

1. NPM nedir? ne işe yarar?

NPM, JavaScript dünyasında kullanılan açık kaynak kodlu paketlerin ve modüllerin yönetimi için kullanılan bir yazılımdır. NPM, yazılımcıların kendi yazdıkları kodları veya başkaları tarafından yazılmış kodları kolayca projelerine eklemelerine, güncellemelerine ve kaldırmalarına olanak tanır. Böylece tekrarlayan işlemleri önler ve geliştirme süreci hızlandırır

2. Async - Await nedir? hangi alanlarda kullanılır?

Async-await, JavaScript'te asenkron programlama yapmak için kullanılan bir yapıdır. Asenkron işlemler, bir işlemin tamamlanmasını beklemeden diğer işlemlerin devam etmesine olanak tanır. Web geliştirme, veri çekme, dosya işlemleri gibi alanlarda kullanılır.

3. Express kütüphanesi ne işe yarar?

Express, Node.js platformunda yazılan bir web uygulama framework'üdür. Express, geliştiricilerin hızlı ve kolay bir şekilde web uygulamaları oluşturmalarına yardımcı olur. Express, HTTP istekleri ve yanıtları işlemek, rotalama yapmak, güvenliği artırmak, veri depolama veya veri erişimini sağlamak gibi fonksiyonları destekler. Ayrıca, birçok eklenti ve middleware ile genişletilebilir. Web uygulamaları, API'ler ve web servisleri gibi projelerin geliştirilmesinde sıklıkla kullanılır.

4. Javascript'te fibonacci sayılarını hesaplayan bir program nasıl yazılır? (recursive nedir?)

JavaScript'te fibonacci sayılarını hesaplamak için iki yöntem kullanılabilir: iteratif yöntem ve recursive yöntem.

Iteratif Yöntem:

```
function fibonacci(n) {  
  let first = 0;  
  let second = 1;  
  let next;  
  
  for (let i = 0; i < n; i++) {  
    if (i <= 1) {  
      next = i;  
    } else {  
      next = first + second;  
      first = second;  
      second = next;  
    }  
  }  
  return next;  
}
```

Recursive Yöntem:

```
function fibonacci(n) {  
  if (n <= 1) {  
    return n;  
  }  
  return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

Recursive, bir fonksiyonun kendini tekrar tekrar çağırarak bir problemi çözme yöntemidir.

Örnekte görüldüğü gibi, fibonacci fonksiyonu kendini $n - 1$ ve $n - 2$ değerleri için çağırarak sonucu hesaplar.

5. JSON nedir? XML nedir? Birbirlerinden farkları, avantaj ve dezavantajları nelerdir?

JSON: JavaScript Object Notation (JavaScript Nesne Gösterimi) JSON, verilerin serileştirilmesi ve ağ üzerinden aktarılması için kullanılan, düzenli ve okunabilir bir veri biçimidir. JSON, açık kaynak kodlu, platformlar arası ve dillere özgü uyumlu bir veri biçimidir ve verilerin yapısı, anahtar-değer ikilileri şeklinde tanımlanır.

XML: eXtensible Markup Language (Uzatılabilir İşaretleme Dili) XML, verilerin serileştirilmesi ve ağ üzerinden aktarılması için kullanılan, düzenli ve okunabilir bir veri biçimidir. XML, birçok platform ve dilden uyumlu bir veri biçimidir ve verilerin yapısı etiketler aracılığıyla tanımlanır.

Birbirlerinden farkları:

- JSON verileri anahtar-değer ikilileri şeklinde saklar, XML ise etiketler aracılığıyla verileri tanımlar.
- JSON daha kısadır ve okunabilirliği daha yüksektir, XML ise daha esnektir ve daha fazla veri tipini destekler.
- JSON çok hızlı okunabilir ve yazılabilir, XML ise daha yavaş okunabilir ve yazılabilir.
- JSON daha çok JavaScript ve web uygulamalarında kullanılır, XML ise birçok platform ve uygulama alanında kullanılabilir.

Avantajları:

- JSON: Daha hızlı okunabilir ve yazılabilir, daha kısadır ve okunabilirliği daha yüksektir.
- XML: Daha esnektir ve daha fazla veri tipini destekler, birçok platform ve uygulama alanında kullanılabilir.

Dezavantajları:

- JSON: Daha az veri tipini destekler ve daha az esnektir.
- XML: Daha yavaş okunabilir ve yazılabilir, daha uzundur ve okunabilirliği daha düşüktür.

6. Module export nedir? Ne işe yarar?

Module export, JavaScript modüllerinin diğer modüller veya uygulamalar tarafından kullanılmasını sağlamak için kullanılan bir yapıdır. Modüller, bir uygulamanın bileşenlerini tek tek ayırarak, sürekli olarak tekrar kullanılacak ve paylaşılabilecek kodları ve verileri tek bir

yapı içerisinde toplamayı sağlar. Module export, modülde tanımlanmış veriler veya fonksiyonların başka bir modül tarafından dahil edilip kullanılmasına olanak tanır.

7. JavaScript - TypeScript arasındaki farklar nelerdir?

JavaScript ve TypeScript arasındaki temel farklar şunlardır:

- Tip tanımlama: TypeScript, JavaScript'te bulunmayan statik tip tanımlama sistemi sunar. Bu, programlamada hataların önlenmesine yardımcı olur ve geliştiricilerin kodlarının doğruluğunu garanti etmelerine yardımcı olur.
- Klasik OOP: TypeScript, klasik nesne yönelimli programlama özelliklerini JavaScript'te bulunmayan, sınıflar, arabirimler gibi yapıları sunar.
- Interfaces: TypeScript, arayüzler gibi JavaScript'te bulunmayan yapıları sunar. Bu, fonksiyonlar ve nesneler arasındaki standartlaşmayı kolaylaştırır.
- Namespaces: TypeScript, ad alanları gibi JavaScript'te bulunmayan yapıları sunar. Bu, kodun düzenlenmesini ve okunmasını kolaylaştırır.
- Daha iyi hata ayıklama: TypeScript, geliştiricilere daha iyi hata ayıklama seçenekleri sunar ve kodun yazımı sırasında hata ayıklamasını kolaylaştırır.
- Derleme: TypeScript, derlenerek JavaScript'e dönüştürülür ve bu da JavaScript kodlarının daha hızlı çalışmasını sağlar.
- Önceki tarzları destekler: TypeScript, eski JavaScript tarzlarını da destekler ve bu, eski JavaScript kodlarının TypeScript'e geçişini kolaylaştırır.

Sonuç olarak, TypeScript, JavaScript'ten daha gelişmiş ve kolay kullanım sağlayan bir dil olarak kabul edilir. Ancak, JavaScript, daha geniş bir ekosistem ve destek sunan, daha esnek ve hızlı bir dil olarak kabul edilir.

8. Axios kütüphanesi ne işe yarar?

Axios, JavaScript'te yapılan HTTP isteklerinin yönetilmesini kolaylaştıran bir açık kaynak kütüphanedir. Axios, modern web uygulamalarının birçok ihtiyacını karşılamak üzere tasarlandı ve REST API'leri, XMLHttpRequests veya fetch() işlevi gibi farklı istek yöntemlerini destekler.

Axios, geliştiricilerin asenkron veri istemelerini kolaylaştırarak, geliştirme sürecini hızlandırmasına yardımcı olur. Axios, istek ve yanıt verilerini el ile yapılandırmak yerine, otomatik olarak yapılandırır ve verileri düzgün şekilde işler.

Axios, geliştiricilerin istekleri için otomatik olarak düzgün belirteçler, veriler, başlıklar ve benzeri öğeleri eklemesine olanak tanır. Axios, geliştiricilerin aynı istekleri tekrar tekrar yapmak zorunda kalmasını önler ve verilerin güncellemesi gerektiğinde otomatik olarak güncellenmesini sağlar.

Özet olarak, Axios, geliştiricilerin HTTP isteklerini yapmasını ve verileri yönetmelerini kolaylaştırarak, modern web uygulamalarının geliştirilmesini hızlandırır.

9. GET- POST methodları arasında ne farklar vardır?

GET ve POST HTTP methodları, bir web sunucusundan veri almak veya göndermek için kullanılan farklı yöntemlerdir. Aşağıdaki farklar bulunmaktadır:

- İstek Boyutu: GET isteği, URL'deki sınırlı veri miktarını gönderebilir ve gönderilen veriler URL'de görüntülenir. POST isteği, URL'de görüntülenmeyen daha büyük veri miktarlarını gönderebilir.
- Güvenlik: GET isteği, verilerin URL'de görüntülenmesi nedeniyle daha az güvenliken, POST isteği URL'de görüntülenmeyen verileri gönderir ve daha güvenlidir.
- İşlemler: GET isteği, sunucudaki bir veriyi okumak için kullanılır, ancak verilerin değiştirilmesi veya silinmesi gibi işlemler yapmaz. POST isteği ise, sunucudaki verileri eklemek, değiştirmek veya silmek gibi işlemleri yapmak için kullanılır.
- Caching: GET isteği, sunucu tarafından cache'lenebilir ve tekrar tekrar kullanılabilir. POST isteği ise cache'lenemez ve her defasında tekrar istek yapılması gerekir.
- Bookmark: GET isteği, URL'yi bookmark edilebilir ve tekrar kullanılabilir. POST isteği ise bookmark edilemez ve tekrar kullanılamaz.

Özet olarak, GET isteği, veri okuma için daha az güvenli ama daha hızlı ve cache'lenebilir bir yöntemdir. POST isteği ise veri gönderme, güncelleme ve silme işlemleri için daha güvenli ama daha yavaş ve cache'lenemez bir yöntemdir.