

JS Advanced

23/12/2024

מבוא לתכנות א-סנכרוני



מבוא לתכנות א-סנכרוני

JS_Advanced_Asynchronous1.html

בחלק זה נלמד מגוון נושאים.
בסיום הנושא תוכלו לענות על השאלות הבאות:

- מהו תכנות סנכרוני?
- מהו תכנות א-סנכרוני?
- איזה בעיות יכולות להיווצר בקוד?

```
function connectToServer() {
    setTimeout(function () { console.log("response from server"); }, 3000);
}
```

```
console.log("do something 1..");
console.log("do something 2..");
connectToServer();
console.log("do something 3..");
console.log("do something 4..");
```

קריאה לשרת עשויה לקחת
זמן, אך הקוד ממשיך לרוץ

```
do something 1..
do something 2..
do something 3..
do something 4..
response from server
```

מבוא לתכנות א-סנכרוני

JS_Advanced_Asynchronous2.html

```
function fn1() {
  console.log(1);
}
```

```
function fn2() {
  console.log(2);
}
```

```
function fn3() {
  console.log(3);
}
```

```
function fn4() {
  console.log(4);
}
```

```
setTimeout(fn1, 500);
fn2();
fn3();
fn4();
```

הפעולה עשויה לקחת זמן,
אך הקוד ממשיך לרוץ

בחלק זה נלמד מגוון נושאים.
בסיום הנושא תוכלו לענות על השאלות הבאות:

- מהו תכנות סנכרוני?
- מהו תכנות א-סנכרוני?
- איזה בעיות יכולות להיווצר בקוד?

2
3
4
1

קריאות `XMLHttpRequest`



קריאות XMLHttpRequest

JS_Advanced_XMLHttpRequest.html

```
function demo1() {
```

```
// Create New XMLHttpRequest object
const request = new XMLHttpRequest();
```

יצירת אובייקט חדש מסוג
XMLHttpRequest

```
// callback function for onload
```

```
request.onload = function () {
  console.log(request);
  console.log(request.status); // Checking the status
  console.log(request.response); // Checking the response (data)
```

הגדרת מתודה שתרוץ לאחר
שליחה (במצב onload)

```
const myObj = JSON.parse(request.response)
console.log(myObj)
```

```
// Open and Send request
```

```
request.open("GET", "https://pokeapi.co/api/v2/pokemon/squirtle");
request.send();
```

שליחה והפעלת הבקשה

פתיחת הבקשה
והגדרת הבקשה

בחלק זה נלמד מגוון נושאים.
בסיום הנושא תוכלו לענות על השאלות הבאות:

- מהי קריאה לשרת?
- איך מבצעים קריאה לשרת?
- מה מכיל אובייקט מסוג XMLHttpRequest?
- אילו פעולות צריך לבצע לביצוע הקריאה לשרת?
- על מה אחראית המתודה שב-onload?
- על מה אחראית המתודה open?
- על מה אחראית המתודה send?
- באיזה סדר צריך לבצע את הפעולות שיש לבצע?
- איך ניתן לגשת למידע שהתקבל?
- איך ניתן לבדוק את סטטוס הפניה?
- מהם אפשרויות ה-http הנפוצים בשימוש? (GET, POST, PUT, DELETE)
- איך ניתן לגשת להודעות שגיאה (במידה וקיימות)?

תרגול קריאות XMLHttpRequest

צרו קובץ חדש בשם `JS_XMLHttpRequest` לטובת הנושא ופתרו את התרגילים לפי הסדר חשוב להדפיס הודעות הצלחה ושגיאה למשתמש וכמובן לבדוק כל תרגיל בסיום.

getData

country	"Israel"
---------	----------

printData

country	Prompt
countryData	getData(country)

תרגיל	תיאור המשימה
Ex-1	צרו פונקציה חדשה וקראו לה <code>getData</code> . פונקציה זו תדע לקבל פרמטר שהוא שם של מדינה. פונקציה זו תדע לפנות ל-API של מדינות העולם על מנת להביא מידע על המדינה שקיבלנו כפרמטר. כתובתו של ה-API היא: <code>https://restcountries.com/</code>
Ex-2	באמצעות שימוש ב- <code>XMLHttpRequest</code> , פנו ל-API וייבאו את הנתונים המבוקשים. היפכו את התוצאה המתקבלת ל-JSON, הפונקציה תדפיס לקונסול ותחזיר תוצאה זו.
Ex-3	צרו פונקציה חדשה בשם <code>printData</code> . פונקציה זו תבקש מהמשתמש שם של מדינה, לאחר מכן הפונקציה תשתמש בפונקציה <code>getData</code> על מנת להעביר את שם המדינה ולקבל את התוצאה. בסוף פונקציה זו תדפיס את התוצאה לקונסול.
Ex-4	השתמשו ב- <code>printData</code> באמצעות שליחת ערכים שונים ובידקו האם ההדפסות שבפונקציה זו יוצאות לפועל לפני או אחרי שחזרה התשובה מהשרת.

תודה על ההקשבה

אני וצוות המכללה כאן עבורכם