**Interview questions**

1. Semantic elements

Semantik elementlər həm brauzer, həm də developer üçün mənası olan kodlar deməkdir. Semantik elementlərdən ona görə istifadə edirik ki, easy to read, accessibility(disability olan insanlar üçün nəzərdə tutulub) və SEO(search engine optimization). //article, aside,h1,form,header,nav,section

Screen readers veb səhifələri emal etmək və görmə qabiliyyəti zəif ola istifadəçilərə onları naviqasiya etmələrinə kömək edir. Semantik teqlər istifadəçilərə hal hazırda harda olduqlarını və səhifə ilə necə interact edə biləcəklərini göstərir.

1. SEO nədir?

SEO axtarış zamanı veb saytın daha öndə olmağına kömək edir. Search Engine səhifədən səhifəyə keçidləri izləyərək interneti tarayır və tapılan məzmunu indeksləşdirir. Axtardığınız zaman search engine indekslənmiş məzmunu göstərir.

1. Var, let, const fərqi

Varla təyin olunanlar ya global ya da functional scopedur, lakin let və const block scopedur. var 1 neçə dəfə dəyişəni declare etmək mümkündür, lakin const və let-də yalnız 1 dəfə declare etmək olur. Let və const-a birinci dəyər verilməlidir, sonra consola yazılmalıdır, lakin varda dəyər mənimsətmədən öncə consola yazmaq mümkündür.

1. Hoisting

Function declaration-da 1ci funksiyanı çağırıb sonra yaza bilərik. Bu da hoisting adlanır. Arrow və expression function hoisting olmur. //Funksiya müəyyən taskı yerinə yetirmək üçün yenidən istifadə olunan kod blokudur.

Dəyər tanıtmadan öncə çağıranda varla edəndə undefined alırıq, let və const-la edəndə tdz-ə düşür, referance error alırıq.

1. Closure nədir?

Bizdə iç-içə 2 funksiya olur, əgər inner functionun outer functiona-a accesi varsa bu closure sayılır.

1. DOM

Veb səhifə yüklənəndə brauzer səhifəfinin DOM-nu(Document Object Model) yaradır. Bu bizim veb səhifəmizi təmsil edir. DOM metodları vasitəsilə HTML elementləri üzərində actions edə bilərsən, propertilərlə value-ları dəyişə və ya set edə bilərsən. DOM-da Html elemetləri obyekt kimi müəyyən olur.

1. React component

React Uİ komponentlərini qurmaq üçün istifadə olunan JS kitabxanasıdır. Components müstəqil və reusable kodlardır. Html elementlərini return edən function kimi baxmaq olar. 2 cür olur: Function və Class component

Component böyük hərflə başlamalıdı. Class component mütləq extends React.Component statementi olmalıdır. Components render() metodunu tələb edir. //React single page app-dir

1. Virtual DOM

Bizdə real və virtual dom vardır. Real dom bizim əsas səhifəmiz, virtual dom isə onun kopyasıdır. Virtual dom-a JS tərəfdə Uİ-ı özündə saxlayan obyekt kimi baxmaq olar. Biz real dom hər hansısa bir hissəni dəyişəndə virtual dom diffing alqoritmi ilə müqayisə edir ki, hansı hissə dəyişilibsə yalnız o hissəni update etsin. Bütöv səhifəni yeniləməsin, bu proses reconciliation proses adlanır. real domdan daha sürətli işləyir.

1. Reactda listləri render edəndə niyə keylərdən istifadə edirik?

Keylər unique olmalıdır. Key-lər kömək edir ki React müəyyən etsin ki, hansı item dəyişib, əlavə olunub və ya silinir. Key-lər elementlərə arrayin içində sabit id-lərlə verilməlidir.

1. Hooks

Hooklar js obyekti sayılır. Class yazmadan state və reactin digər özəlliklərindən istifadə etməyə şərait yaradır. Əsas hooklar: useState, UseEffect, useContext-dir.

UseState – static və dəyişkən datalarımız mövcuddur. Əgər hər hansısa datamızda dəyişiklik etmək istəyirik ki, UseState istifadə edirik.

Hooklar xüsusi functionlardır, bununla state-ləri lifecycle metodları idarə edə bilirik. UseState stateləri idarə etmək üçündür. Arrayin içinə 2 var gəlir, biri statein özüdür, digəri isə state-i set etmək üçün functiondur(setState). 1 dəyər qəbul edir bu da initial dəyəri olur.

UseEffect lifecycle-i idarə etmək üçün olan hookdur. Callback function, və defendency array qəbul edir. Array-i boş saxlayanda saytı açanda usecallback işə düşür. Array-in içində dəyər varsa onlar və usecallbackin içi göndərilir.

UseCallback bizim functionlarımızı memorize etmək üçün lazım olan hookdur. Function props dəyşəndə, state dəyişəndə, parent element dəyişəndə render olunur. Hər dəfə render olanda function yenidən çağırılır, bunun qarşısını almaq üçün usecallback istifadə olunur. Hər dəfə arxa tərəfdə yeni function yaradılmır.

1. Class component və function component fərqi

Class-da hooklar yoxdur, functionda var. Classda lifecycle metodlar var, functionda yoxdur və bunu callbacklə idarə edirik. Class-da error handling etmək olur, functionda olmur.

1. UseMemo və UseCallback fərqi

Hər ikisi hooklardır. Dəyəri memorize etmək üçün istifadə olunur. Hər ikisi arqument alır və funksiyanın memorize olunmuş versiyonunu qaytarır. Usememo run olunmuş functionun nəticəsi olan yaddaşda saxlanılan dəyəri qaytarır, lakin usecallback yalnız hər hansı bir dependenci dəyişirsə return edir.

useCallback and useMemo both expect a function and an array of dependencies. The difference is that useCallback returns its function when the dependencies change while useMemo calls its function and returns the result.

1. Lifecycle

3 hissədən ibarətdir: mount, update, unmount. Mount component yaranıb render olanda baş verir. Update update olanda(state və ya propsu dəyişəndə) baş verir. Unmount bəzi şeyləri resetləyəndə baş verir.

1. Defer

Script teqini body-nin axrında yazırıqki, əvvəlcə html-i oxuyub load etsin daha sonra scripti işə salsın. Head-də yazmaq istəsək deferdən istifadə edirik. Defer scripti oxuyur amma işə salmır, html-ləri yükləyəndəndən sonra execution edir.

1. ÜMUMİ

Block line elements- div, semantic,br- height, vidth, margin

İnline-block-span, a

İnline- vidth height qəbul etməyəcək. Margin, top, bottom. Contentin tutduğu yeri tutur.

Flex, Grid- 2dimension / vertical, horizontal- 1 dimension

Display:none – elementi silir, tutduğu yeri saxlamır

Visibility:none – elementin tutduğu space-i saxlayır.

<main> -təkrarlana bilməz.

<div>, <span> - semantik element deyil.

<meta> - Web page mobile, view port, description, author, utf-8(content)

Position static- default top bottom right left götürmür

Position relative götürür.

Array-lərin itemlarını unique etmək, set-lə destruct edib göstəririk.

! importance

İnline-style - <style>

Pseudo class linked, visited,active, hover

Pseudo elements- before, after

Animation keyframe: 0%/100%/from-to

Tranfor: translate(scale, rotate)

Box-model: content, padding(contentlə border arasındakı məsafə), border, margin(borderlə digər elementlər arasındakı məsafə)

Box-sizing: content-box, border-box

JS 2 data type(stack-də tutduğu yer) var: primitive və nonprimitive.

Primitive: string, number, Boolean, null, undefined, simvol, bigİnt

Non-primitive: array, object, map, set- value-lar heapdə saxıanılır, reference type-lar stackdə, pointerler stackde

Class declaration-birbaşa,constructor kimi də istifadə edə bilərik.

Class expression- value-a assign olunur.

Arrov function fərqi: hoisting olunmur, enclosed scope-dur(this binding yoxdur), arguments olmur, super-I yoxdur, nev keyvord istifadə olunur.