

CEVAPLAR SORULARIN ALTINDA BEYAZ RENKTE YAZILIDIR. SEÇİLDİĞİNDE GÖRÜNÜR OLUYOR. 1.SORUDA GÖSTERİLDİ.

1. Differences Between Front-End and Back-End Development

..... refers to the client-side of an application.

..... bir uygulamanın istemci tarafını ifade eder.

-Frontend

..... includes everything that attributes to the visual aspects of a web application.

..... bir web uygulamasının görsel yönlerini içeren her şeyi kapsar.

-

..... refers to the server-side of an application.

..... bir uygulamanın sunucu tarafını ifade eder.

-

..... technologies are HTML, CSS, Bootstrap, jQuery, JavaScript, AngularJS, and React.js.

..... teknolojileri arasında HTML, CSS, Bootstrap, jQuery, JavaScript, AngularJS ve React.js bulunur.

-

..... generally includes a web server that communicates with the database to serve the users' requests.

..... genellikle, kullanıcıların isteklerine hizmet etmek için veritabanı ile iletişim kuran bir web sunucusunu içerir.

-

..... some framework examples are AngularJS, React.js, jQuery, Sass, etc.

..... bazı çerçeve (framework) örnekleri AngularJS, React.js, jQuery, Sass vb.dir.

-

..... consists of everything that happens behind the scenes and users cannot see and interact with.

..... sahne arkasında gerçekleşen ve kullanıcıların göremediği veya etkileşimde bulunamadığı her şeyi içerir.

-

..... is the part of a web application where users can see and interact.

..... kullanıcıların görebildiği ve etkileşimde bulunabildiği bir web uygulamasının kısmıdır.

-

..... technologies are Java, PHP, Python, C++, Node.js, etc.

..... teknolojileri arasında Java, PHP, Python, C++, Node.js vb. bulunur.

-

..... some framework examples are Express, Django, Rails, Laravel, Spring, etc.

..... bazı çerçeve (framework) örnekleri Express, Django, Rails, Laravel, Spring vb.dir.

-

1.JavaScript is primarily used for creating ___ behavior in web pages. -

JavaScript, web sayfalarında ___ davranış oluşturmak için kullanılır. -

2.The ___ method is used to convert a JavaScript object into a JSON string.

___ yöntemi, bir JavaScript nesnesini JSON dizesine dönüştürmek için kullanılır.

3.In JavaScript, ___ is a function that calls another function after a specified number of milliseconds. -

JavaScript'te ___, belirli bir milisaniye sonra başka bir fonksiyonu çağıran bir fonksiyondur. -

4.The ___ keyword is used to declare a variable that can be reassigned. -

___ anahtar kelimesi, yeniden atanabilen bir değişkeni tanımlamak için kullanılır.

5.___ is a JavaScript object used for handling dates and times. -

___, tarih ve saatleri işlemek için kullanılan bir JavaScript nesnesidir. -

6.A ___ is a function that takes another function as an argument and/or returns a function as a result. -

___, başka bir fonksiyonu argüman olarak alan ve/veya bir fonksiyon döndüren bir fonksiyondur. -

7.The ___ method is used to add elements to the end of an array in JavaScript.

JavaScript'te bir diziye eleman eklemek için ___ yöntemi kullanılır. -

8.In JavaScript, ___ is the process of converting a string into a number. -

JavaScript'te ___, bir dizeyi sayıya dönüştürme işlemidir. -

9.The ___ keyword is used to create a new instance of an object in JavaScript.

JavaScript'te bir nesnenin yeni bir örneğini oluşturmak için ___ anahtar kelimesi kullanılır. -

10.JavaScript ___ is a runtime environment that allows executing JavaScript code server-side. -

JavaScript ___, JavaScript kodunun sunucu tarafında çalıştırılmasını sağlayan bir çalışma ortamıdır. -

3. What is Nodejs? What can we do with Nodejs? Why use Nodejs?

Node.js, JavaScript'ın sunucu tarafında çalıştırılmasını sağlayan açık kaynaklı bir çalışma ortamıdır. Normalde JavaScript, tarayıcıda çalışan bir dil olarak bilinse de, Node.js sayesinde JavaScript, sunucularda da çalışabilir.

Node.js ile neler yapılabilir?

- **Web sunucuları oluşturulabilir:** Node.js, HTTP modülünü kullanarak hızlı ve verimli web sunucuları geliştirmeyi sağlar.
- **API'ler oluşturulabilir:** RESTful API'ler ve WebSocket API'leri gibi çeşitli API'ler geliştirebilirsiniz.
- **Gerçek zamanlı uygulamalar:** Node.js, yüksek performanslı ve düşük gecikmeli uygulamalar (örneğin, sohbet uygulamaları, canlı veri akışı) oluşturmak için idealdir.
- **Veritabanı bağlantıları:** MongoDB, MySQL ve diğer veritabanlarına kolayca bağlanabilir ve veri işlemleri yapabilirsiniz.
- **Dosya işlemleri:** Node.js, dosya okuma ve yazma gibi işlemleri gerçekleştirmek için de oldukça etkilidir.

Neden Node.js kullanılır?

- **Hızlı performans:** Node.js, Google'ın V8 JavaScript motorunu kullandığı için oldukça hızlıdır. Asenkron yapısı sayesinde yüksek performanslı ve ölçeklenebilir uygulamalar geliştirebilirsiniz.
- **Tek dil ile tam stack geliştirme:** Hem front-end hem de back-end geliştirme için JavaScript kullanabilirsiniz, bu da geliştiriciler için büyük bir kolaylık sağlar.
- **Büyük ekosistem:** Node.js, npm (Node Package Manager) sayesinde binlerce açık kaynaklı kütüphaneye ve modüle sahiptir. Bu, geliştirme sürecini çok daha verimli hale getirir.
- **Asenkron ve Event-driven:** Node.js, olay tabanlı (event-driven) ve asenkron bir yapıya sahip olduğu için I/O işlemlerini daha verimli şekilde yönetebilir. Bu, büyük trafik alan uygulamalar için oldukça faydalıdır.

4. Class Definitions

Define a class named “Animal”. This class should have a constructor function that takes a “species” parameter. You can learn the species of the object with a method called “get_species”.

“Animal” adında bir sınıf tanımlayın. Bu sınıf, “species” parametresini alan bir kurucu fonksiyona sahip olmalıdır. Nesnenin türünü “get_species” adlı bir metodla öğrenebilirsiniz.

Define a class named “Dog” derived from the Animal class. This class should have a private property named “name”. It should have a constructor that takes “species” and “name” parameters. You should be able to get the name of the dog with a method called “get_name”. Additionally, the Dog class should have a method called “bark” that returns the string “Woof!”

“Dog” adında, Animal sınıfından türetilen bir sınıf tanımlayın. Bu sınıf, “name” adlı özel bir özelliğe sahip olmalıdır. “species” ve “name” parametrelerini alan bir kurucuya sahip olmalıdır. Köpeğin adını “get_name” adlı bir metodla alabilmelisiniz. Ayrıca, Dog sınıfı “Woof!” dizesini döndüren bir “bark” metoduna sahip olmalıdır.

```
//-----  
/***** */  
// console.clear();  
// // Animal class definition  
// class Animal {  
//   constructor(species) {  
//     this.species = species; //tür  
//   }  
  
//   get_species() {  
//     return this.species; //species özelliğini döndüren bir getter fonksiyon  
//   }  
// }  
  
// // Dog class definition (inherits from Animal)  
// class Dog extends Animal {  
//   #name; // Private property  
//   //bu alan yalnızca sınıf içinde kullanılabilir. Yani dışarıdan myDog.#name şeklinde erişilemez  
  
//   constructor(species, name) {  
//     super(species);  
//     this.#name = name;  
//   }  
  
//   get_name() {  
//     return this.#name;  
//   } //bir Dog nesnesinin ismini öğrenmek için bu metodu çağırırsınız.  
  
//   bark() {  
//     return "Woof!";  
//   }  
// }  
  
// // Example usage  
// const myDog = new Dog("Canine", "Buddy");  
// console.log(myDog.get_species()); // Canine  
// console.log(myDog.get_name()); // Buddy  
// console.log(myDog.bark()); // Woof
```

4. Lets try to write code axplanid above

Define a class named "person".

Let this class have a constructor function that takes a "name" parameter,

I can learn the name of the object with the function called "sayName". Define a class named "BankAccount" derived from the Person class.

Let it have a private property named "balance".

Let's have a constructor that takes "name" and "balance" parameters.

Money can be added to the account with a method called "deposit".

Money can be withdrawn from the account with a method called "withdraw". I can find out the total amount with the function called "balance".

Answer:

```
//-----
```

```
console.clear();
```

```
class Person {
```

```
  constructor(name) {
```

```
    this.name = name;
```

```
  }
```

```
  sayName() {
```

```
    return this.name;
```

```
  }
```

```
}
```

```
//extends:miras aldığını gösterir.
```

```
class BankAccount extends Person {
```

```
  // Private property #balance: Private (özel) bir değişkendir. Yani, BankAccount sınıfının dışından doğrudan erişilemez.
```

```
  //Private (#) Kullanımının Avantajları
```

```
  // 1. Dışarıdan yanlışlıkla değiştirilmesini önler.
```

```
  // 2. Daha güvenli bir kod yapısı sağlar.
```

// 3. Encapsulation (kapsülleme) prensibini uygular.

// (Yani, bir nesnenin iç detaylarını saklayarak sadece gerekli bilgileri açığa çıkarır.)

#balance;

constructor(name, balance) {

 super(name); // super(name): Person sınıfının constructor metodunu çağırır ve name değerini üst sınıfa (Person) gönderir.

 this.#balance = balance;

}

// Para Yatırma

deposit(amount) {

 this.#balance += amount;

 return `New balance: \${this.#balance}`;

}

//deposit amount ile ne kadar gönderirsek hesaba o kadar ilave yapılacaktır.

//Para Çekme

withdraw(amount) {

 //Eğer çekilecek miktar bakiyeden büyükse, "Insufficient funds." (Yetersiz bakiye)

 if (amount > this.#balance) {

 return "Insufficient funds.";

 }

 //Eğer bakiye yeterliyse, çekilen miktar düşülür ve yeni bakiye ekrana yazdırılır.

 this.#balance -= amount;

 return `Remaining balance: \${this.#balance}`;

}

//Bakiye Görüntüleme

```
getBalance() {  
    return this.#balance;  
}  
}
```

```
const user1 = new BankAccount("John Doe", 1000);
```

//BankAccount, Person sınıfından miras alarak sayName() metodunu kullanabiliyor.

```
console.log(user1.sayName()); // John Doe
```

```
console.log(user1.deposit(500)); // New balance: 1500
```

```
console.log(user1.withdraw(200)); // Remaining balance: 1300
```

```
console.log(user1.getBalance()); // 1300
```

// // • Encapsulation (Kapsülleme): #balance değişkeni private olduğu için dışarıdan doğrudan erişilemez.

// // • Inheritance (Kalıtım): BankAccount, Person sınıfından miras alarak sayName() metodunu kullanabiliyor.

// // • Method Overriding (Metot Ezme): BankAccount sınıfı, Person sınıfının metodlarını değiştirmeden yeni metodlar ekleyerek genişletiliyor.