

# TW-08 GROUP VERSION

---



CLARUSWAY  
WAY TO REINVENT YOURSELF

## Meeting Agenda

---

- ▶ Icebreaking
- ▶ Workshop Activities
  - ▶ Questions
- ▶ Teamwork Activities
  - ▶ Interview Questions
  - ▶ Video of the week
  - ▶ Case study / project
- ▶ Retro meeting

## Workshop Activities

90m

- 04 - TR ID Validator

# Teamwork Schedule

---

## Ice-breaking

10m

- Personal Questions (Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

## Ask Questions

30m

### 1. What is the purpose of the `async` keyword in JavaScript?

A. To define an asynchronous function.

B. To specify a function that runs synchronously.

C. To create a generator function.

D. To indicate a callback function.

JavaScript'te `async` anahtar kelimesinin amacı, bir fonksiyonun asenkron (asynchronous) olarak tanımlanmasını sağlamaktır. Bu, fonksiyonun Promise döndürebileceği anlamına gelir. Asenkron fonksiyonlar, zaman alan işlemlerin (örneğin, API çağrıları veya dosya okuma gibi) daha okunabilir ve yönetilebilir bir şekilde yapılmasına olanak tanır.

### 2. In an `async` function, what does the `await` keyword do?

A. To forcefully stop the function.

B. To pause the function execution until a promise is resolved.

C. Rejects a promise with an error.

D. Cancels the `async` function.

`await` anahtar kelimesi, bir `async` fonksiyon içinde kullanılır ve bu fonksiyonun çalışmasını, belirtilen bir promise çözülene veya reddedilene kadar duraklatır. Bu sayede, asenkron işlemler daha okunabilir bir şekilde yazılabilir.

### 3. Which of the following is a valid use case for using `async/await` in JavaScript?

A. Iterating through an array of numbers.

B. Declaring constants.

C. Defining a switch statement.

D. Fetching data from an API.

`async/await`, JavaScript'te asenkron işlemleri yönetmek için kullanılır. Özellikle, bir API'den veri almak gibi zaman alıcı işlemleri senkron bir kod akışı gibi yazmak için idealdir.

#### 4. What is the purpose of the Fetch API in JavaScript?

- A. To create an array from a function.
- B. To manipulate the DOM.
- C. To send HTTP requests and receive responses.
- D. To define JavaScript functions.

Fetch API, JavaScript'te HTTP istekleri göndermek ve sunucudan gelen yanıtları almak için kullanılan bir web API'sidir. Modern tarayıcılarda yerle ik olarak bulunur ve genellikle asenkron veri alı veri i için kullanılır.

#### 5. When using the Fetch API, what method is used to handle a response and extract JSON data?

- A. .text()
- B. .json()
- C. .response()
- D. .xml()

#### 6. Which statement is true about error handling in async/await?

- A. Errors are automatically handled by async/await.
- B. Use .then() and .catch() to handle errors.
- C. Errors in async/await cannot be caught.
- D. Use try-catch blocks to catch errors.

async/await, hata yönetimi için try-catch bloklarıyla birlikte kullanılabilir. Bir await ifadesi hata olu turursa, bu hata try-catch blo u içinde yakalanabilir. Bu yöntem, asenkron hataları senkron kod gibi yönetmeyi sa lar.

#### 7. What is the primary benefit of using async/await over traditional callbacks?

- A. Simpler and more readable code
- B. Easier handling of errors
- C. Smaller bundle size
- D. Improved performance

async/await, geleneksel geri ça rılar (callbacks) ve Promise zincirlerine kıyasla kodun daha okunabilir ve anla ılır olmasını sa lar. Asenkron i lemler, senkron kod gibi yazılabildi i için karma ık "callback hell" durumlarını önler.

#### 8. What does the setInterval() function do in JavaScript?

- A. Halts the execution of the JavaScript code.
- B. Delays the execution of a function for a specified time.
- C. Executes a function repeatedly at a specified time interval.
- D. Sets a timeout for a function to run only once.

setInterval() fonksiyonu, belirli bir zaman aralı ı ile bir fonksiyonu sürekli olarak tekrarlamak için kullanılır. Bu fonksiyon, her belirtti i niz süre (milisaniye cinsinden) sonunda belirtilen fonksiyonu tekrar ça ırır.

#### 9. How do you stop the execution of a function scheduled with setInterval()?

- A. Using the return statement in the function.
- B. By calling the clearInterval() function with the interval ID.
- C. By setting the interval time to 0.
- D. By using the break statement.

setInterval() fonksiyonu bir interval ID'si döndürür ve bu ID, intervalı durdurmak için clearInterval() fonksiyonu ile kullanılır. Bu, belirli bir fonksiyonun sürekli olarak ça lı masını durdurmanıza olanak tanır.

### 10. When using `clearInterval()`, what is the required argument?

- A. The function to be cleared.
- B. The time interval.
- C. The DOM element
- D. The interval ID returned by `setInterval()`.

`clearInterval()` fonksiyonu, `setInterval()` tarafından döndürülen interval ID'sini alarak belirli bir intervalin çalışmasını durdurur. Bu ID, intervalin hangi fonksiyonla iliştirildiğini belirlemek için kullanılır.

### 11. What is the primary advantage of using Axios over the Fetch API?

- A. Simplicity and ease of use.
- B. Wider browser support.
- C. Smaller bundle size.
- D. Better performance.

Axios, Fetch API'ye göre bazı avantajlar sunar, ancak en büyük avantajı daha basit ve daha kullanıcı dostu olmasıdır. Axios, bir dizi özellik sunar ve bu özellikler Fetch API'de ekstra kod gerektirir. Örneğin:

### 12. When would 'results shown' be logged to the console?

```
let modal = document.querySelector('#result');
setTimeout(function () {
  modal.classList.remove('hidden');
}, 10000);
console.log('Results shown');
```

- A. after 10 second
- B. immediately
- C. after results are received from the HTTP request
- D. after 10000 seconds

`console.log('Results shown')` hemen çalıştırılır. 10 saniye sonra `setTimeout` fonksiyonu çalışır ve `modal.classList.remove('hidden')` çalıştırılır.

### 13. Why might you choose to make your code asynchronous?

- A. to ensure that tasks further down in your code are not initiated until earlier tasks have completed
- B. to make your code faster
- C. to ensure that the call stack maintains a LIFO (Last in, First Out) structure
- D. to start tasks that might take some time without blocking subsequent tasks from executing immediately

Asenkron programlama, zaman alacak görevlerin (örneğin, istekleri, dosya okuma/yazma işlemleri) yürütülmesini sağlar, ancak bu görevler tamamlanana kadar programın geri kalanı engellenmez. Bu, uygulamanın başka görevleri yapabilmesi ve daha verimli çalışabilmesi anlamına gelir.

### 14. What is the HTTP verb to request the contents of an existing resource?

- A. DELETE
- B. PATCH
- C. GET
- D. CALL

GET HTTP metodu, mevcut bir kaynağın içeriğini istemek için kullanılır. Bu istek, sunucudan veri almak amacıyla yapılır ve genellikle bir web sayfasının ya da API'nin içeriğini çekmek için kullanılır.

### 15. Which method call is chained to handle a successful response returned by `fetch()`?

- A. `done()`
- B. `catch()`
- C. `then()`
- D. `finally()`

`then()` metodu, `fetch()` fonksiyonu tarafından döndürülen Promise nesnesi üzerinde ba arılı bir yanıt alındı ında ça rılır. Bu metod, ba arılı bir HTTP yanıtı alındı ında yapılacak i lemleri tanımlamak için kullanılır.

### 16. Which statement is applicable to the `defer` attribute of the HTML `<script>` tag?

- A. `defer` allows the browser to continue processing the page while the script loads in the background.
- B. `defer` causes the script to be loaded from the backup content delivery network (CDN).
- C. `defer` blocks the browser from processing HTML below the tag until the script is completely loaded.
- D. `defer` lazy loads the script, causing it to download only when it is called by another script on the page.

`defer` özelli i, `<script>` etiketine eklendi inde, tarayıcının HTML belgesinin geri kalanını i lemeye devam etmesine izin verirken, script'in arka planda yüklenmesine olanak tanır. Bu, sayfanın hızlı bir eilde yüklenmesini sa lamak için özellikle kullanı lıdır. `defer` özelli i, script'in tüm HTML içeri i yüklendikten sonra, belgeyi i leme devam ederken ça lı tılmasını garanti eder.

### 17. What will be logged to the console?

```
console.log('I');
setTimeout(() => {
  console.log('love');
}, 0);
console.log('Javascript!');
```

`console.log('I')` hemen ça lı tılır ve "I" konsola yazdırılır.

- ☒ A.

`setTimeout()` fonksiyonu, 0 milisaniye (veya hemen) bekleyerek `console.log('love')` fonksiyonunu ça lı tırmaya karar verir, ancak `setTimeout()` asenkron bir i lem oldu u için bu i lem call stack'e eklenmez. Bunun yerine, event loop'a aktarılır.

```
I
Javascript!
love
```

`console.log('Javascript!')` hemen ça lı tılır ve "Javascript!" konsola yazdırılır.

Son olarak, `setTimeout()` içindeki `console.log('love')` fonksiyonu, call stack bo aldıktan sonra ça lı tılır ve "love" konsola yazdırılır.

- ☐ B.

Bu davranı , JavaScript'in asenkron yapısına ve event loop mekanizmasına dayanmaktadır. `setTimeout()` fonksiyonu, ça rı yı nına (call stack) hemen eklenmez, bunun yerine event loop aracılı ıyla eklenir ve mevcut stack bo aldıktan sonra ça lı tılır.

```
love
I
Javascript!
```

- ☐ C. The output may change with each execution of code and cannot be determined.
- ☐ D.

```
I
love
Javascript!
```

## Interview Questions

30m

### 1. What is 'callback hell' in the context of using callbacks, and what are its disadvantages?

Callback hell (veya "callback çukuru"), JavaScript gibi asenkron programlama dillerinde birden fazla callback fonksiyonunun iç içe yazılması sonucu ortaya çıkan karmaşık ve okunması zor bir durumu tanımlar. Bu durum, özellikle asenkron i lemler birbiri ardına çağıldı ında veya iç içe fonksiyonlar kullanıldı ında yaygınla ır.

### 2. Can you explain what asynchronous programming is?

Asenkron programlama, programın bir i lemi veya görevi başlattı ında, bu i lemi beklemeden di ğer görevlerin çalışmaya devam etmesine olanak tanıyan bir programlama modelidir.

### 3. How does async/await help with performance and scalability?

Async/await, performans ve ölçeklenebilirlik açısından önemli faydalar sağlar. Asenkron programlamanın temel amacına uygun olarak, async/await, zaman alıcı i lemleri beklerken programın di ğer bölümlerinin çalışmasına olanak tanır, bu da daha verimli bir kaynak kullanımı ve daha hızlı tepki süreleri anlamına gelir.

### 4. What is the purpose of the Promise constructor in JavaScript?

Promise constructor'ı, JavaScript'te asenkron i lemleri yönetmek için kullanılan bir yapıdır. Bir Promise nesnesi, gelecekte tamamlanacak (ya da tamamlanmayacak) bir i lemin sonucunu temsil eder. Promise, i lemin başarılı olup olmadığını belirten bir durum ta ır ve başarılı durumda bir de ğer döner, başarısızlık durumunda ise bir hata (reason) döner.



## Coffee Break

10m



## Video of the Week

10m

- Asynchronous Vs Synchronous Programming

## Case study/Project

15m

- 05 - Random User

## Retro Meeting on a personal and team level

10m

Ask the questions below:

- What went well?
- What could be improved?
- What will we commit to do better in the next week?

## Closing

10m

- Next week's plan
  - QA Session
-