

# JS Advanced

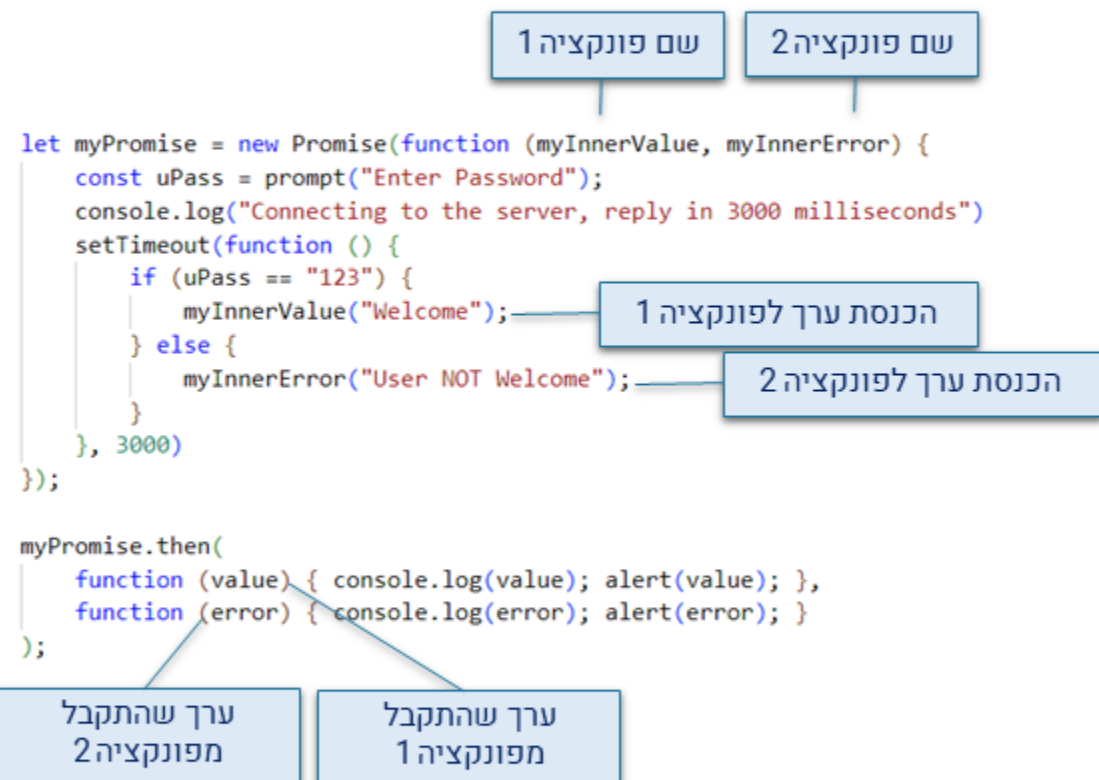
26/12/2024

# מחלקת Promise



# מחלקת Promise

JS\_Advanced\_Promise.html



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

- מה זה Promise?
- מה ההבדל בין resolve ל-reject?
- מה נצטרך לשלוח לבנאי של Promise?
- באיזה מצבים נזדקק להשתמש בטכניקה זו?
- מה הם המצבים האפשריים של Promise?
- מה הם המאפיינים והמתודות המיוחדים למחלקת Promise?

# מחלקת Promise

JS\_Advanced\_Promise.html

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

```
let myPromise = new Promise(function (funcValue, funcError) {
  console.log("Connecting to the server, reply in ??? milliseconds");
  const url = 'https://pokeapi.co/api/v2/pokemon/squirtle';
  const request = new XMLHttpRequest();
  request.open("GET", url);
  request.onload = function () {
    if (request.readyState == 4 && request.status == 200) {
      funcValue(request.response);
    } else {
      funcError("Error !!!!");
    }
  };
  request.send();
  console.log("end");
});

myPromise.then(
  function (value1) { console.log(value1); },
  function (value2) { console.log(value2); }
);
```

- מה זה Promise?
- מה ההבדל בין resolve ל-reject?
- מה נצטרך לשלוח לבנאי של Promise?
- באיזה מצבים נזדקק להשתמש בטכניקה זו?
- מה הם המצבים האפשריים של Promise?
- מה הם המאפיינים והמתודות המיוחדים למחלקת Promise?

# מחלקת Promise

JS\_Advanced\_Promise.html

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

```
let myPromise = new Promise(function (funcValue, funcError) {

    console.log("Connecting to the server.. ");

    const request = new XMLHttpRequest();
    request.onload = function () {
        if (request.readyState == 4 && request.status == 200) {
            funcValue(request.response);
        } else {
            funcError("Error !!!!");
        }
    };

    request.open("GET", 'https://pokeapi.co/api/v2/pokemon/squirtle');
    request.send();
    console.log("end");
});

myPromise.then(
    function (value1) { console.log(value1); },
    function (value2) { console.log(value2); }
);
```

- מה זה Promise?
- מה ההבדל בין resolve ל-reject?
- מה נצטרך לשלוח לבנאי של Promise?
- באיזה מצבים נזדקק להשתמש בטכניקה זו?
- מה הם המצבים האפשריים של Promise?
- מה הם המאפיינים והמתודות המיוחדים למחלקת Promise?

# מחלקת Promise - שימוש ב then - catch

JS\_Advanced\_Promise\_Then\_Catch.html

JS\_Advanced\_PromiseAll.html

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

```
function getPromise(url) {
  let myPromise = new Promise(function (resolve, reject) {
    let request = new XMLHttpRequest();
    request.open("GET", url);
    request.onload = () => {
      if (request.status === 200) {
        resolve(request.response);
      } else {
        reject("Error");
      }
    };
    request.send();
  });
  return myPromise;
}

const pokemonsUrl = 'https://pokeapi.co/api/v2/pokemon?limit=20';
let promise = getPromise(pokemonsUrl);

const promiseHandler = () => {
  promise.then((result) => {
    let pokemonsArray = [...JSON.parse(result)['results']]
    pokemonsArray.forEach(item => console.log(item['name']));
  }).catch((error) => {
    console.log(error.message); // The code will run only if there is an error
  }).then(() => {
    console.log('\n this will be printed at the end of program, bye')
  });
}

promiseHandler();
```

- למה משמשת המתודה ?then
- למה משמשת המתודה ?catch
- איזה פרמטרים ניתן לשלוח למתודה ?then
- איך ניתן להשתמש ב-then למטרות סנכרון?
- איזה פרמטרים ניתן לשלוח למתודה ?catch
- מה היתרון בשימוש ב-catch?
- איך נוכל למשוך נתונים מ-API באמצעות שימוש ב-Promise ו-XMLHttpRequest?
- מה נוכל לעשות במקרה של שגיאה שמקורה בשרת?

# תרגול מחלקת Promise

צרו קובץ חדש בשם **JS\_Promise** 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)

## setImages

countries	[country1, country2, country3, country4, country5]
-----------	--

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: <a href="https://restcountries.com/">https://restcountries.com/</a>
Ex-2	באמצעות שימוש במחלקות <code>Promise</code> ו- <code>XMLHttpRequest</code> , פנו ל-API וייבאו את הנתונים המבוקשים. הפונקציה תחזיר את ה-Promise.
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 תמיר את התוצאה ל-JSON ותדפיס את אותה לקונסול באמצעות שימוש ב- <code>then</code> .
Ex-4	צרו פונקציה חדשה בשם <code>setImages</code> , ייבאו בה 5 דגלים של מדינות על ידי שימוש בפונקציה <code>getData</code> . לבסוף הציגו תמונות אלו על גבי הדפדפן ללא יצירת אלמנטים מראש במסמך ה-HTML באמצעות שימוש ב- <code>then</code> .
Ex-5	שילחו שמות שגויים של מדינות ל-API, טפלו בשגיאות באמצעות שימוש במתודה <code>catch</code> כך שבמידה והתקבלה שגיאה אז תוצג תמונה קבועה מראש במקום התמונה שלא התקבלה מהשרת.

# תודה על ההקשבה

אני וצוות המכללה כאן עבורכם