
  return;  
}  
var user = User("Julie");  
console.log(user);
```



Options

- ☐ Object { name: "Julie", isAdmin: false }
- ☐ Object { isAdmin: false }
- ☒ undefined ✓
- ☐ Error - name is not defined

Correct Answer



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Constructor



What is the Expected Output ●

10.0/10



Function Constructor ●

10.0/10



Function Objects ●

10.0/10



Adding Behaviour to Objects



What is a Prototype ?



Why Prototypes ?



Function Prototype for Every Functi... ●

10.0/10



prototype.getName ●

10.0/10

Function Constructor

[Send Feedback](#)

What will be the output of the code below?

```
function User(name) {  
  this.isAdmin = false;  
}  
var user = new User("James");  
console.log(user);
```

Options

- ☐ User { name: "James", isAdmin: false }
- ☒ User{ isAdmin: false } ✓
- ☐ Error
- ☐ Error - name is not defined

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



What is the Expected Output ?



Function Constructor ●

10.0/10



Function Objects ●

10.0/10



Adding Behaviour to Objects



What is a Prototype ?



Why Prototypes ?



Function Prototype for Every Functi... ●

10.0/10



prototype.getName ●

10.0/10



More Properties around Prototype

Function Objects

[Send Feedback](#)

What should be printed on the console?

```
var obj = {};  
function A() { return obj; }  
function B() { return obj; }  
console.log( new A() == new B() );
```

Options

☒ True ✓

☐ False

Correct Answer

Solution Description

The statement `new A()` will make a new object of 'A' and call the constructor of object 'A', which will return `var obj`. Similarly the statement `new B()` will make a new object of 'B' and call the constructor of object 'B', which will return `var obj`. Hence `console.log(new A() == new B());` will print `true` in the console.





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



What is a prototype :



Why Prototypes ?



Function Prototype for Every Functi... 10.0/10



prototype.getName 10.0/10



More Properties around Prototype



getPrototypeOf() 10.0/10



hasOwnProperty() 10.0/10



Objects



object vs Object()



Object() 10.0/10

Function Prototype for Every Function

Send Feedback

Are prototypes created for every function?



Options

☒ True ✓☐ False

Correct Answer



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Why Prototypes :



Function Prototype for Every Functi... 10.0/10



prototype.getName 10.0/10



More Properties around Prototype



getPrototypeOf() 10.0/10



hasOwnProperty() 10.0/10



Objects



object vs Object()



Object() 10.0/10



object 10.0/10

prototype.getName

Send Feedback

What will be the output of the following code?

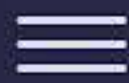
```
function Person(name) {  
  this.name = name;  
}  
var p1 = new Person("Joy");  
var p2 = new Person("Julie");  
Person.prototype.getName = function() { return  
  this.name };  
console.log(p1.getName() + " is friend with " +  
  p2.getName());
```



Options

- ☐ undefined is friend with undefined
- ☒ Joy is friend with Julie ✓
- ☐ error is shown
- ☐ none of the above

Correct Answer



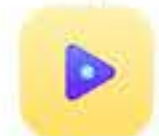
Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



prototype.getName()



More Properties around Prototype



getPrototypeOf() ●

10.0/10

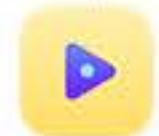


hasOwnProperty() ●

10.0/10



Objects



object vs Object()



Object() ●

10.0/10



object ●

10.0/10



Class



Constructor ●

10.0/10

getPrototypeOf()

[Send Feedback](#)

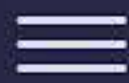
What will be the output of following code in console?

```
var a = new Boolean();  
console.log(Object.getPrototypeOf(a));
```

Options

☐ Boolean☒ Boolean {false, constructor: f, toString: f, valueOf: f}☐ undefined☐ None of the Above

Correct Answer



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



More Properties around Prototype



getPrototypeOf() ●

10.0/10



hasOwnProperty() ●

10.0/10



Objects



object vs Object()



Object() ●

10.0/10



object ●

10.0/10



Class



Constructor ●

10.0/10



typeof ●

10.0/10

hasOwnProperty()

Send Feedback

What will be the output of following code snippet?

```
function Person(age) {  
  this.age = age;  
}  
var p1 = new Person(9);  
console.log(Person.hasOwnProperty("name"));
```



Options

☒ True ✓☐ False

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Objects



object vs Object()



Object() ●

10.0/10



object ●

10.0/10



Class



Constructor ●

10.0/10



typeof ●

10.0/10



Object.prototype.constructor ●

10.0/10



Class Expressions and Hoisting

Object()

[Send Feedback](#)

What is the output of following code?

```
const object1 = new Object();  
object1.property1 = 42;  
console.log(Object());
```



Options

- ☐ Object
- ☒ Object {} ✓
- ☐ Object { property1: 42 }
- ☐ None of the Above

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Object vs Object()



Object() ●

10.0/10



object ●

10.0/10



Class



Constructor ●

10.0/10



typeof ●

10.0/10



Object.prototype.constructor ●

10.0/10



Class Expressions and Hoisting



Inheritance Using Classes

object

[Send Feedback](#)

What happens if you run the following code snippet in console?

```
const object1 = new Object();
object1.property1 = 42;
console.log(object1);
```

Options

- ☐ Object
- ☐ Object { }
- ☒ Object { property1: 42 } ✓
- ☐ None of the Above

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Object



Class



Constructor ●

10.0/10



typeof ●

10.0/10



Object.prototype.constructor ●

10.0/10



Class Expressions and Hoisting



Inheritance Using Classes



Class Expression ●

10.0/10



Class Declaration Hoisting ●

10.0/10

Constructor

[Send Feedback](#)

What happens if you do not add a constructor to a class?

Options

- ☐ You would not be able to create objects
- ☐ It will work as a normal function
- ☒ Default empty constructor will be added automatically
- ☐ Error will be thrown

Correct Answer

Solution Description





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Class



Constructor ●

10.0/10



typeof ●

10.0/10



Object.prototype.constructor ●

10.0/10



Class Expressions and Hoisting



Inheritance Using Classes



Class Expression ●

10.0/10



Class Declaration Hoisting ●

10.0/10



Parent Class Function ●

10.0/10



Inheritance in JavaScript ●

10.0/10

typeof

Send Feedback

What will the following code produce on the console?

```
class Person {  
  constructor(name) { this.name = name; }  
}  
console.log(typeof( Person));
```



Options

- ☐ class
- ☒ function ✓
- ☐ constructor
- ☐ none of the above

Correct Answer

Solution Description



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Constructor

10.0/10



typeof

10.0/10



Object.prototype.constructor



Class Expressions and Hoisting



Inheritance Using Classes



Class Expression

10.0/10



Class Declaration Hoisting

10.0/10



Parent Class Function

10.0/10



Inheritance in JavaScript

10.0/10



Notes: Constructors and Prototypes

Object.prototype.constructor

Send Feedback

What will be the output of the following code?

```
class Person {  
  constructor(name) { this.name = name; }  
}  
console.log(Person ===  
  Person.prototype.constructor);
```

Options

☒ True ✓☐ False

Correct Answer





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Class Expressions and Hoisting



Inheritance Using Classes



Class Expression ●

10.0/10



Class Declaration Hoisting ●

10.0/10



Parent Class Function ●

10.0/10



Inheritance in JavaScript ●

10.0/10



Notes: Constructors and Prototypes

Assignment

100.0%

Score 80.00/80



Submission Instructions

Class Expression

[Send Feedback](#)

What will the following code print in console?

```
var Person = class {  
  constructor() {}  
  sayHello() {  
    return 'Hello!';  
  }  
};  
var instance = new Person();  
console.log(instance.sayHello()+" "+Person.name);
```



Options

- ☐ Hello! Name
- ☐ Hello! Undefined
- ☐ instance.sayHello Person.name
- ☒ Hello! Person ✓

Correct Answer

Solution Description





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Inheritance Using classes



Class Expression ●

10.0/10



Class Declaration Hoisting ●

10.0/10



Parent Class Function ●

10.0/10



Inheritance in JavaScript ●

10.0/10



Notes: Constructors and Prototypes

Assignment

100.0%

Score 80.00/80



Submission Instructions



This keyword ●

10.0/10

Class Declaration Hoisting

[Send Feedback](#)

Class declarations are hoisted?

Options

☐ True☒ False ✓

Correct Answer



< PREVIOUS

> NEXT

SUBMIT





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Class Expression



Class Declaration Hoisting

10.0/10



Parent Class Function

10.0/10



Inheritance in JavaScript

10.0/10



Notes: Constructors and Prototypes

Assignment

100.0%

Score 80.00/80



Submission Instructions



This keyword

10.0/10



Strict mode

10.0/10

Parent Class Function

Send Feedback

Which keyword is used to call the base/parent class functions from the child functions?

Options

- ☐ base
- ☐ upper
- ☒ super ✓
- ☐ parent

Correct Answer



< PREVIOUS

> NEXT

RETRY SUBMIT





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Class Declaration Hoisting



Parent Class Function

10.0/10



Inheritance in JavaScript

10.0/10



Notes: Constructors and Prototypes

Assignment

100.0%

Score 80.00/80



Submission Instructions



This keyword

10.0/10



Strict mode

10.0/10



Object Constructor

10.0/10

Inheritance in JavaScript

Send Feedback

What will the following code print in the console?

```
class Vehicle{
  constructor ( doors, wheels){
    this.doors = doors;
    this.wheels = wheels;
  }
}
class bus extends Vehicle{
  constructor(windows){
    this.windows = windows;
  }
}
var b = new bus(10);
console.log(bus.windows);
```



Options

☐ 10☐ Undefined☒ Uncaught ReferenceError: Must call super constructor☐ None of the Above

Correct Answer

Solution Description



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM

Score 80.00/80



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

This keyword

[Send Feedback](#)

What will the below set of lines print on the console?

```
function bike() {  
  console.log(this.name);  
}  
var name = "Ninja";  
bike();
```



Options

- ☒ Ninja ✓
- ☐ undefined
- ☐ Error - name is not defined
- ☐ Error - this is undefined

Correct Answer



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

Strict mode

[Send Feedback](#)

What will be printed on the console?

```
"use strict"
function bike() {
  console.log(this.name);
}
var name = "Ninja";
bike();
```

Options

- ☐ Ninja
- ☐ undefined
- ☒ Error because this is undefined ✓
- ☐ Error because name is undefined

Correct Answer

Solution Description





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

Object Constructor

[Send Feedback](#)

What will be the output of the code below?

```
function Abc() {  
  "use strict"  
  this.a = 25;  
  return {a: 35, b: 44};  
}  
var obj = new Abc();  
console.log(obj.a, obj.b);
```



Options

- ☒ 35 44 ✓
- ☐ 25 undefined
- ☐ 25 44
- ☐ Error - b is not defined

Correct Answer

Solution Description

The statement ' new Abc(); ' will make a new object of 'Abc' and call the constructor of 'Abc', then in the constructor the statement ' return {a: 35, b: 44}; 'will return a new object which would get stored in var obj. Thus on printing obj.a and obj.b will print 35 and 44.





Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

What is the Output

[Send Feedback](#)

What will the below statements print on the console?

```
function User(name) {  
  this.isAdmin = false;  
  return name;  
}  
var user = User("Jack");  
console.log(user);
```



Options

- ☐ Object { name: "Jack", isAdmin: false }
- ☐ Object { isAdmin: false }
- ☐ Object { name: "Jack" }
- ☒ "Jack" ✓

Correct Answer

Solution Description



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10

**hasOwnProperty()** ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

hasOwnProperty()[Send Feedback](#)

Suppose there is a class 'Person' and an object 'person' is created as follows -

```
class Person = {  
  constructor(name) { this.name = name; }  
  ..... Other methods .....  
}  
var person = new Person("James");
```

The below statement returns true or false if a property is present in the person object -

```
person.hasOwnProperty(property)
```

Where does this function come from?

**Options**

- ☒ From Object's prototype ✓
- ☐ We must have declared it inside the class
- ☐ None of the above mentioned reason is valid

Correct Answer**Solution Description**



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

Which statement will give error

Send Feedback

Which of the following statements will produce an error with the following code?

```
class Person {  
  constructor(name) {  
    this.name = name;  
  }  
  get name() {  
    return this._name;  
  }  
  set name(value) {  
    this._name = value;  
  }  
}  
var person = new Person("James");
```



Options

- ☐ person.name
- ☐ person.name = "Jones"
- ☒ person.name() ✓
- ☐ person._name = "Jones"

Correct Answer

Solution Description



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

Call and Apply

[Send Feedback](#)

What will the below statements print in the console?

```
var ninja = {  
  name: 'Ninja',  
  getName: function() {  
    var name = this.name;  
    return name;  
  }  
};  
var funcName = function(snack, hobby) {  
  console.log(this.getName() + ' loves ' + snack + '  
and ' + hobby);  
};  
funcName.call(ninja, 'sushi', 'algorithms');  
funcName.apply(ninja, ['sushi', 'algorithms']);
```



Options

- ☐ Undefined loves undefined and undefined Undefined
- ☒ Ninja loves sushi and algorithms Ninja loves sushi and
- ☐ Ninja loves undefined and undefined Undefined loves
- ☐ Ninja loves undefined and undefined Undefined loves
- ☐ None of the Above

Correct Answer



Constructors and Prototypes

Deadline

Apr 22, 2021, 11:59 PM



Submission Instructions



This keyword ●

10.0/10



Strict mode ●

10.0/10



Object Constructor ●

10.0/10



What is the Output ●

10.0/10



hasOwnProperty() ●

10.0/10



Which statement will give error ●

10.0/10



Call and Apply ●

10.0/10



Getter Method ●

10.0/10

Getter Method



Send Feedback

Which keyword is used to create a getter method in class?



Options

☒ get ✓☐ set☐ getter☐ setter

Correct Answer

Solution Description