**Veb İnkişaf – Frontend İmtahan Sualları**

**Nəzəri suallar**

**1. Frontend inkişafı nədir və müasir veb dizaynda niyə vacibdir?**Frontend inkişafı veb sayt və ya tətbiqin istifadəçi ilə birbaşa qarşılıqlı əlaqədə olan hissəsinin hazırlanmasıdır. Buraya saytın görünüşü (dizayn), istifadəçi interfeysi (UI), interaktivlik və istifadəçi təcrübəsi (UX) daxildir. Frontend inkişafı HTML, CSS və JavaScript kimi texnologiyalardan istifadə edir. Müasir veb dizaynda vacibdir, çünki istifadəçilər saytın sürətli, estetik və istifadəsi asan olmasını gözləyir. Yaxşı frontend istifadəçilərin saytda daha çox vaxt keçirməsinə və məmnun qalmasına səbəb olur.

**2. HTML nədir? Veb inkişafda hansı rolu oynayır?**HTML (HyperText Markup Language) veb səhifələrin əsas strukturunu yaradan işarələmə dilidir. O, mətn, şəkil, video, link, cədvəl və digər elementlərin səhifədə necə yerləşəcəyini və bir-biri ilə necə əlaqədə olacağını müəyyən edir. HTML olmadan veb səhifə yaratmaq mümkün deyil, çünki bütün məzmun və struktur HTML vasitəsilə qurulur. CSS və JavaScript isə bu strukturun üzərində işləyir.

**3. <div> və <section> elementləri arasındakı fərq nədir?**<div> qeyri-semantik konteynerdir, yəni heç bir məna daşımır, sadəcə elementləri qruplaşdırmaq üçün istifadə olunur. Məsələn, bir neçə elementi bir yerdə saxlamaq və ya onlara ümumi stil vermək üçün.<section> isə semantik elementdir və səhifənin müəyyən bir bölməsini təmsil edir. Məsələn, məqalənin bir hissəsi, xidmətlər bölməsi və s. <section> istifadə etmək səhifənin strukturunu aydınlaşdırır və SEO üçün faydalıdır.

**4. Daxili və xarici CSS arasındakı fərq nədir?**

* **Daxili CSS**: HTML faylının <head> hissəsində <style> tagı içində yazılır. Kiçik layihələr və ya bir neçə element üçün uyğundur.
* **Xarici CSS**: Ayrı .css faylında yazılır və HTML-ə <link rel="stylesheet" href="style.css"> ilə əlavə olunur. Böyük layihələrdə kodun təmiz və təkrar istifadəsini asanlaşdırır, bir neçə səhifədə eyni stili tətbiq etməyə imkan verir.

**5. Responsiv dizayn nədir? Real həyatdan bir nümunə göstərin.**Responsiv dizayn veb saytın müxtəlif cihazlarda (telefon, planşet, kompüter) düzgün və rahat görünməsini təmin edən yanaşmadır. Bu, media sorğuları və çevik şablonlar vasitəsilə həyata keçirilir.**Nümunə:** Bir onlayn mağaza saytına kompüterdən baxanda məhsullar 4 sütunda, telefondan baxanda isə 1 sütunda düzülür. Məsələn, Amazon-un veb saytı.

**6. DOM nədir və JavaScript onunla necə işləyir?**DOM (Document Object Model) HTML sənədinin proqramlaşdırıla bilən obyekt modelidir. DOM-da hər bir HTML elementi bir obyekt kimi təmsil olunur və JavaScript bu obyektlərə daxil olaraq onları dəyişə, əlavə və ya silə bilər.Məsələn, bir düyməyə klik edəndə JavaScript həmin düymənin rəngini dəyişə bilər.

**7. HTML və CSS-də class istifadəsinin məqsədi nədir?**Class-lar elementləri qruplaşdırmaq və onlara xüsusi stil və ya funksionallıq vermək üçün istifadə olunur. Eyni class bir neçə elementə tətbiq oluna bilər.Məsələn, bütün "button" class-ına sahib düymələrə eyni rəng və ölçü vermək mümkündür.

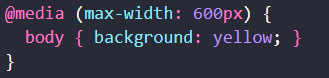
**8. Git və GitHub frontend inkişafında necə kömək edir?**Git kodun versiya nəzarətini təmin edir, yəni kodda edilən bütün dəyişiklikləri izləməyə, lazım olduqda əvvəlki versiyalara qayıtmağa imkan verir.GitHub isə kodun buludda saxlanılması, komanda ilə birgə işləmək, kodun paylaşılması və layihənin idarə olunmasını asanlaşdırır. Məsələn, bir neçə frontend proqramçı eyni layihədə paralel işləyə bilər.

**9. Flexbox və CSS Grid arasında yerləşdirmə baxımından fərqləri izah edin.**

* **Flexbox**: Bir ölçülü yerləşdirmə üçündür (ya sıra, ya da sütun). Elementləri asanlıqla düzülüşdürmək, aralarındakı məsafəni tənzimləmək üçün idealdır.
* **CSS Grid**: İki ölçülü yerləşdirmə üçündür (həm sıra, həm də sütun). Daha mürəkkəb layout-lar yaratmaq üçün istifadə olunur.

Məsələn, Flexbox ilə naviqasiya paneli, Grid ilə isə tam səhifə layout-u qurmaq olar.

**10. Media sorğuları responsiv dizaynda necə istifadə olunur?**Media sorğuları (@media) CSS-də müxtəlif ekran ölçülərinə uyğun fərqli stillər tətbiq etməyə imkan verir.Məsələn:



Bu kod 600px-dən kiçik ekranlarda fon rəngini dəyişir.

**11. Frontend üçün inkişaf mühitini necə qurursunuz?**

* Kod redaktoru (məsələn, VS Code) quraşdırılır.
* Brauzer (Chrome, Firefox və s.) istifadə olunur.
* Node.js və npm quraşdırılır (lazım olan paketləri idarə etmək üçün).
* Git və GitHub hesabı yaradılır.
* Layihə üçün fayl strukturu qurulur, lazım olan framework-lər (React, Vue və s.) əlavə olunur.
* Layihə lokal serverdə işlədilir və test olunur.

**12. Semantik HTML elementləri nədir? Onları işlətməyin üstünlüyü nədir?**Semantik elementlər səhifənin strukturunu və məzmununu daha aydın göstərən HTML elementləridir: <header>, <nav>, <main>, <footer>, <article>, <section>, <aside>.Üstünlükləri:

* SEO üçün faydalıdır (axtarış motorları səhifəni daha yaxşı başa düşür).
* Əlçatanlıq artır (oxuyucu proqramlar səhifəni düzgün oxuyur).
* Kodun oxunması və saxlanılması asanlaşır.

**13. JavaScript istifadə edərək istifadəçi təcrübəsini necə yaxşılaşdırmaq olar?**

* Formların yoxlanılması (validation) səhifə yenilənmədən.
* Dinamik məzmunun yüklənməsi (AJAX, fetch).
* Animasiya və keçidlər.
* Modal pəncərələr, açılan menyular, sürüşdürülə bilən qalereyalar.
* Real vaxtda axtarış və filtrasiya.

Bunlar istifadəçinin saytda rahat və sürətli işləməsini təmin edir.

**14. CSS-də pseudo-class nədir? İki nümunə və onların istifadəsini göstərin.**Pseudo-class-lar elementin xüsusi vəziyyətini seçmək üçün istifadə olunur.Nümunələr:

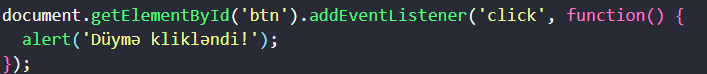
* :hover — elementin üzərinə mouse gələndə:



* :focus — input sahəsi seçiləndə:



**15. JavaScript-də hadisələr nədir? Event listener necə işləyir?**Hadisələr istifadəçi və ya brauzer tərəfindən baş verən aksiyalardır (klik, klaviatura basılması, səhifənin yüklənməsi və s.).Event listener bu hadisələri izləyir və baş verəndə müəyyən funksiyanı işə salır.Məsələn:

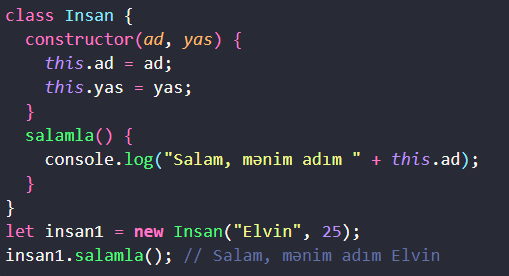


**16. position: absolute; xüsusiyyətinin hansı hallarda istifadə olunduğunu izah edin.**Bu xüsusiyyət elementin valideyninə və ya səhifəyə nisbətən tam dəqiq yerləşdirilməsi üçün istifadə olunur.Məsələn, bir şəkli səhifənin sağ yuxarı küncündə göstərmək və ya bir düyməni müəyyən bir yerdə sabit saxlamaq üçün.

**17. Frontend kodun performansını necə optimallaşdırmaq olar?**

* Şəkilləri sıxmaq və ölçüsünü azaltmaq.
* CSS və JavaScript fayllarını minify etmək.
* Kodun təkrar hissələrini aradan qaldırmaq.
* Lazy loading tətbiq etmək (lazım olmayan məzmunu sonradan yükləmək).
* CDN istifadə etmək.
* Brauzer keşindən istifadə etmək.
* Asinxron script yükləmək.

**18. JavaScript-də obyekt-yönümlü proqramlaşdırma anlayışını nümunə ilə izah edin.**OOP (Obyekt-Yönümlü Proqramlaşdırma) obyektlər və onların xüsusiyyətləri/metodları üzərində qurulur.Nümunə:



**19. Veb sayt dizaynında əlçatanlıq nədir və niyə vacibdir?**Əlçatanlıq (accessibility) veb saytın fiziki məhdudiyyətli insanlar üçün də istifadəyə yararlı olmasıdır.Vacibliyi:

* Daha geniş auditoriyaya çatmaq.
* Qanunlara uyğun olmaq (bəzən məcburidir).
* İstifadəçi təcrübəsini artırmaq.

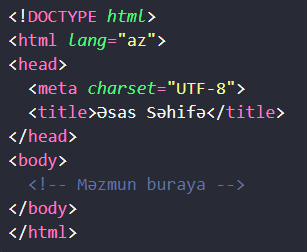
Məsələn, şəkillərə alt atributu əlavə etmək, klaviatura ilə idarə olunan menyular yaratmaq.

**20. Sinxron və asinxron JavaScript fərqlərini müqayisə edin.**

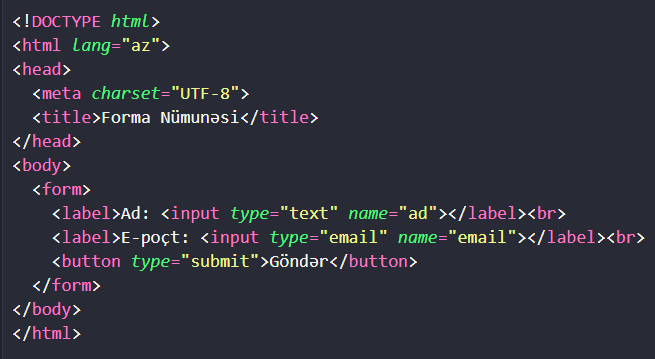
* **Sinxron**: Kod ardıcıl işləyir, hər əməliyyat bitmədən növbəti başlamır. Məsələn, böyük bir dövrədə kod işləyirsə, səhifə donur.
* **Asinxron**: Kod paralel işləyə bilər, yəni bir əməliyyat gözlənilərkən digərləri davam edir. Məsələn, setTimeout, fetch, Promise, async/await. Bu, istifadəçi təcrübəsini yaxşılaşdırır, çünki səhifə donmur və digər əməliyyatlar davam edir.

**B hissəsi: Praktiki kod əsaslı suallar**

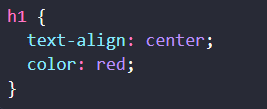
1. **Bir veb səhifə üçün əsas HTML skeletini (boilerplate) yazın. (asan)**

****

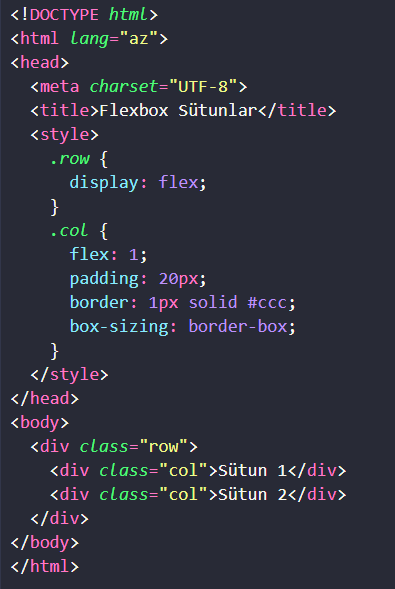
1. **Ad, e-poçt və göndər düyməsi olan HTML forma yaradın. (asan)**

****

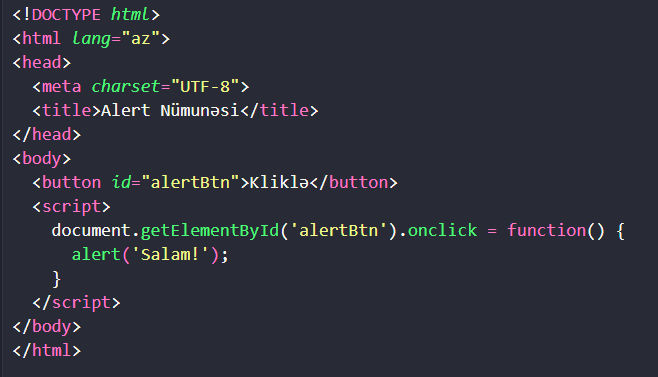
1. **Bütün <h1> mətnlərini mərkəzləşdirilmiş və qırmızı rəngə çevirən CSS kodu yazın. (asan)**

****

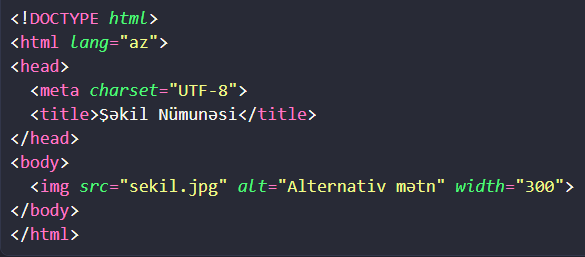
1. **Flexbox istifadə edərək eyni enə malik iki sütunu bir sırada düzün. (asan)**

****

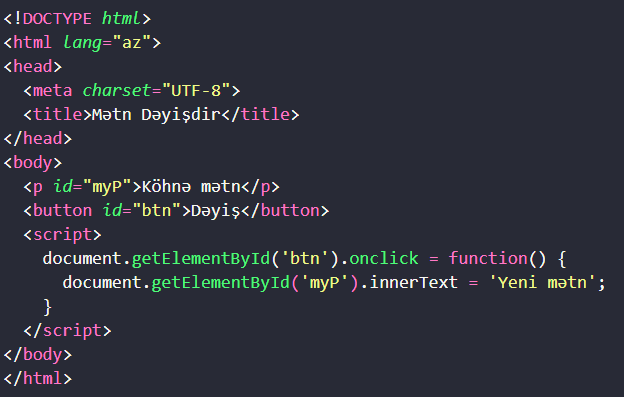
1. **JavaScript-də klik hadisəsinə qulaq asan və xəbərdarlıq (alert) göstərən bir funksionallıq əlavə edin. (asan)**

****

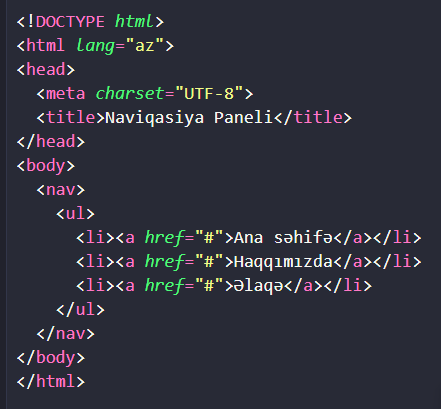
1. **Alternativ mətni və eni 300px olan bir şəkli HTML ilə yerləşdirin. (asan)**

****

1. **JavaScript ilə düyməyə klik edildikdə bir <p> elementinin mətnini dəyişin. (asan)**

****

1. **Semantik teqlərdən istifadə edərək naviqasiya paneli yaradın. (asan)**

****

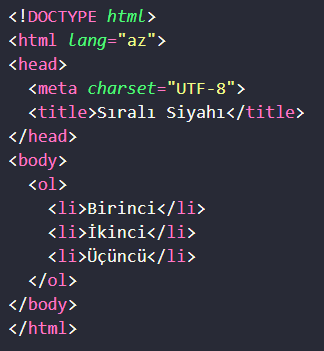
1. **Ekranın eni 500px-dən az olduqda fon rəngini sarıya çevirən media sorğusu (media query) yazın. (asan)**

****

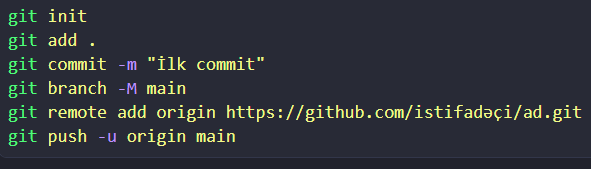
1. **CSS ilə, üzərinə siçanı gətirdikdə rəngi dəyişən bir düymə yaradın. (asan)**

****

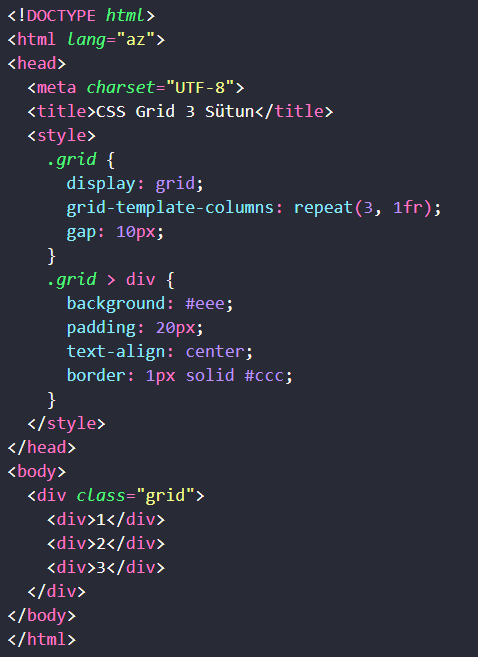
1. **HTML-də 3 elementli sadə sıralı siyahı yaradın. (asan)**

****

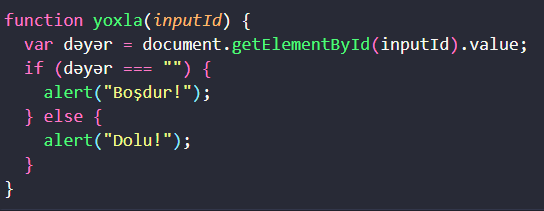
1. **Git əmrləri ilə repo yaradın və onu GitHub-a göndərin (push edin). (asan)**

****

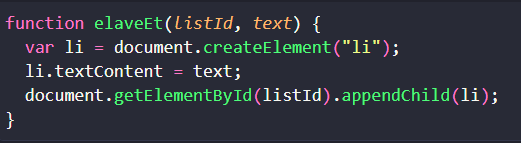
1. **CSS Grid istifadə edərək adaptiv 3 sütunlu düzülüş yaradın. (orta)**

****

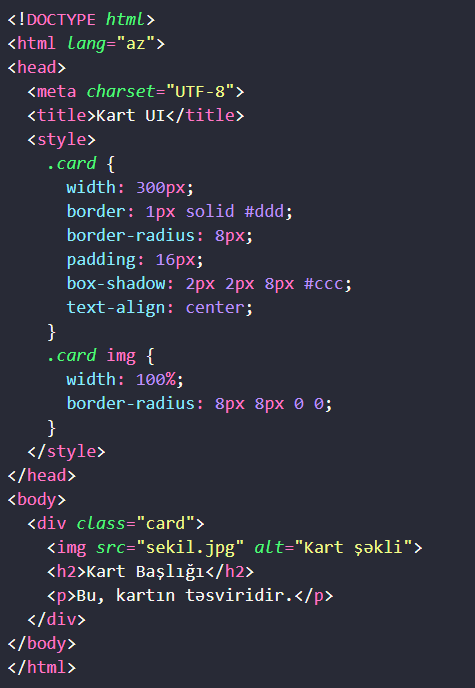
1. **Bir forma sahəsinin boş olub-olmadığını yoxlayan JavaScript funksiyası yazın. (orta)**

****

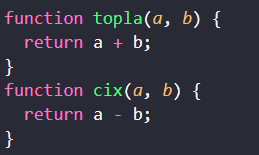
1. **JavaScript istifadə edərək nömrələnməmiş siyahıya dinamik şəkildə yeni element əlavə edin. (orta)**

****

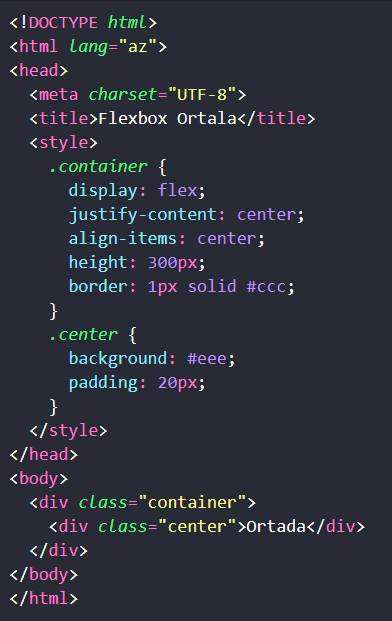
1. **Şəkil, başlıq və abzasdan ibarət olan kart UI üçün HTML və CSS yazın. (orta)**

****

1. **JavaScript ilə toplama və çıxma funksiyalarını yerinə yetirən sadə kalkulyator yaradın. (orta)**

****

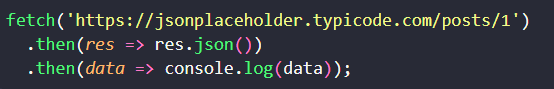
1. **Flexbox istifadə edərək bir div-i şaquli və üfüqi olaraq ortalayacaq CSS kodu yazın. (orta)**

****

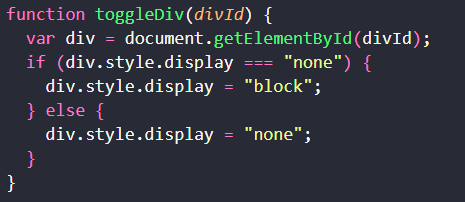
1. **CSS xüsusiyyətlərindən istifadə edərək bir səhifə hissəsini yapışqan (sticky) edin. (orta)**

****

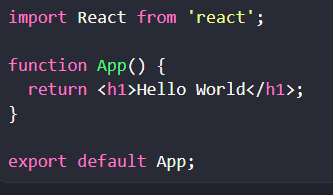
1. **JavaScript ilə bir API son nöqtəsindən məlumat əldə edin və nəticəni konsola yazın. (orta)**

****

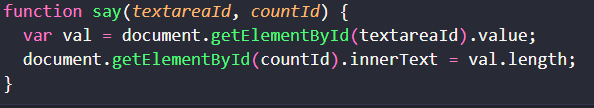
1. **JavaScript ilə bir div-i göstər/gizlət funksionallığı əlavə edin. (orta)**

****

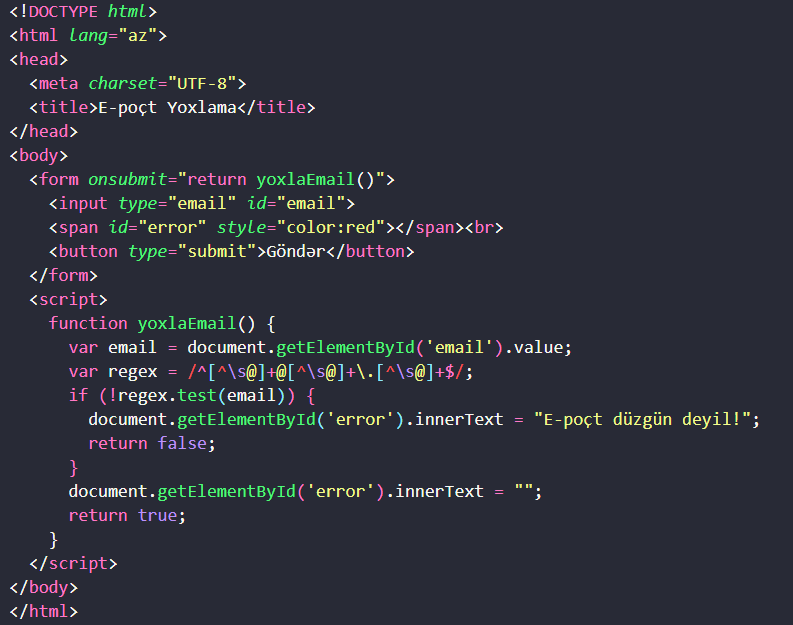
1. **"Hello World" göstərən funksional komponentlə əsas React tətbiqi qurun. (orta)**

****

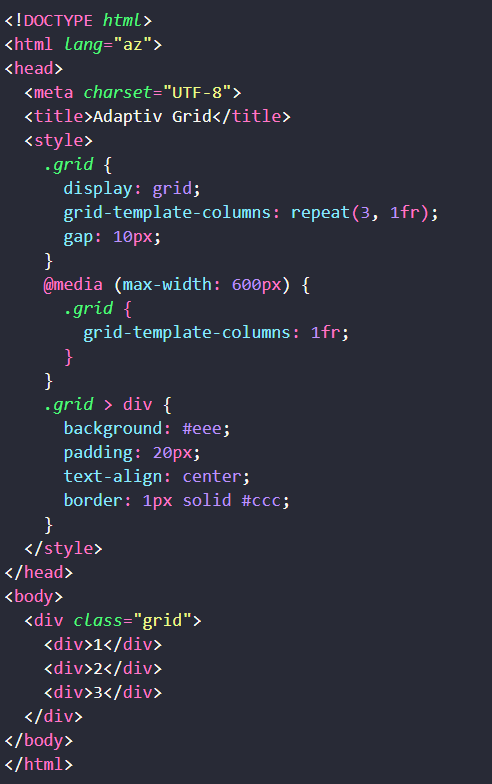
1. **JavaScript ilə textarea sahəsində yazılan simvolların sayını real vaxtda hesablayın. (orta)**

****

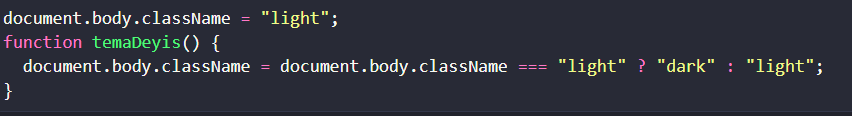
1. **E-poçt düzgün olmadıqda səhv mesajı göstərən yoxlama ilə forma yaradın. (orta)**

****

1. **Yalnız CSS Grid və media sorğuları ilə adaptiv veb səhifə düzülüşü yaradın. (çətin)**

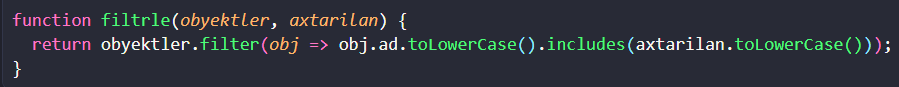
****

1. **JavaScript və CSS istifadə edərək tema (açıq/qara) dəyişdirən kiçik tətbiq yaradın. (çətin)**

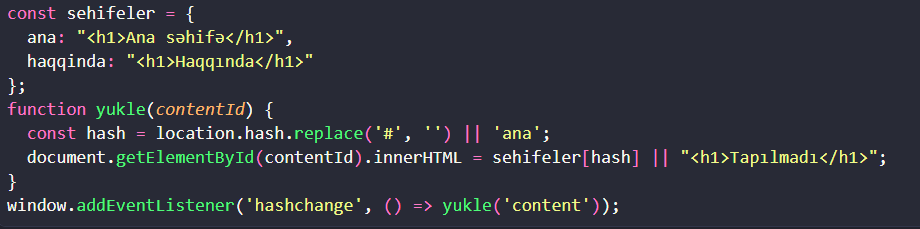
****

1. **İstifadəçi daxiletməsinə əsaslanaraq obyektlər massivini süzgəcdən keçirən JavaScript funksiyası yazın.**

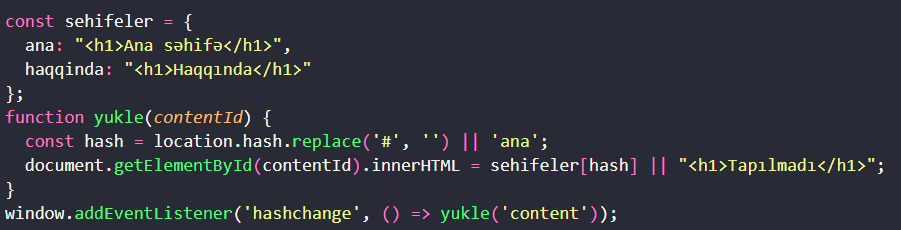
**(çətin)**

****

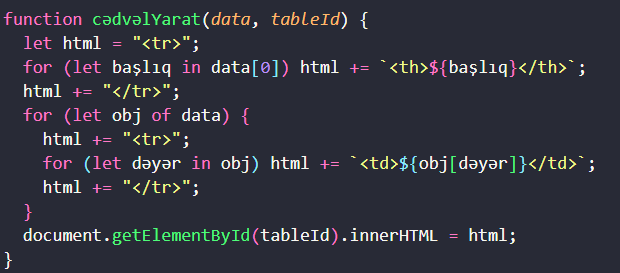
1. **Sadə SPA (Single Page Application) davranışını təmiz JavaScript və DOM manipulyasiyası ilə yaradın. (çətin)**

****

1. **Formanın daxil edilən məlumatlarını localStorage-də saxlayan və səhifə yeniləndikdə onları bərpa edən forma yaradın. (çətin)**

****

1. **JSON verilənlərinə əsaslanaraq dinamik şəkildə cədvəl yaradan JavaScript funksiyası yazın. (çətin)**

****