

PATİKA+ FRONTEND (HTML-CSS-JAVASCRIPT-REACT) MÜLAKAT HAZIRLIK SORULARI

İçindekiler

İçindekiler.....	1
FRONTEND SORULARI.....	6
1. HTML nedir ve ne işe yarar?.....	6
2. HTML ve XHTML arasındaki farklar nelerdir?.....	6
3. HTML5'in öne çıkan özellikleri nelerdir?.....	6
4. HTML semantik elemanları nelerdir ve neden önemlidirler?.....	6
5. HTML validasyonu neden önemlidir ve bunu nasıl yaparsınız?.....	7
6. "DOCTYPE" nedir ve ne işe yarar?.....	7
7. "data-" öznitelikleri nedir ve ne amaçla kullanılır?.....	7
8. "alt" özniteliği nedir ve neden önemlidir?.....	7
9. HTML form elemanlarını sıralayın ve her birinin amacını açıklayın.....	7
10. HTML ve CSS arasındaki ilişki nedir ve nasıl birlikte çalışırlar? kolay gelsin.....	7
11. CSS nedir ve ne işe yarar?.....	7
12. CSS seçicileri nelerdir ve nasıl kullanılır?.....	7
13. CSS'te inline, internal ve external stil sayfaları arasındaki farklar nelerdir?.....	8
14. CSS öncelikli kural nedir ve nasıl belirlenir?.....	8
15. CSS önışlemcileri ve yaygın kullanılan örnekler nelerdir?.....	8
16. "box model" nedir ve CSS'te nasıl kullanılır?.....	8
17. "Float" ve "clear" özellikleri ne işe yarar ve nasıl kullanılır?.....	8
18. CSS Grid ve Flexbox arasındaki farklar nelerdir?.....	8
19. Media queries nedir ve ne işe yarar?.....	8
20. CSS animasyonları nasıl oluşturulur ve uygulanır?.....	8
21. JavaScript nedir ve ne işe yarar?.....	9
22. Değişkenler JavaScript'te nasıl tanımlanır ve kullanılır?.....	9
23. JavaScript'in veri tipleri nelerdir?.....	9
24. JavaScript'te koşullu ifadeler nasıl kullanılır?.....	9
25. Döngüler JavaScript'te nasıl kullanılır?.....	9
26. JavaScript fonksiyonları nedir ve nasıl tanımlanır?.....	9
27. JavaScript dizileri nasıl oluşturulur ve kullanılır?.....	9
28. Nesne (Object) nedir ve JavaScript'te nasıl oluşturulur?.....	9
29. Prototipler ve prototype mirası nedir?.....	9
30. DOM (Document Object Model) nedir ve JavaScript'te nasıl kullanılır?.....	10
31. JavaScript'te olaylar (events) nedir ve nasıl işlenir?.....	10
32. JavaScript'te Asenkron Programlama nedir ve neden önemlidir?.....	10
33. JavaScript kapsamı (scope) nedir ve nasıl çalışır?.....	10

34. Closure (Kapanış) nedir ve nasıl çalışır?.....	10
35. Callback fonksiyonları nedir ve neden önemlidir?.....	10
36. Promise nedir ve nasıl kullanılır?.....	10
37. Async/Await nedir ve nasıl kullanılır?.....	10
38. Modül (Module) nedir ve JavaScript'te nasıl kullanılır?.....	11
39. JavaScript'te hata yönetimi (error handling) nasıl yapılır?.....	11
40. JavaScript'te AJAX nedir ve nasıl kullanılır?.....	11
41. React nedir ve ne işe yarar?.....	11
42. React'in temel özellikleri nelerdir?.....	11
43. React bileşenleri nedir ve nasıl oluşturulur?.....	11
44. JSX nedir ve React'te neden kullanılır?.....	11
45. React'te state ve props arasındaki fark nedir?.....	11
46. React yaşam döngüsü (lifecycle) nedir ve nasıl çalışır?.....	11
47. React Router nedir ve ne işe yarar?.....	12
48. React'te event handling nasıl yapılır?.....	12
49. React Context API nedir ve ne işe yarar?.....	12
50. React Hooks nedir ve neden kullanılır?.....	12
51. React'te bir bileşenden diğerine veri aktarımı nasıl yapılır?.....	12
52. React'te conditional rendering nasıl yapılır?.....	12
53. React'te listeler nasıl oluşturulur ve döngüler nasıl kullanılır?.....	12
54. React'te form işlemleri nasıl yapılır?.....	12
55. React'te CSS nasıl uygulanır?.....	12
56. React'te animasyonlar nasıl oluşturulur?.....	12
57. React ve Redux arasındaki fark nedir?.....	13
58. React ve Angular arasındaki fark nedir?.....	13
59. React ve Vue arasındaki fark nedir?.....	13
60. React SSR (Server-Side Rendering) nedir ve neden kullanılır?.....	13
61. React Lazy Loading nedir ve nasıl yapılır?.....	13
62. React'te SSR ve CSR (Client-Side Rendering) arasındaki farklar nelerdir?.....	13
63. React'te key özelliği nedir ve neden önemlidir?.....	13
64. React'te HOC (Higher Order Component) nedir ve nasıl kullanılır?.....	13
65. React'te props drilling nedir ve nasıl önlenir?.....	14
66. React Hooks'ta useState ve useEffect nedir ve nasıl kullanılır?.....	14
67. React'te portallar (portals) nedir ve neden kullanılır?.....	14
68. React ve React Native arasındaki fark nedir?.....	14
69. React'te bir bileşenin state'ini başka bir bileşen nasıl yönetir?.....	14
70. React'te testing (test etme) nasıl yapılır?.....	14

FRONTEND SORULARI

1. HTML nedir ve ne işe yarar?

HTML (HyperText Markup Language), web sayfalarının yapılandırılmasını ve içeriğinin tanımlanmasını sağlayan bir işaretleme dilidir. Web tarayıcıları bu işaretleme dilini kullanarak web sayfalarını görüntüler.

2. HTML ve XHTML arasındaki farklar nelerdir?

HTML (HyperText Markup Language), daha esnek bir yapıya sahip olan ve hatalara daha hoşgörülü olan bir işaretleme dilidir. XHTML (eXtensible HyperText Markup Language), XML tabanlı bir işaretleme dilidir ve daha katı bir yapıya sahiptir, hatalara daha az hoşgörülüdür.

3. HTML5'in öne çıkan özellikleri nelerdir?

HTML5, video ve ses gibi multimedya içeriklerini destekler, yerel depolama yetenekleri sunar, daha semantik etiketler içerir (örneğin <header>, <footer>, <nav>), form geliştirmeleri sağlar ve daha iyi çizim ve grafik yetenekleri sunar.

4. HTML semantik elemanları nelerdir ve neden önemlidirler?

HTML semantik elemanlar, içeriklerinin anlamını tanımlamak için kullanılan etiketlerdir. Örneğin <header>, <footer>, <nav>, <article> gibi. Bu etiketler, web tarayıcılarına ve arama motorlarına sayfanın yapısını daha iyi anlamaları için rehberlik eder.

5. HTML validasyonu neden önemlidir ve bunu nasıl yaparsınız?

HTML validasyonu, web sayfalarının standartlara uygun olup olmadığını kontrol etmenin bir yoludur. Doğru yapılandırılmış bir HTML, tarayıcılar arasında tutarlılık ve erişilebilirlik sağlar. HTML validasyonunu, çeşitli çevrimiçi araçlar veya W3C'nin resmi doğrulama servisi aracılığıyla gerçekleştirebilirsiniz.

6. "DOCTYPE" nedir ve ne işe yarar?

<!DOCTYPE> belgesi, web tarayıcısına hangi HTML veya XHTML sürümünün kullanıldığını bildiren bir beyandır. Tarayıcı, bu bildirimi kullanarak sayfanın nasıl işleneceğini belirler.

7. "data-" öznitelikleri nedir ve ne amaçla kullanılır?

"data-" öznitelikleri, özel veri noktalarını HTML ve JavaScript arasında iletmek için kullanılan özel özniteliklerdir. Bu öznitelikler, özelleştirilmiş veri tutma ve işleme için kullanılabilir.

8. "alt" özniteliği nedir ve neden önemlidir?

"alt" özniteliği, bir görüntünün alternatif metnini tanımlar. Görüntü yüklenemediğinde veya kullanıcılar için görüntüleri engelleyen durumlarda, "alt" metni, kullanıcıya görüntünün içeriğini açıklar.

9. HTML form elemanlarını sıralayın ve her birinin amacını açıklayın.

Form elemanları arasında <input>, <textarea>, <select>, <button> gibi ögeler bulunur. Bunlar kullanıcı girişi almak veya kullanıcıyla etkileşimde bulunmak için kullanılır.

10. HTML ve CSS arasındaki ilişki nedir ve nasıl birlikte çalışırlar? kolay gelsin

HTML, web sayfalarının yapısını tanımlarken CSS (Cascading Style Sheets), bu yapının görüntülenme şeklini belirler. HTML, içeriği düzenlerken, CSS, bu içeriğin renkleri, boyutları, yerleşimi gibi görsel özelliklerini kontrol eder. Birlikte kullanıldıklarında, içeriğin sunumu ve stilini belirlemek için güçlü bir araç seti sunarlar.

11. CSS nedir ve ne işe yarar?

CSS (Cascading Style Sheets), web sayfalarının görünümünü düzenlemek için kullanılan bir stil dilidir. HTML'in yapısını belirlerken, CSS, bu yapının nasıl görüntüleneceğini tanımlar.

12. CSS seçicileri nelerdir ve nasıl kullanılır?

CSS seçicileri, HTML elementlerini ve öğelerini hedeflemek için kullanılan kod parçacıklarıdır. Örneğin, element adıyla (div, p), sınıf adıyla (.class), ID ile (#id), alt elementlerle (parent > child) ve daha fazlasıyla seçim yapabilirsiniz.

13. CSS'te inline, internal ve external stil sayfaları arasındaki farklar nelerdir?

Inline CSS, HTML elementinin içine yazılan stil tanımlarıdır. Internal CSS, <style> etiketi içinde HTML sayfasının <head> bölümünde tanımlanan stil tanımlarıdır. External CSS ise ayrı bir CSS dosyasında tanımlanan ve HTML sayfasına <link> etiketi ile bağlanan stil dosyasıdır.

14. CSS öncelikli kural nedir ve nasıl belirlenir?

CSS öncelikli kural, bir elemente uygulanacak olan birden fazla stil kuralı arasında hangisinin öncelikli olduğunu belirler. Öncelik, genellikle spesifiklik, önem ve stil sırasına göre belirlenir.

15. CSS önışlemcileri ve yaygın kullanılan örnekler nelerdir?

CSS önışlemcileri, CSS yazımını daha verimli ve etkili hale getiren araçlardır. Sass, Less, Stylus gibi önışlemci diller, CSS yazımını daha modüler, organize ve yeniden kullanılabilir hale getirir.

16. "box model" nedir ve CSS'te nasıl kullanılır?

Box model, bir HTML elementinin içeriğini, dolgusunu, kenarlığını ve dış boşluğunu belirler. Bu özellikler, width, height, padding, border, margin gibi CSS özellikleriyle kontrol edilir.

17. "Float" ve "clear" özellikleri ne işe yarar ve nasıl kullanılır?

Float, bir elementin diğer elementlerin etrafına sarılmasını sağlar. Clear, float edilmiş elementlerin altına geçecek yeni içeriklerin başlangıcını belirler.

18. CSS Grid ve Flexbox arasındaki farklar nelerdir?

CSS Grid, iki boyutlu düzenler oluşturmak için kullanılırken, Flexbox genellikle tek boyutlu düzenler oluşturmak için kullanılır. Grid, satırlar ve sütunlar arasında düzen sağlar, Flexbox ise içeriklerin bir eksen boyunca hizalanmasını sağlar.

19. Media queries nedir ve ne işe yarar?

Media queries, ekran boyutu, cihaz oryantasyonu, piksel yoğunluğu gibi çeşitli cihaz özelliklerine göre CSS'in nasıl uygulanacağını belirler. Bu, sayfaların farklı cihazlarda daha iyi görüntülenmesini sağlar.

20. CSS animasyonları nasıl oluşturulur ve uygulanır?

CSS animasyonları, @keyframes kuralıyla belirli bir zaman dilimi içinde elementlerin nasıl değişeceğini tanımlar. Bu animasyonlar, animation özelliğiyle elementlere uygulanır ve belirli bir süre boyunca çalıştırılır.

21. JavaScript nedir ve ne işe yarar?

JavaScript, web tarayıcılarında çalışabilen bir programlama dilidir. Genellikle web sayfalarının dinamik içeriğini yönetmek, kullanıcı etkileşimlerini sağlamak ve web uygulamalarını geliştirmek için kullanılır.

22. Değişkenler JavaScript'te nasıl tanımlanır ve kullanılır?

Değişkenler var, let, veya const anahtar kelimeleriyle tanımlanır. Değişkenler, değer ataması yapılarak veya boş olarak tanımlanabilir ve sonrasında bu değerler değiştirilebilir.

23. JavaScript'in veri tipleri nelerdir?

JavaScript'te temel veri tipleri şunlardır: string, number, boolean, null, undefined, object, ve symbol.

24. JavaScript'te koşullu ifadeler nasıl kullanılır?

if, else if ve else ifadeleriyle koşullu ifadeler belirlenir. Bu ifadeler, belirli koşulların karşılanması durumunda farklı kod bloklarının çalıştırılmasını sağlar.

25. Döngüler JavaScript'te nasıl kullanılır?

for, while, ve do-while gibi döngü yapıları kullanılarak JavaScript'te döngüler oluşturulur. Bu döngüler belirli koşullar sağlandığı sürece belirli kod bloklarının tekrarlanmasını sağlar.

26. JavaScript fonksiyonları nedir ve nasıl tanımlanır?

JavaScript fonksiyonları, belirli bir işlevi gerçekleştirmek için kullanılan kod bloklarıdır. Fonksiyonlar function anahtar kelimesiyle tanımlanır ve parametreler alabilirler.

27. JavaScript dizileri nasıl oluşturulur ve kullanılır?

JavaScript dizileri, köşeli parantez içinde virgülle ayrılmış öğeler listesi şeklinde tanımlanır. Diziler, indeks numarası kullanılarak elemanlara erişilir ve değiştirilebilirler.

28. Nesne (Object) nedir ve JavaScript'te nasıl oluşturulur?

Nesneler, özellik-değer çiftlerinden oluşan karmaşık veri yapılarıdır. JavaScript'te nesneler, süslü parantez içinde özellikler ve değerleriyle tanımlanır.

29. Prototipler ve prototype mirası nedir?

JavaScript'te, nesneler arasında prototip tabanlı bir miras mekanizması vardır. Bir nesne, diğer bir nesnenin özelliklerini ve metodlarını miras alabilir.

30. DOM (Document Object Model) nedir ve JavaScript'te nasıl kullanılır?

DOM, bir HTML veya XML belgesinin yapısını temsil eden bir programlama arayüzüdür. JavaScript kullanılarak, DOM API'si aracılığıyla HTML öğelerine erişilebilir ve bu öğeler üzerinde değişiklikler yapılabilir.

31. JavaScript'te olaylar (events) nedir ve nasıl işlenir?

Olaylar, kullanıcı etkileşimleri veya tarayıcıda meydana gelen diğer etkinliklerdir. JavaScript ile, HTML öğelerine atanmış olay dinleyicileri aracılığıyla olaylar işlenebilir.

32. JavaScript'te Asenkron Programlama nedir ve neden önemlidir?

JavaScript, tarayıcıda ve sunucuda asenkron programlama yapabilen bir dildir. Asenkron programlama, belirli işlemlerin diğer işlemlerin bitmesini beklemeden eşzamanlı olarak gerçekleştirilmesini sağlar, bu da daha hızlı ve etkili programlar oluşturulmasını sağlar.

33. JavaScript kapsamı (scope) nedir ve nasıl çalışır?

JavaScript kapsamı, bir değişkenin erişilebilirliğini ve görünürlüğünü belirleyen bir kavramdır. Global kapsam, fonksiyon kapsamı, ve blok kapsamı gibi farklı kapsam türleri vardır.

34. Closure (Kapanış) nedir ve nasıl çalışır?

Closure, bir iç fonksiyonun dış kapsamındaki değişkenlere erişmesine izin veren bir JavaScript özelliğidir. Closure, dış kapsamındaki değişkenlere erişimi kapanmış bir iç fonksiyonu içeren fonksiyonlardır.

35. Callback fonksiyonları nedir ve neden önemlidir?

Callback fonksiyonları, başka bir fonksiyona parametre olarak geçirilen ve o fonksiyonun tamamlandığında çağrılan fonksiyonlardır. JavaScript'te, asenkron işlemleri yönetmek ve kodun daha okunabilir ve modüler olmasını sağlamak için sıkça kullanılırlar.

36. Promise nedir ve nasıl kullanılır?

Promise, JavaScript'te asenkron programlamada kullanılan bir yapıdır. Bir işlemin tamamlanması veya başarısız olması durumunda gerçekleştirilecek işlemleri temsil eder.

37. Async/Await nedir ve nasıl kullanılır?

Async/Await, JavaScript'te Promise tabanlı asenkron kodu daha temiz ve okunabilir hale getirmek için kullanılan bir yapıdır. async işlevleri, asenkron işlemleri gerçekleştirmek için kullanılırken, await anahtar kelimesi, bir async işlevindeki bir Promise'in tamamlanmasını bekler.

38. Modül (Module) nedir ve JavaScript'te nasıl kullanılır?

JavaScript modülleri, kodu parçalara bölmek ve parçalar arasındaki bağımlılıkları yönetmek için kullanılır. ES6 modül sistemi, export ve import anahtar kelimeleriyle modüllerin tanımlanması ve kullanılmasını sağlar.

39. JavaScript'te hata yönetimi (error handling) nasıl yapılır?

JavaScript'te hata yönetimi, try, catch, finally ifadeleriyle ve throw anahtar kelimesiyle gerçekleştirilir. try bloğu içinde olası hata oluşturabilecek kodlar yazılır, olası hatalar catch bloğunda işlenir ve finally bloğu, hata olsun olmasın her zaman çalıştırılır.

40. JavaScript'te AJAX nedir ve nasıl kullanılır?

AJAX (Asenkron JavaScript ve XML), web sayfalarının arka planda sunucu ile iletişim kurmasını ve sayfa yeniden yüklenmeden içerikleri güncellemesini sağlayan bir tekniktir. JavaScript ile, XMLHttpRequest veya fetch API'si kullanılarak AJAX istekleri gönderilir ve alınan veriler sayfa içinde dinamik olarak kullanılabilir.

41. React nedir ve ne işe yarar?

React, kullanıcı arayüzü oluşturmak için kullanılan bir JavaScript kütüphanesidir. Tek sayfa uygulamaları geliştirmek için kullanılır.

42. React'in temel özellikleri nelerdir?

React, bileşen tabanlı bir yapıya sahiptir, sanal DOM kullanır, tek yönlü veri akışı prensibini benimser ve yeniden kullanılabilir bileşenler sağlar.

43. React bileşenleri nedir ve nasıl oluşturulur?

React bileşenleri, UI'nin farklı parçalarını temsil eder. Sınıf bileşenleri veya fonksiyonel bileşenler olarak oluşturulabilirler.

44. JSX nedir ve React'te neden kullanılır?

JSX, JavaScript ve XML'in bir kombinasyonudur ve React bileşenlerinin UI'sini tanımlamak için kullanılır. JSX, daha okunabilir ve yazılması kolay kod yazmayı sağlar.

45. React'te state ve props arasındaki fark nedir?

State, bir bileşenin iç durumunu tutar ve bileşen içinde değiştirilebilir. Props, bir bileşene dışarıdan geçirilen verilerdir ve değiştirilemezler.

46. React yaşam döngüsü (lifecycle) nedir ve nasıl çalışır?

React bileşenleri, monte edilme (mounting), güncellenme (updating), ve boşaltılma (unmounting) olmak üzere üç aşamalı bir yaşam döngüsüne sahiptir. Bu döngü boyunca belirli metotlar çağrılır.

47. React Router nedir ve ne işe yarar?

React Router, tek sayfa uygulamalarında sayfa yönlendirmesi ve routing işlemlerini kolaylaştıran bir React kütüphanesidir.

48. React'te event handling nasıl yapılır?

React'te event handling, JSX içinde olay işleyicileri atanarak veya sınıf bileşenlerinde metodlar aracılığıyla gerçekleştirilir.

49. React Context API nedir ve ne işe yarar?

React Context API, bileşenler arasında veri paylaşımını kolaylaştıran bir React API'sidir. Veri, bileşenler hiyerarşisi boyunca geçirilirken ara bileşenlerden geçirme zorunluluğunu azaltır.

50. React Hooks nedir ve neden kullanılır?

React Hooks, fonksiyonel bileşenlerde state ve yaşam döngüsü özelliklerini kullanabilmemizi sağlayan ve sınıf bileşenlerinden kurtulan React özellikleridir.

51. React'te bir bileşenden diğerine veri aktarımı nasıl yapılır?

Props kullanılarak bir bileşenden diğerine veri aktarımı yapılır. Eğer component'ler aynı parent içindeyse context api de kullanılabilir.

52. React'te conditional rendering nasıl yapılır?

JSX içinde if-else kullanılamaz, bunun yerine koşullu ifadeler veya koşullu operatörler kullanılır.

53. React'te listeler nasıl oluşturulur ve döngüler nasıl kullanılır?

Listeler, .map() metodu kullanılarak oluşturulur. Bu metot, bir diziyi döngü içinde dönerek her eleman için yeni bir bileşen oluşturmanızı sağlar.

54. React'te form işlemleri nasıl yapılır?

Form işlemleri, kontrol bileşenleri ve olay işleyicileri kullanılarak gerçekleştirilir. Form verisi genellikle state içinde tutulur ve bir olay tetiklendiğinde güncellenir.

55. React'te CSS nasıl uygulanır?

CSS, .css dosyalarıyla ayrı bir dosyada tutulabilir veya styled-components gibi kütüphaneler kullanılarak bileşenlerle birlikte yazılabilir.

56. React'te animasyonlar nasıl oluşturulur?

React'te animasyonlar genellikle CSS ile yapılır. Ancak react-spring veya framer-motion gibi kütüphaneler de kullanılabilir.

57. React ve Redux arasındaki fark nedir?

Redux, React uygulamalarında durum yönetimi için kullanılan bir JavaScript kütüphanesidir. Redux, uygulamanın genel durumunu merkezi bir yerde yönetirken, React sadece UI bileşenlerini temsil eder.

58. React ve Angular arasındaki fark nedir?

React, bir JavaScript kütüphanesi iken, Angular bir JavaScript framework'üdür. React, bileşen tabanlı bir yaklaşım benimserken, Angular tam bir MVC framework'üdür ve daha kapsamlı bir yapı sunar.

59. React ve Vue arasındaki fark nedir?

React ve Vue ikisi de modern JavaScript framework'leridir. React, Facebook tarafından geliştirilmiştir ve büyük bir topluluğa sahiptir. Vue ise daha küçük bir ekip tarafından geliştirilmiştir ve daha basit bir yapı sunar.

60. React SSR (Server-Side Rendering) nedir ve neden kullanılır?

React SSR, React uygulamalarının sunucu tarafında oluşturularak kullanıcılara sunulması anlamına gelir. Bu, SEO iyileştirmeleri, daha iyi performans ve daha iyi kullanıcı deneyimi sağlar.

61. React Lazy Loading nedir ve nasıl yapılır?

React Lazy Loading, yalnızca kullanıcının ihtiyaç duyduğu bileşenleri yüklemeyi sağlayan bir tekniktir. Bunun için React.lazy() fonksiyonu ve Suspense bileşeni kullanılır.

62. React'te SSR ve CSR (Client-Side Rendering) arasındaki farklar nelerdir?

SSR'da, sunucu ilk sayfa yüklemesi sırasında HTML içeriğini oluştururken CSR'da, sunucu sadece başlangıç HTML'ini gönderir ve ardından istemci (tarayıcı) JavaScript tarafından içeriğin oluşturulması gerçekleştirilir.

63. React'te key özelliği nedir ve neden önemlidir?

Key özelliği, React'te yeniden render edilen öğeleri birbirinden ayırmak için kullanılır. Bu, React'in öğelerin değişikliklerini izlemesine ve daha verimli bir şekilde güncelleme yapmasına olanak tanır.

64. React'te HOC (Higher Order Component) nedir ve nasıl kullanılır?

HOC, bir veya birden fazla bileşeni alarak yeni bir bileşen döndüren bir fonksiyondur. Bu, bileşenleri yeniden kullanarak ve kod tekrarını önleyerek işlevselliği genişletmenin bir yoludur.

65. React'te props drilling nedir ve nasıl önlenir?

Props drilling, bir bileşenden diğerine props'ların geçirilmesi gerektiğinde oluşan bir durumdur. Bu durum, verimliliği azaltabilir. Bu durumu önlemek için, context API veya Redux gibi durum yönetimi araçları kullanılabilir.

66. React Hooks'ta useState ve useEffect nedir ve nasıl kullanılır?

useState, fonksiyonel bileşenlerde state yönetimini sağlar. useEffect, bileşenin yaşam döngüsü metotlarının işlevselliğini yerine getirir ve belirli bir yan etkiyi tetikler.

67. React'te portallar (portals) nedir ve neden kullanılır?

Portallar, bir bileşeni DOM hiyerarşisi dışındaki bir yere monte etmek için kullanılır. Örneğin, bir modal penceresini ana uygulama ağacı dışında oluşturmak için kullanılabilir.

68. React ve React Native arasındaki fark nedir?

React, web uygulamaları oluşturmak için kullanılırken, React Native, iOS ve Android platformları için mobil uygulamalar oluşturmak için kullanılır. React Native, JavaScript ve React'in gücünü kullanarak native mobil uygulamalar oluşturur.

69. React'te bir bileşenin state'ini başka bir bileşen nasıl yönetir?

Bileşenler arasında state yönetimi için genellikle veri akışını yukarı doğru (yani, parent bileşene) taşıma (lifting state up) veya context API kullanma yaklaşımları tercih edilir.

70. React'te testing (test etme) nasıl yapılır?

React bileşenlerini test etmek için genellikle Jest ve React Testing Library gibi test kütüphaneleri kullanılır. Bu kütüphaneler, bileşenlerin davranışlarını ve görünümlerini test etmek için birçok araç ve fonksiyon sağlar.