

Ödev

1. JavaScript nedir ve tarihsel gelişiminden bahsedin
2. Java ile javascript arasındaki fark nedir
3. Javascript teki veri tipleri nelerdir açıklayınız
4. null ile undefined arasındaki fark nedir açıklayınız
5. NaN nedir açıklayınız
6. Javascript'te yorum satırı eklemenin kaç farklı yolu vardır
7. Global değişken ne demektir açıklayınız
8. Javascript'te this anahtar kelimesi nedir açıklayınız
9. == ile === farkını örnekler ile açıklayınız
10. let var const farkını tablo yapınız
11. Arrow fonksiyonun normal fonksiyondan farkları nelerdir
12. swich bloğu içinde hatasız nasıl değişken tanımlanır
13. Pure fonksiyon ne demektir açıklayınız
14. Rest operatör nedir örnekle açıklayınız
15. Object destructuring nedir örnekle açıklayınız
16. 2 elemanlı bir objeyi 6 farklı şekilde oluşturunuz
17. 2 elemanlı bir objenin key ve value değerlerinin karakter sayısı ile 2 farklı döngü methodu kullanarak yeni bir obje oluşturunuz
18. Cookie, local storage ve session storage farkını tablo yapınız
19. asenkron ve senkron işlem farkı nedir
20. promise nedir ve neden ihtiyaç duyarız

Array Soruları

```
var dolap = ["Shirt", "Pant", "TShirt"];
```

1. dolap arrayindeki son elemanı silip consola yazdırın
2. dolap arrayindeki ilk elemanı silip yerine “Hat” elemanını gönderip consola yazdırın
3. dolap değişkeninin array olup olmadığını kontrol edin ve sonucu bir değişkene eşitleyin
4. dolap arrayinde “Pant” elemanın olup olmadığını 3 farklı method ile kontrol edin
5. dolap arrayindeki elemanların karakter sayısını toplayıp geriye döndürecek fonksiyonu yazın
6. dolap arrayindeki tüm elemanları büyük harfe çevirip yeni bir değişkene 3 farklı yöntemle atayın
7. dolap arrayini index sayıları key olacak şekilde objeye çeviriniz
8. slice ile splice farkı nedir

```
const arr = [1,2,3,4,5,6,7,7,8,6,10];
```

1. arrayindeki yinelenen sayıları bulun
2. arrayindeki tüm yinelenen sayıları silip yeni bir arrayi 2 farklı method ile oluşturun
3. arrayindeki en yüksek ve en düşük değeri 2 farklı methodla bulun

```
// Bu kodun çıktısı nedir neden ?
function job() {
  return new Promise(function(resolve, reject) {
    reject();
  });
}

let promise = job();

promise

.then(function() {
  console.log('Success 1');
})
.then(function() {
  console.log('Success 2');
})
```

```

})
.then(function() {
    console.log('Success 3');
})
.catch(function() {
    console.log('Error 1');
})
.then(function() {
    console.log('Success 4');
});

```

```

// Bu kodun çıktısı nedir neden ?
function job(state) {
    return new Promise(function(resolve, reject) {
        if (state) {
            resolve('success');
        } else {
            reject('error');
        }
    });
}

let promise = job(true);

promise
.then(function(data) {
    console.log(data);

    return job(true);
})

.then(function(data) {
    if (data !== 'victory') {
        throw 'Defeat';
    }

    return job(true);
})
.then(function(data) {
    console.log(data);
})
.catch(function(error) {
    console.log(error);

    return job(false);
})
.then(function(data) {

```

```
        console.log(data);

        return job(true);
    })
    .catch(function(error) {
        console.log(error);

        return 'Error caught';
    })
    .then(function(data) {
        console.log(data);

        return new Error('test');
    })
    .then(function(data) {
        console.log('Success:', data.message);
    })
    .catch(function(data) {
        console.log('Error:', data.message);
    });
```