

CSS CEVAPLAR

Cevap 1

Css bir web sayfasına html taglerinin style attribute'u ile ya da style etiketi arasına yazılacak css ifadesi ile ya da .css uzantılı dosyada yazılacak css ifadeleri olmak üzere üç farklı şekilde uygulanabilir.

Cevap 2

Specificity: Uygulanan Css ifadelerinden hangisinin baskın olacağını hangilerinin override edileceğini ifade eder. Örneğin aynı h1 html elementine color özelliği, class özelliği ile green, id özelliği ile black, style özelliği ile yellow atanırsa style attribute en spesifik tanımlama olduğundan id ve class ifadelerindeki color override edilir ve color yellow olur. Id scope'nun içindeki ifadeler de, class scope'unun içindeki aynı ifadeleri override eder. Örneğin aynı örnekte id ve class scope'unda fon-size ifadesi tanımlanmış ancak style attribute'nde tanımlanmamış olsun bu durumda id scope'unda tanımlanan font-size değeri geçerli olacak class scope'undaki fon-size tanımlaması override edilecektir.

JS CEVAPLAR

Cevap 1

Değişkenler ikiye Primitive Type ve Reference Type olmak üzere ikiye ayrılır. Primitive Type değişkenler başka bir değişkene atandığı zaman kopyalanırlar dolayısıyla yeni değişkenin değerinin değişmesi diğer değişkenleri etkilemez. Ramde ayrı yer tutarlar. Reference Type değişkenler, Primitive Type değişkenlerin tam tersi şeklinde çalışırlar. Örneğin person adında bir değişken olsun ve bu değişkenin veri tipi Object olsun person2 = person yordamıyla person2 adında bir değişken oluşturup bu değişken üzerinde işlemler yapılırsa person değişkeni de etkilenir çünkü ikisi de Ramde aynı yeri kaplıyor. person2 = person ifadesiyle değişkenin içeriği değil, ramdeki adres kopyalanır.

Primitive Type değişkenlerin bazı veri tipleri şunlardır: String, Number, Symbol

Reference Type değişkenlerin bazı veri tipleri şunlardır: Object, Array, Function

Cevap 2

Javascriptte birçok değişken veri tipi vardır. Bazıları şunlardır: Number, Object, Array, String, Symbol, Function

Cevap 3

Hoisting: Javascript'te var keyword'u ile tanımlanan değişkenler ve fonksiyonlar(anonymous function olmamak koşuluyla) program çalışmadan önce taranıp hafızaya alınırlar. Daha sonra kodlar yukarıdan aşağıya doğru çalıştırılır. Bundan dolayıdır ki bir fonksiyonu 20 satırda tanımlanmasına rağmen daha üst satırlarda çağrılıp kullanılabilir.

Scope: Değişkenleri hangi alanda çalışabileceğini ifade eder. Örneğin fonksiyon içinde tanımlanan değişkenler tanımlanan fonksiyonun dışından erişilemezler. Bu da değişkenlerin birbirine karışması programın daha güvenli çalışmasına olanak tanır.

Closure: Tanımlanan değişkenler kendi scope'u sınırlarındaki tüm alt scope'larda da erişilebilir olmasına closure denilir. Örneğin bir fonksiyonda for scope dışında tanımlanan bir değişkenin for scope'unda kullanılabilmesi ya da bu fonksiyonun içinde tanımlanan başka bir fonksiyon varsa üst fonksiyonda tanımlanan değişkenin alt fonksiyonda kullanılabilmesi.

Cevap 4

Strict equality(===): Bu operator iki değişkenin hem değerini hem de veri tipini karşılaştırarak geriye bool veri tipinde bir değer(true ya da false) döndürür.

Loose equality(==): Bu operator iki değişkenin veri tipine kontrol etmeden sadece değerine bakarak karşılaştırma yaparak geriye bool veri tipinde değer döndürür.

Cevap 5

This anahtar sözcüğü Javascriptte duruma göre farklı objelere işaret edebilir. Örneğin bir sınıfın içinde this anahtar sözcüğü sınıftan türeyecek instanceye işaret edebilir, ya da bir Constructor fonksiyonunda yine aynı şekilde instance işaret eder. Bir objenin fonksiyonunda(arrow function olmamak koşuluyla) o objeye işaret eder. This sözcüğü herhangi bir objeye ait olmayan bir fonksiyonun içinde global this işaret eder. Global this window olabilir.

Cevap 6

Matematiksel ifadede önce çarpma işlemi yapılır $2 \times 101 = 202$

Daha sonra 202 sayısını(veri tipi Number), "1" string ile toplama işlemi iki ifadeyi birleştirilir. Yani aslında 202 stringe dönüştürülür ve "1" ile birleştirilip veri tipi string olan 2021 değeri alınır.

Cevap 7

Dom yani Document Object Model Tarayıcılarda gördüğümüz arayüzün obje olarak ifade edilmesidir. Bir ağaç yapısındadır.

TS CEVAPLAR

Cevap 1

Javascriptin dinamik bir dil olduğundan yazılım geliştirme sırasında veri tiplerinden kaynaklı hatalar yazılım ancak çalıştırılınca program tarafından geliştiriciye gösterildiği için büyük projelerde bu hataları ayıklamak çok zahmetli ve kimi zamanda saatler süren bir hal alabiliyor. Bundan dolayı Microsoft 2012 yılında static tipli veri tipi hatalarını geliştirme sırasında geliştiriciye gösteren Javascript üzerine kurulu Typescript adında yeni bir dil geliştirdi. Ancak Typescript tamamen Javascript üzerine kuruludur. Typescript kodları çalıştırılmadan önce Javascript koduna çevrilir ve javascript kodları çalıştırılır aslında.

Cevap 2

Interface, bir fonksiyonun çok fazla parametre alması durumunda bu parametreleri her seferinde sırasıyla fonksiyona göndermek yerine tüm bu değişkenlerin bir paket içerisinde sıralamasının önemli olmadığı bir şekilde göndermeyi amaçlar.

REACT CEVAPLAR

Cevap 1

React, sayfada yapılan küçük bir değişiklikle tüm sayfayı yenilemek yerine sadece değişen componenti güncellemek için Virtual Dom'dan yararlandı. Virtual Dom tıpkı Dom gibi html dokümanını saklar. Herhangi bir değişiklik olduğunda yeni bir Virtual Dom oluşturulur ve eskisiyle karşılaştırılır. Değişen kısımlar saptanır ve Dom'da da bu değişim yapılır ve eski Virtual Dom silinir, yeni Virtual Dom artık Virtual Dom olur. Bu sayede karşılaştırılacak Virtual Dom da her değişiklik sonrası güncel tutulmuş olur.

Cevap 2

Controlled Component: Controlled Component her veri girişinde state'i güncellenen ve dom'u update eden componentlerdir. Örneğin bir formda email input'una her tuşa tıklandığında component güncellenir. Bu da istenmeyen karakterlerin girildiği anda fark edilmesini ve uyarı vermeyi sağlar.

Uncontrolled Component: State'i olmayan componentlardır. Inputa girilen veri değiştiğinde component dom tarafında güncellenir. İstenmeyen karakterlerin olduğu durumlarda kullanılması tavsiye edilmez.

Cevap 3

Stateful Component'ler bir state olan, state değiştiğinde dom'u güncellenmesini sağlayan componentlardır.

Stateless Component'lar state sahip olmayan birden fazla componenti içinde barındıran containerlardır.

Cevap 4

Bir işlem birden fazla component'te yapılmak istendiğinde high order components kullanılır. Tıpkı high order function gibi. Örneğin Javascript arraylerinde map fonksiyonu bir high order fonksiyondur. Parametre olarak bir fonksiyon alır ve dizinin tüm elemanlarına bu işlemi uygular. High order component'lerde de aynı işlemin birden fazla farklı component'te yapılmasını sağlar. Burada dikkat edilmesi gereken değişken değişkenin bir component olduğudur. Component'i değişken olarak alıp bir işlemi icra eden component'lara high order component denir.

Avantajları farklı componentlerde tekrarlanacak bir işlemin bir component'te yapılması ve geriye o işlem yapılmış aynı component döndürülmesidir. Örneğin bir sayfa yüklenirken ekrana loding yazısı yazılması gereken bir durum olsun. Oluşturulan her component'te durumu kontrol edip loading yazmak yerine tarayıcıda yüklenen component değişkenlik gösterdiği için withLoading adında high order component oluşturulur. Bu component parametre olarak bir component alır. Örneğin BookList, Book componentler oluşturduğumuzu düşünelim. WithLaoding high order component'e parametre olarak Book gönderdiğimizde geriye WithLoadingBook ya da BookList gönderdiğimizde WithLoadingBookList componentlerini elde ederiz. Gereksiz kod yazma zahmetinden kaçınılmış olunur.