BATCH:

DT/NT :

LESSON :

SUBJECT:

TECHPRO EDUCATION

techproeducation.com

+1 (585) 304 29 59

JavaScript

Javascript

Introduction

- Javascript nedir?
- Compiler & Interpreter
- Kullanım amaçları
- Sürümleri
- Popülarite
- İlk kodlar

Javascript nedir?

INTERVIEW

JS

Yüksek seviyeli, nesne yönelimli, yorumlayıcı tabanlı ve dinamik bir programlama dilidir.

Yüksek Seviyeli

Hafıza yönetimi gibi karmaşık görevleri düşünmemize gerek yoktur.

Nesne Yönelimli

Nesne özelliklerinin (kalıtım, çok şekillilik v.b) kullanılmasına imkan sağlamaktadır.

Yorumlayıcı Tabanlı

Derleyicide olduğu gibi tüm komutların bir kere de makine koduna çevirmek yerine tek-tek alınıp makine koduna çevrilip çalıştırılmasını sağlar.

TECHPRO EDUCATION

Nerede kullanılır?

HTML

JAVASCRIPT

CSS

TECHPRO EDUCATION

Neden Javascript

Front-end Dev

React, Svelte, Vue, Angular gibi JS framework ve kütüphanleri ile web ve mobil uygulamalar geliştirilebilmektedir.

Facebook, Netflix, Dropbox, AirBnb

Back-end Dev

NodeJs ile sunucu tarafında uygulamalar geliştirilebilmektedir.

Netflix, Uber, E-Bay

TECHPRO EDUCATION

Neden Javascript

Mobile Dev

React Native

IONIC

NativeScript

React Native, Ionic, Native Script gibi kütüphaneler ile mobile uygulama geliştirilebilir.

Facebook, Instagram, Skype, Uber, Pinterest

Desktop Dev

Electron kütüphanesi ile masaüstünde çalışacak uygulamalar geliştirilebilir.

VSCode, Whatsapp, Slack, Skype, Twitch, Teams

Java Script in Gelişimi

- 1995 — Mocha
- 1997 — ES 1
- 1998 — ES 2
- 2000 — ES 3
- 2008 — ES 4
- 2009 — ES 5
- 2015 — ES 6
- 2016 — ES 7
- 2017 — ES 8
- 2018 — ES 9
- 2019 — ES 10
- 2020 — ES 11
- 2021 — ES 12
- 2022 — ES 13
- 2023 — ES 14

TECHPRO EDUCATION

Google Trends

https://pypl.github.io/PYPL.html

TECHPRO EDUCATION

A Survey

https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

TECHPRO EDUCATION

Github repos

2 milyar civarındaki internet sitesinde %95 oranında Javascript kullanılmaktadır.

Neden popüler?

Kurulum gerekmez

Object Oriented

End to end programming

Library & Framework

Community

TECHPRO EDUCATION

İlk kodlar

Javascript in test amaçlı kullanımı

Boş bir chrome sayfasında sağ tıklayıp inspect (incele) dedikten sonra

1 Sources (Kaynaklar) tıklanır

2 Snippets tıklanır

3 New snippet a tıklanıp isim verilir.

4 Kodlar yazılır

5 Ctrl+Enter ile çalıştırılır

TECHPRO EDUCATION

İlk kodlar

```
<html>
    <head>

    </head>
    <body>


        <script>
            …
        </script>
    </body>
</html>
```

- Javascript kodlarını yazmak için script tagı kullanılır.
- Tüm javascript kodları script tagları arasına yazılır
- Script taglarının body kapanmadan hemen önce konulması tavsiye edilir.
- Bu kullanıma internal script denir ve javascript kodları sadece mevcut sayfa için geçerli olur.

TECHPRO EDUCATION

İlk kodlar

```
<html>
    <head>

    </head>
    <body>


        <script src="assets/js/script.js">
    </body>
</html>
```

- Javascript kodları harici bir dokümana konularak kullanılırsa buna external script denir
- Bu dosyayı kullanan tüm html sayfalarında kullanılabilir hale gelir.

TECHPRO EDUCATION

Kodların çalıştırılması

1

Web sayfalarında bulunan Javascript kodları, tarayıcı tarafından çalıştırıldığı için html dosyasına dosyaya çift tıklayarak açmak javascript dosyalarının çalışması için yeterlidir.

2

VSCode Live Server eklentisi ile çalıştırılırsa, yapılan değişiklikler de anında yansıtılmış olur. Ve gerçeğe yakın bir ortamda geliştirme yapılabilir.

TECHPRO EDUCATION

JavaScript

Fundamentals

- Syntax
- Console
- Alert, confirm, prompt
- Comment
- Variables and constants
- Operators

Javascript syntax

INTERVIEW

- Case sensitive bir dildir.
- Satır sonlarına genellikle noktalı virgül (;) konulur.
- Identifier lar(variables, constants, functions) harf, _ veya $ ile başlayabilir.
- Identifier lar camel case ile yazılır.

TECHPRO EDUCATION

Case Types

camelCase

snake_case

kebab-case

Train-Case

PascalCase

MACRO_CASE

TECHPRO EDUCATION

Console

Javascript ile tarayıcıların console bölümüne birşeyler yazdırmak için console object kullanılır.

```
console.log("bla bla bla");
```

TECHPRO EDUCATION

Console

Console object inin farklı kullanımları vardır.

```
console.log("Hello World");

console.warn("Sayılar eşit");

console.error("ID alanı boş olamaz");
```

TECHPRO EDUCATION

Alert, Confirm, Prompt

Alert
Bilgilendirme yapmak için kullanılır

```
alert("Silmeye yetkiniz yok");
```

Confirm
Soru sormak için kullanılır.

```
confirm("Silmek istediği….");
```

Prompt
Bilgi almak için kullanılır.

```
prompt("Adınızı giriniz");
```

TECHPRO EDUCATION

Comment

INTERVIEW

```
// Bu bir comment satırıdır.


/*
 Bu şekilde birden çok satırı aynı
anda comment içine alabilirsiniz.
*/
```

TECHPRO EDUCATION

Variables (Değişkenler), Constants (Sabitler)

Programlama dillerinde geçici olarak değerleri saklamak için kullanılan yapılara değişken ya da sabit adı verilir.

Variables - Constants

INTERVIEW

Javascript te değişken veya sabit tanımlamak için 3 farklı ifade kullanılır

var

ES 6

let

Değişken

ES 6

const

sabit

TECHPRO EDUCATION

Variables - Constants

```
var sayi;
var Sayi;
var genelOrtalama;
const kdv = 1.18;
```

- case sensitive yani büyük küçük harf duyarlıdır.
- Sabitler tanımlanırken değerleri de verilmelidir.
- Değişkenler camelCase olarak tanımlanır.
- İçinde saklayacağı veriyi anımsatacak bir isim verilir.

TECHPRO EDUCATION

Variables - Constants

KURALLAR

- Değişken isimlendirilirken hem harfler hem de sayılar kullanılabilir. Ancak sayılar başa gelmez. Örneğin **sayi1** doğru bir isimlendirmeyken **1sayi** doğru bir isimlendirme değildir.
- Değişken isimlendirilirken alt tire (_ ve $) kullanılabilir. Ancak boşluk ve diğer özel karakterler (?,%,!, ., + vb.) kullanılmaz. Örneğin **ev adresi** ya da **kimlik%no** gibi değişken isimleri kurallara aykırı olduğundan hataya neden olacaktır.
- Değişken isimlendirilirken özel kullanım için ayrılmış olan **if, for, true vb.** ifadeler kullanılmamalıdır.
- İngiliz alfabesinde bulunmayan karakterler **(ç,ğ,ı,ö,ş,ü)** kullanılamaz.

Variables - Constants

✓ ders ✗ 1not ✓ gen_ort

✓ $maas ✗ ilk sayi ✗ son_şekil

✗ gun% ✗ harf-bir ✓ dGunu

TECHPRO EDUCATION

Data types (Veri tipleri)

INTERVIEW

String

var adi="Ali";

Number

var yas=18;

Tam sayı: ± 2^{53}-1

Decimal: ± 2^{1024}

ES 11

BigInt

var x = 100n;

Boolean

var emekli=true;

ES 6

Symbol

var id= Symbol();

Undefined

var isim;

null

var isim = null;

Object

array, object

! typeof

! isNaN

TECHPRO EDUCATION

var

```
var degisken = "Ali";
console.log(degisken);
console.log(typeof(degisken));

var degisken = 50;
console.log(degisken);
console.log(typeof(degisken));

var degisken = true;
console.log(degisken);
console.log(typeof(degisken));
```

Ali
String

50
number

true
boolean

!

var ile değişken tanımlandığında, aynı isimde başka bir değişken tanımlanmasına izin verir.

TECHPRO EDUCATION

let

```
let isim = "Ali";
console.log(isim);
console.log(typeof(isim));

let puan = 50;
console.log(puan);
console.log(typeof(puan));

let emekli = true;
console.log(emekli);
console.log(typeof(emekli));
```

```
Ali
string

50
number

true
boolean
```

TECHPRO EDUCATION

const

```
const degisken = "Ali";
console.log(degisken);
console.log(typeof(degisken));

var degisken = 50;
console.log(degisken);
console.log(typeof(degisken));
```

Ali
string

Burada
hata verir

!

const veya let ile değişken tanımlandığında, aynı scope içinde aynı isimde başka bir değişken tanımlanmasına izin vermez.

TECHPRO EDUCATION

Undefined, empty, null

INTERVIEW

Variable with value
let x = "Hello";

Empty variable
let x = "";

Null variable
let x=null;

undefined
let x;

TECHPRO EDUCATION

Operatörler

Aritmetik Operatörler

Karşılaştırma Operatörleri

Mantıksal Operatörler

Atama Operatörler

TECHPRO EDUCATION

Aritmetik operatörler

Operatör	Sembol	Örnek
Toplama	+	45+6
Çıkarma	-	35-7
Çarpma	*	25*2
Bölme	/	16/5
Üs Alma	**	5**3
Mod Alma	%	12%5
Bir artırma	++	x++
Bir azaltma	--	y--

TECHPRO EDUCATION

Aritmetik operatörler (+)

```
const ekmek = 2;
const yumurta = 30;
const peynir = 40;
const toplamHarcama = ekmek + peynir + yumurta;
console.log("HARCAMA:" + toplamHarcama + " TL");
```

ÇIKTI
HARCAMA: 72 TL

```
const ad = 'Ali';
const soyAd = 'Gel';
console.log(ad + soyAd);
console.log(ad + ' ' + soyAd);
```

ÇIKTI
AliGel
Ali Gel

ÖNEMLİ
+ operatörü ile string birleştirme de gerçekleştirilebilir .

```
const x = 5;
const y = "5";
const birlestir = x + y;
console.log(birlestir);
console.log(x-y);
```

ÇIKTI
55

ÇIKTI
0

TECHPRO EDUCATION

Tip dönüşümleri

```
const para = "100";
console.log(para + 15);
console.log(Number(para) + 15);
```

ÇIKTI
10015

ÇIKTI
115

ÖNEMLİ
Number() fonksiyonu tip çevrimi yapılabilir.

```
const dil = "Javascript";
console.log(Number(dil));
console.log(Number("123abc"));
```

ÇIKTI
NaN

ÇIKTI
NaN

ÖNEMLİ
Number() fonksiyonu harfleri sayıya çeviremeyeceği için NaN (Not a number – Sayı değil) döndürüyor.

```
const para = 5400;
console.log(para+15);
console.log(String(para)+15);
```

ÇIKTI
5415

ÇIKTI
540015

ÖNEMLİ
String() fonksiyonu ile verilen İfadeyi string'e çevirmek mümkündür.

TECHPRO EDUCATION

Tip dönüşümleri

```
const s1 = 5;
const s2 = -7;
const isim = "John";
console.log(Boolean(isim));
console.log(Boolean(s1));
console.log(Boolean(s2));
```

ÇIKTI

```
true
true
true
```

ÖNEMLİ

0, null, undefined, NaN, ve " " Javascript tarafından false olarak kabul edilir.

Diğer değerler Boolean'a çevrildiğinde true olarak kabul edilir.

```
const sifir = 0, nal = null;
const tanimsiz = undefined;
const bos = "", sayiDegil = NaN;

console.log(Boolean(sıfır),Boolean(nal));
console.log(Boolean(tanımsız), Boolean(boş));
console.log(Boolean(sayıDeğil));
```

ÇIKTI

```
false
false
false
```

TECHPRO EDUCATION

İşlem önceliği

Aritmetik işlemler yapılırken bilgisayar operatör önceliğine göre işlem yapar.

( ) Parantez

** Üs alma

* Çarpma

/ Bölme

+ Toplama

- Çıkarma

! Öncelik seviyesi aynıysa soldaki ifadeyi önce yapar.

TECHPRO EDUCATION

İşlem önceliği

8/2*(2+2)	16
30 – 3**2 / 3 + 10	37
16 / 2 * 3 – 2**(4 / 2)	20
(14 * 2 / 7)**2 / 4 + 5	9

TECHPRO EDUCATION

String literals

ES 6

Metinleri daha dinamik bir şekilde birleştirmek için string şablonları (string literals, template literals) kullanabiliriz. Bu birleştirme işlemine de string interpolation denir.

```
let isim = "Ali";
let soyisim = 'Gel';
```

Tek tırnak veya çit tırnak

```
let isim = "Ali";
console.log(`Merhaba ${isim}`);
```

string interpolation bactick ile yapılır

Backtick multiline text tanımlamaya da izin verir.

TECHPRO EDUCATION

Aritmetik operatörler (-)

```
const yil = 2021;
const dogumTarihi = 1980;
const yas = yil - dogumTarihi;
console.log("YAŞ:" + yas);
console.log("YAŞ:" + yil - dogumTarihi);
```

ÇIKTI

YAŞ: 41

NaN

ÖNEMLİ

Ekstra parantez kullanılmaz ise string birleştirme yapmaya çalışır. – den dolayı birleştiremez ve NaN döndürür.

NaN = Not a Number (Sayı değil)

TECHPRO EDUCATION

Aritmetik operatörler (*, **)

```
const pi = 3;
const r = 3;
const alan = pi*r**2;
const çevre = 2*pi*r
console.log(çevre, alan);
console.log("ÇEVRE:" + çevre, "ALAN:"+ alan);
```

ÇIKTI

18 27

ÇEVRE:18 ALAN:27

TECHPRO EDUCATION

Aritmetik operatörler (++, --, %)

```
let a = 3;  let b = ++a;  let c = --a;
console.log(a,b,c);

a += 5;
console.log(a);
```

ÇIKTI
3 4 3

ÇIKTI
8

```
const z = 3;
let k = z++;
console.log(k);
```

HATA
const değişkenin değeri arttırılamaz.

```
let sayi = 123;
console.log("Birler Basamağı:" + sayi%10);
console.log(sayi++);
```

ÇIKTI
Birler Basamağı: 3

ÇIKTI
123

TECHPRO EDUCATION

Karşılaştırma operatörleri

INTERVIEW

Operatör	Sembol	Örnek
Eşittir	== veya ===	Not==75
Eşit değildir	!= veya !==	Not != 45
Büyüktür	>	Yas > 45
Küçüktür	<	Yas < 18
Büyük veya eşit	>=	Ort >=80
Küçük veya eşit	<=	Ort<=90

! Karşılaştırma işlemlerinin sonucu TRUE veya FALSE olur

TECHPRO EDUCATION

Karşılaştırma operatörleri

```
const s1 = 5;
console.log(s1 == 5);        true
console.log(s1 == "5");      true
console.log(s1 === "5");     false
console.log(s1 !== 5);       false
console.log(s1 != "5");      false
console.log(s1 !== "5");     true
```

```
console.log(s1 > 5);         false
console.log(s1 > "4");       true
console.log(s1 >= 5);        true
console.log(s1 > "6");       false
```

ÖNEMLİ:
=== ve !== operatörleri hem veriyi hem de veri tipini kontrol eder.
== ve != operatörleri ise sadece veriyi kontrol ederler.

ÖNEMLİ:
Büyük eşit ve küçük eşit işlemlerinde veri tipi kontrolü yapılmıyor.

TECHPRO EDUCATION

Mantıksal operatörler

Operatör	Sembol	Örnek
And	&&	ort > 50 && ort < 70
Or	||	yas < 16 || yas > 70
Not	!	! (yas > 45)

A	B	A && B	A || B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

A	!A
1	0
0	1

TECHPRO EDUCATION

Mantıksal operatörler

```
let s2 = true;
let s3 = true;
console.log(s2 && true);          true
console.log(s2 && s3);            true
console.log(s2 && s3 && false);   false

s3 = false;
console.log(s2 || s3 || false);   true
console.log(s3 || (s2 && s3) );   false

s3 = null;
console.log(s2 && s3);            null
console.log(s2 || s3);            true
```

ÖNEMLİ:
0, FALSE, NULL, undefined, "" ve NaN dışındaki durumlar TRUE kabul edilir.

TECHPRO EDUCATION

Mantıksal operatörler

```
s2 = "kuş";
s3 = "kedi";
console.log(s2 || s3);      kuş
console.log(s2 && s3);      kedi

s2 = true;
s3 = false;
console.log(!s2);           false
console.log(!s3);           true

s3 = null;
console.log(!s3);           true
```

ÖNEMLİ:

- VEYA işleminde ilk TRUE değerin bulunması yeterlidir. Diğerlerinin kontrolüne gerek yoktur. Bu yüzden, ilk true olan değişkenin değeri döndürülür. Hiçbiri true değilse en sondaki değer döndürülür.

- VE işleminde ilk FALSE değerin bulunması yeterlidir. Diğerlerinin kontrolüne gerek yoktur. Bu sebeple ilk false olan değişkenin değeri döndürülür. Hiçbiri false değilse en sondaki değer döndürülür.

TECHPRO EDUCATION

Mantıksal Operatörler

not1	50
not2	60
not3	70

T `not1 > 30` && `not1 < 70`

F `not2 >= 60` && `not3 != 70`

T `not3 < 100` || `not2 > 80`

F `not1 + not2 < not3` || `not1 < 50`

T `not1 < 30` || `not2 > 50` && `not3 == 70`

TECHPRO EDUCATION

Atama operatörleri

Operatör	Örnek	Açıklama
=	x = y	Sağdaki değişkenin değerini soldakine kopyalar.
+=	x += 1	x= x+1 işlemi gerçekleştirir.
-=	x -= 2	x= x-2 işlemi gerçekleştirir.
*=	x *= 3	x= x*3 işlemi gerçekleştirir.
/=	x /= 4	x= x/4 işlemi gerçekleştirir.
**=	x **= 2	x= x^2 işlemi gerçekleştirir.
%=	x %= 3	x = x mod 3 işlemi gerçekleştirir.
&=	x &= y	x = x VE y işlemi gerçekleştirir.
|=	x |= y	x = x VEYA y işlemi gerçekleştirir.

TECHPRO EDUCATION

Atama operatörleri

```
let x = 22;
let y = 5;
```

`x += 4;`	x=x+4	x=22+4	x=26
`x /= 2;`	x=x/2	x=26/2	x=13
`x %= y;`	x=x%y	x=13%5	x=3
`y = x;`	y=x		y=3
`x *= y;`	x=x*y	x=3*3	x=9
`y **= 2;`	$y=y^2$	$x=3^2$	y=9

```
console.log(x-y);
```

0

TECHPRO EDUCATION

Referans type value type

INTERVIEW

Value Type Variables

```
let score1 = 25
let score2 = score1;
score2 = 21
```

Reference Type Variables

```
let star1 = { size: 3, color: "red" }
let star2 = star1;
star2.color = "blue"
```

TECHPRO EDUCATION

Temel DOM İşlemleri

- document object
- querySelector
- innerText
- innerHtml
- value
- onclick

Document object

Bir web sayfasındaki diğer tüm nesnelerin sahibi olan ana nesneye document object denir. Document nesnesi web sayfamızı temsil eder. Eğer sayfa içindeki bir nesneye erişmek istenilirse document nesnesi kullanılır.

```
document.querySelector(selector);
```

Selectorle eşleşen ilk elemanı seçer

css selectorleri kullanılır. (.menü li)

TECHPRO EDUCATION

Document object

```
document.querySelectorAll(selector);
```

Selectorle eşleşen tüm elemanları seçer ve bir dizi içinde tutar.

css selectorleri kullanılır. (.menü li)

TECHPRO EDUCATION

Document object

- **innerText** — Bir elementin içindeki yazıyı okumak veya değiştirmek için kullanılır
- **innerHTML** — Bir elementin içindeki HTML'i okumak veya değiştirmek için kullanılır
- **classList** — Bir elementin sahip olduğu class ları manipüle etmek için kullanılır
  - **add** — Elemente bir class eklemek için kullanılır
  - **remove** — Elementin sahip olduğu bir class ı kaldırmak için kullanılır

TECHPRO EDUCATION

querySelector

```
<h1 class="name">Ali Gel</h1>
<p class="mesaj"></p>
```

```
let isim = document.querySelector('.name').innerText;
document.querySelector('.mesaj').innerText = `Merhaba ${isim}`;


document.querySelector('.mesaj').innerHTML = `<b>Merhaba ${isim}</b>`;
```

TECHPRO EDUCATION

querySelector

```
<h1 class="name blue">Ali Gel</h1>
<p class="mesaj"></p>
```

```
document.querySelector('.name').classList.add('red');
document.querySelector('.name').classList.remove('blue');
```

TECHPRO EDUCATION

querySelector

value

Bir form elemanının değerini okumak ya da değiştirmek için kullanılır

```
<input type="text" id="name">
<input type="text" id="mesaj">
```

```
let isim = document.querySelector('#name').value;
document.querySelector('#mesaj').value = `Merhaba ${isim}`;
```

TECHPRO EDUCATION

onclick

```
<p onclick="sayHello()">Merhaba Dünya</p>
```

```
function sayHello(){
    alert("Hello World!");
}
```

TECHPRO EDUCATION

DOM

PRACTISE

Textbox lardan alınan iki sayıyı, butona basılınca toplayan ve sonucu bir label içinde gösteren sayfayı yapınız

Sayı 1 | Sayı 2 | Topla

Sonuç:

TECHPRO EDUCATION

Conditional Statements

- if
- if-else
- if elseif
- switch
- ternary
- || and &&
- Optional chaning
- Nullish coalescing

Conditional Statements

- Programlamada program akışını çeşitlendirmek için kullanılır.
- Javascript te bunun için if veya switch blokları kullanılır.

TECHPRO EDUCATION

if

```
if (x>5){

}
```

TECHPRO EDUCATION

if

PRACTISE

Bir Textbox tan alınan 500₺ den büyük fiyatlar için %10 indirim uygulayan programı yapınız.

TECHPRO EDUCATION

if

PRACTISE

```
if (…) {

}
```

null
undefined
NaN
5
"Hello"
true
false
{name: 'John'}
{ }
[5,3,1,5]
[ ]

TECHPRO EDUCATION

if-else

```
if (x>5){

}
else{

}
```

TECHPRO EDUCATION

if-else

PRACTISE

Bir Textbox tan alınan fiyat bilgisi 500 den büyükse %10 indirim uygulayıp sonucu yazan, değilse %5 indirip uygulayıp sonucu yazan programı yapınız.

TECHPRO EDUCATION

DOM

PRACTISE

Dark-light mode switcher uygulaması yapınız.

	Dark Mode	Light Mode
Page background color	black	White
Page text color	white	Black
Button background color	white	Black
Button text color	black	White

TECHPRO EDUCATION

if-elseif-else

```
if (x>5){

}
elseif(x<3){

}
else{

}
```

TECHPRO EDUCATION

if-elseif-else

PRACTISE

Bir Textbox tan alınan fiyat bilgisi 500 den büyükse %10 indirim uygulayıp sonucu yazan, 300 den büyükse %7 indirip uygulayıp sonucu yazan, değilse %5 indirip uygulayıp sonucu yazan programı yapınız.

TECHPRO EDUCATION

switch

```
switch(true){
  case(x>3):
     …
     break;
  case(x>20):
     …
     break;
  case(x>30):
     …
     break;
  default:
     …
}
```

TECHPRO EDUCATION

switch

PRACTISE

Bir Textbox tan alınan fiyat bilgisi
1500 den büyükse %15,
1000 den büyükse %13,
700 den büyükse %10,
500 den büyükse %7,
300 den büyükse %5,
Diğer durumlarda %2 indirim uygulayıp sonucu yazan programı yapınız.

TECHPRO EDUCATION

Ternary

ES 6

if-else bloğunun tek satırda basit bir şekilde yazılmasıdır

mantıksal_ifade ? doğruysa : yanlışsa

```
let maas = 15000;
console.log( maas>13000 ? "Normal" : "Az" );
```

TECHPRO EDUCATION

Ternary

PRACTISE

○ Emekli
◉ Çalışan
Maaş=13500

- Şekildeki gibi form oluşturunuz. Çalışan seçildiğinde tam, emekli ise %10 kesinti uygulayarak maaşı gösteriniz. Bunu ternary kullanarak yapınız.

TECHPRO EDUCATION

|| ve &&

if bloğunun tek satırda basit bir şekilde yazılmasıdır

mantıksal_ifade && doğruysa

mantıksal_ifade || yanlışsa

```
var x = 15
x == 10 && alert("x is 10");
x == 10 || alert("x is not 10");
```

TECHPRO EDUCATION

Optional Chaining

ES 11

Object yapılarında, object in olmayan elemanlarına erişim sağlandığında hata almamak için kullanılır

```
const user = {
    name : 'John’,
    age: 23
}
console.log(user.salary);
```

```
Type error: Cannot read property ‘salary’ of undefined
```

```
const user = {
    name : 'John’,
    age: 23
}
console.log(user?.salary);
```

```
undefined
```

TECHPRO EDUCATION

Nullish Coalescing

ES 11

Sadece Null ve undefined durumlarını kontrol ederek buna göre aksiyon almamızı sağlayan bir operatördür. Eğer ilk parametre null veya undefined değilse ilk parametreyi, yoksa ikinci parametreyi döndürür

```
const user;
alert(user ?? "Anonymous");
```

nullish coalescing

```
const user;
alert(user !== null && user !== undefined ? user : "Anonymous");
```

ternary

TECHPRO EDUCATION

Nullish Coalescing

PRACTISE

Dikdörtgenin alanını bulun bir fonksiyona gönderilen kenar uzunluklarını nullish coalescing ile kontrol ederek default değer atayınız.

TECHPRO EDUCATION

Hesap Makinesi

HOME WORK

Şekildeki gibi bir hesap makinesi tasarlayınız

TECHPRO EDUCATION

Logical Assignment Operators

TASK

ES 12

Nullish Assignment

??=

And Assignment

&&=

Or Assignment

||=

TECHPRO EDUCATION

- for
- while
- do-while

Basic Loops

Döngüler

Tekrarlanan işlemleri daha az kod yazarak gerçekleştirmek için döngüler kullanılır.

for

while

do-while

TECHPRO EDUCATION

For

Check condition
True
Mantıksal ifade doğru olduğu sürece buradaki kodlar sürekli tekrarlanır
False

for

```
for(var i=0; i<10; i++){
    .
    .
}
```

Döngünün başlangıç sayısı

Doğru olduğu sürece döngü tekrar eder

Artış miktarı

- En temel döngü türüdür.
- Tekrar ettirilecek kodun baştan kaç defa tekrar ettirileceği biliniyorsa bu tür döngüler kullanılabilir.

```
for(let i = 1 ; i<= 10 ; i++){

  console.log("Merhaba");

}
```

TECHPRO EDUCATION

for

PRACTISE

Uygula

- Notebook 1500$
- Monitor 500$
- Klavye 10$
- Mouse 5$

Bir ul listesinde ürün isimleri ve fiyat bilgileri bulunmaktadır. Textbox a girilen indirim oranı kadar listeye indirim uygulayan programı yazınız.

TECHPRO EDUCATION

While

Check condition
True
Mantıksal ifade doğru olduğu sürece buradaki kodlar sürekli tekrarlanır
False

while

```
while(a<50){
    …
}
```

Doğru olduğu sürece döngü tekrar eder

- Döngünün ne kadar döneceği döngü bloğunun içindeki kodlara bağlı ise bu durumda while döngüleri kullanılabilir.

TECHPRO EDUCATION

while

PRACTISE

Bir textbox tan alınan sayının, sayı değerini hesaplayıp sonucu bir h3 içinde gösteren programı yapınız.

8291 => 8+2+9+1 = 20

TECHPRO EDUCATION

Do-While

Mantıksal ifade doğru olduğu sürece buradaki kodlar sürekli tekrarlanır
True
Check condition
False

do-while

```
do{
   …
}
while(a<50)
```

Doğru olduğu sürece döngü tekrar eder

- Do while da kontrol sonda olduğu için döngü içindeki kodlar en az bir kere icra edilir.

TECHPRO EDUCATION

Do-while

PRACTISE

Textbox tan alınan sayının faktöriyelini bulan programı yapınız.

TECHPRO EDUCATION

JavaScript

Functions

- Fonksiyon nedir?
- Nasıl çalışır?
- Fonksiyon tanımlama
- Scope
- Hoisting
- Anonymous fonksiyon
- var-let-const

Fonksiyon nedir?

Fonksiyon, programda tekrar tekrar çalıştırılabilecek alt programlara denir.

- Kod tekrarlarının önüne geçer
- Problemler küçük parçalara ayrılarak daha kolay çözülür.
- Parçalara ayrılan sorun, daha fazla insan tarafından aynı anda çözülür.
- Hata ayıklama daha kolay olur

TECHPRO EDUCATION

Nasıl çalışır

TECHPRO EDUCATION

Fonksiyon tanımlama

Function declaration

```
function fonksiyonIsmi(param1, param2, …){
    // işlemler
    // …
    return donenDeger;
}

var x = fonksiyonIsmi(paramValue1, paramValue2, …);
```

Fonksiyona dışarıdan gelen parametreler

Fonksiyonun çağrıldığı yere döndürdüğü değer

Fonksiyonun çağrılması

TECHPRO EDUCATION

Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(){
    console.log(`Benim yaşım ${2022 - 1990}`);
}

yasHasapla();
yasHasapla();
```

Bu fonksiyon hep aynı yaşı gösteriyor. Farklı doğum tarihlerine göre daha modüler hale getirilebilir mi?

TECHPRO EDUCATION

Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(dogumTarihi){
    console.log(`Benim yaşım ${2022 - dogumTarihi}`);
}

yasHasapla(1980);
yasHasapla(2001);
```

Eğer yaş bilgisi ana programda kullanılmak istenirse ne yapılabilir?

TECHPRO EDUCATION

Fonksiyon tanımlama

Yaş hesaplayan bir fonksiyon yazınız.

```
function yasHasapla(dogumTarihi){
    return 2022 - dogumTarihi;
}

var yas = yasHasapla(1980);
if(yas >= 65){
    …
}
```

TECHPRO EDUCATION

Fonksiyon

PRACTISE

Bir sayının asal olup olmadığını bulan fonksiyonu yazınız.

TECHPRO EDUCATION

Homework

- Bir sayının palindrome olup olmadığını bulan programı yapınız

- 12221 -> palindrome
- 4556 -> palindrome değil

Fonksiyon tanımlama

Declaration yöntemi

```
function yasHasapla(dogumTarihi){
    return 2022 - dogumTarihi;
}

yasHesapla();
```

Expression yöntemi

```
const yasHasapla = function(dogumTarihi){
    return 2022 - dogumTarihi;
}

yasHesapla();
```

	Declaration yöntemi	Expression yöntemi
Hoisting	✓	✗
Anonymous Function	✗	✓
Immediately Invoke Function	✗	✓
Callback Function	✗	✓

TECHPRO EDUCATION

Fonksiyon tanımlama

ES 6

Arrow function => Expression yönteminin biraz daha sade halidir

```
const yasHasapla = (dogumTarihi) => 2022-dogumTarihi;


let yas = yasHesapla(2000);
```

Eğer fonksiyon içeriği tek satır ise {} kullanmadan doğrudan değer döndürülebilir.

```
const yasHasapla = (dogumTarihi) => {
 if(!dogumTarihi) return;
 return 2022-dogumTarihi;
}

let yas = yasHesapla(2000);
```

Eğer fonksiyon içeriği birden fazla satır ise, {} ve return kullanılmalıdır.

TECHPRO EDUCATION

Fonksiyon tanımlama

Arrow function

```
const topla = (a, b) => a+b;
```

Expression

```
const topla = function(a, b){
    return a+b;
}
```

Declaration

```
function topla(a, b){
    return a+b;
}
```

TECHPRO EDUCATION

Fonksiyon

PRACTISE

Ürün fiyatı, kdv oranı ve indirim oranı parametre olarak alıp son fiyatı hesaplayan fonksiyonu arrow function ile yazınız. Ürün fiyatı 1500 üzerinde ise extra 10% indirim uygulanacaktır.

İndirim uygulanmamış ise kdv hesaba katılmayacaktır.

TECHPRO EDUCATION

Scope

INTERVIEW

Değişken ve sabitlerin yaşam alanına scope denir

function scope

Fonksiyon içinde tanımlanan değişkenler sadece fonksiyon içinde geçerlidir.

global scope

Her yerden erişilebilen değişkenler, en tepede tanımlanır.

block scope

Sadece tanımlandığı blok içinde (if, for, while...) geçerlidir.

TECHPRO EDUCATION

Function Scope

```
const fonk1 = function () {
    let sayi1 = 22;
    console.log(sayi1);
};

fonk1();
console.log(++sayi1);
```

```
Uncaught ReferenceError: sayi1 is not defined
    at Script snippet %235:6:9
```

sayi1 değişkeni fonk1 içinde tanımlandığı için, sadece fonk1 fonksiyonu içinde geçerlidir. Fonksiyon dışından erişilmeye çalışıldığında hata alınır.

TECHPRO EDUCATION

Global Scope

```
let sayi = 5;
const fonk = function (){
    sayi = 10;
    console.log(`Fonk. İçi: ${sayi}`);
}

fonk();
console.log(`Fonk. Dışı: ${++sayi}`);
```

```
Fonk. İçi: 10
Fonk. Dışı: 11
```

sayi değişkeni fonk dışında tanımlandığı için, her yerden erişilebilir. Örnekte hem fonk isimli fonksiyondan hem de dışındaki kodlar üzerinden erişilebilmektedir.

TECHPRO EDUCATION

Global vs Function Scope

```
let sayi = 3;
const fonk = function () {
    let sayi = 7;
    console.log(`Fonk. İçi: ${sayi}`);
};

fonk();
console.log(`Fonk. Dışı: ${++sayi}`);
```

```
Fonk. İçi: 7
Fonk. Dışı: 4
```

Hem global he de function scope da aynı isimde değişken tanımlandığında bunları farklı değişkenler olarak algılanır.

TECHPRO EDUCATION

Block Scope

ES 6

```
const fonk = function (sayi) {
    if (sayi < 0){
        let negatif = true;
    }
    console.log(negatif);
};
fonk(-4);
```

```
Uncaught ReferenceError: negatif is not defined
    at fonk (a.html:15:21)
    at a.html:17:7
```

Block scope değişkenler sadece bulundukları { } blok içinde geçerlidir.
Blok dışından erişim yapılamaz.
Block scope sadece let ve const için geçerlidir.

TECHPRO EDUCATION

Hoisting

INTERVIEW

Javascript te default bir davranış olarak tanımlanan değişkenler ve function lar bulundukları scope ta default olarak en tepeye alınır. Böylece tanımlama yapılmadan önce kullanılabilirler. Buna hoisting denir.

```
yasHasapla(2000);
console.log(a);
var a = 5;

function yasHasapla(dogumTarihi){
    return 2022 - dogumTarihi;
}
```

Sadece declaration hoist edilir, initializations edilmez

let ve const ile yapılan tanımlamalar da hoist edilmektedir ancak tanımlamadan önce kullanıma izin vermediklerinden dolayı, kullanılırsa ReferenceError vermektedirler

TECHPRO EDUCATION

Anonymous Function

Fonksiyonların herhangi bir isim vermeden tanımlanmasına anonymous function denir

```
const yasHasapla = function(dogumTarihi){
    return 2022 - dogumTarihi;
}

yasHesapla();
```

TECHPRO EDUCATION

var vs let vs const

	var	let	const
Değiştirilebilir	✓	✓	✗
Global scope	✓	✓	✓
Function scope	✓	✓	✓
Block scope	✗	✓	✓
Hoisting	✓	✗	✗
Re-declaration	✓	✗	✗

TECHPRO EDUCATION

Scope

PRACTISE

«Oyuna Başla» butonuna basıldığında 1-100 arasında rasgele sayı tutulsun. Textbox içine tahmini sayı girilip «tahmin et» butonuna basıldığında tutulan sayıyla girilen sayıyı kontrol edip doğru tahmin olup olmadığını ve büyüklük küçüklük durumunu gösteren programı yapınız.

İpucu

```
let sayi = Math.floor(Math.random() * (max - min + 1)) + min
```

TECHPRO EDUCATION

Event Handlers

- Event handler nedir?
- Nasıl tanımlanır?
- Handler çeşitleri

Event handler

Bir programda olabilecek olayların yakalanması ve olay gerçekleştiğinde istenilen kodların çalıştırılması işlemine event handling, bu işlemi yapan kodlara da event handler denir.

Tanımlama (1.Yöntem)

```
<button onclick="sayHello()">Tıkla</button>
```

event listener

event handler

```
function sayHello(e){
    alert("Hello");
}
```

TECHPRO EDUCATION

Tanımlama (2.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").onclick = sayHello;

function sayHello(e){
    alert("Hello");
}
```

TECHPRO EDUCATION

Tanımlama (3.Yöntem)

```
<button id="btnMsg">Tıkla</button>
```

event listener

event handler

```
document.querySelector("#btnMsg").addEventListener("click", function(e){
    alert("Hello");
});
```

```
document.querySelector("#btnMsg").addEventListener("click", (e) => {
    alert("Hello");
});
```

arrow function

TECHPRO EDUCATION

Function

PRACTISE

Butona basıldığında, iki textbox içinde 100 lük sistemde girilen notları kontrol edip, ortalamasını bulan ve bunu harf sistemine çeviren fonksiyonu yazınız.

Aralık	Not
[90-100]	A
[80-90)	B
[70-80)	C
[50-70)	D
[0-50)	F

TECHPRO EDUCATION

Objects & Arrays

- Object nedir?
- Object tanımlama
- Array nedir?
- Array tanımlama
- Array of objects
- Array Methods
- Array iterations & search
  - for-in
  - forEach
  - map
  - filter
  - reduce
  - every
  - some

Object nedir?

- Programımızda bir çalışana ait çeşitli özellikleri saklamak istiyoruz. Bu durumda, değişken kullanırsak;

```
const firstName = 'Ali';
const lastName= 'Gel';
const birthDate = '1.1.1990';
const salary = 7000;
```

Eğer çalışanın çok daha fazla bilgisi saklanacksa?

Object yapıları kullanılarak bir nesneye ait özellikler ve değerleri gruplandırılabilir.

TECHPRO EDUCATION

Object nedir?

INTERVIEW

Javascript te basit object tanımı { } ile yapılır.

```
const employee = {
    firstName : 'Ali',
    lastName : 'Gel',
    birthDate : '1.1.1990',
    salary : 7000
}
```

```
console.log(employee);

console.log(employee.salary);
```

TECHPRO EDUCATION

Object methods

Nesnedeki alanları ve/veya değerleri bir dizi içinde döndürmek (iteration) ve üzerlerinde işlem yapmak için kullanılır

entries

Nesnedeki alanlar ve değerleri bir dizi içinde döndürülür

keys

Nesnedeki alanlar bir dizi içinde döndürülür

values

Nesnedeki değerler bir dizi içinde döndürülür

TECHPRO EDUCATION

Objects

PRACTISE

- Bir object içinde key e göre arama yapıp value döndüren uygulamayı yapınız

TECHPRO EDUCATION

Array nedir?

- Programımızda araba isimlerini saklamak istiyoruz. Bu durumda, değişken kullanırsak;

```
const araba1 = 'Mercedes’;
const araba2 = 'TOFAŞ’;
const araba3 = 'Anadol’;
const araba4 = 'Ferrari';
```

Saklanacak yüzlerce araba ismi olsaydı?

Diziler, bir veri yapısı (data structure) olup bir veya çok daha fazla veriyi saklamak için kullanılır.

TECHPRO EDUCATION

Array tanımlama

INTERVIEW

Javascript te basit dizi tanımı [ ] ile yapılır.

```
let kurslar = ["Javascript","HTML","CSS","Bootstrap"];
```

Dizi tanımlama için bir diğer yöntem

```
let kurslar = new Array("Javascript","HTML","CSS","Bootstrap");
```

TECHPRO EDUCATION

Dizilerden veri okumak

```
var kurslar = ["Javascript","HTML","CSS","Bootstrap"];
```

Dizi elemanlarına erişim

```
kurslar[1];
```

Dizi uzunluğu

```
kurslar.length
```

```
console.log(kurslar[1]);
console.log(kurslar[kurslar.length-1]);
```

Dizi elemanları sıralı olarak belleğe kaydedilirler.

Sıra numaraları 0 ile başlar ve dizinin eleman sayısının bir eksiğine (length-1) kadar devam eder.

TECHPRO EDUCATION

Dizilere veri yazmak

```
const kurslar = ["Javascript","HTML","CSS","Bootstrap"];


kurslar[1]="React";
kurslar[0]="Java";

console.log(kurslar);

kurslar = ["Ali","Veli];
```

```
Uncaught TypeError: Assignment to constant variable.
    at main.js:23
```

Dizi const ile tanımlanmış olsa da dizinin elemanlarını değiştirebildik. Çünkü non-primitive veri türlerinin içerikleri değiştirilebilir. Ancak, tamamen bir başka dizi ile değiştirilemez.

TECHPRO EDUCATION

Diziler

- Bir dizi içindeki en büyük sayıyı bulan programı yazınız.

```
var sayilar =
[12,56,14,67,89,33,22];
```

TECHPRO EDUCATION

Diziler

PRACTISE

const fiyatlar = [123, 5666, 126, 67];

Şeklindeki dizi elemanlarının değerini %20 artıran fonksiyonu yazınız.

TECHPRO EDUCATION

Array of Objects

```
const arabalar =[{
    marka:'Mercedes',
    model:'S500',
    renk:'Bej',
    vites:'Otomatik'
},
{
    marka:'BMW',
    model:'i8',
    renk:'Mavi',
    vites:'Otomatik’
},
…
]
```

Elemenaları object olan dizilere array of object denir.

TECHPRO EDUCATION

Array Methods

Dizi elemanlarına erişmek, bunları manipüle etmek veya iterasyon yapmak için çeşitli hazır methodlar bulunmaktadır

mutator methods

Dizi içeriğinin değiştirilmesini sağlarlar

access methods

Dizi elemanlarına erişmek için kullanılırlar

iterator methods

Dizi elemanları döngü yoluyla dolaşmak için kullanılır

TECHPRO EDUCATION

Mutator methods

RETURN	ARRAY	METHOD	MUTATED ARRAY
4	🐶 🐶 🐶	.push(🐱)	🐶 🐶 🐶 🐱
🐶	🐱 🐱 🐶	.pop()	🐱 🐱
🐱	🐱 🐶 🐶	.shift()	🐶 🐶
4	🐶 🐶 🐶	.unshift(🐱)	🐱 🐶 🐶 🐶
🐱 🐶 🐼	🐶 🐼 🐱	.sort()	🐱 🐶 🐼
🐱 🐼 🐶	🐶 🐼 🐱	.reverse()	🐱 🐼 🐶
🐼 🐱	🐶 🐼 🐱	.splice(1, 2, 🦆)	🐶 🦆

TECHPRO EDUCATION

Mutator methodlar

```
const meyveler = ["Elma", "Armut", "Muz", "Kivi"];

meyveler.pop();          Kivi silindi
meyveler.push('Ananas');  Sona Ananas eklendi
meyveler.shift();        Elma silindi
meyveler.unshift('Çilek');  Başa Çilek eklendi
meyveler.reverse();      Tersine çevrildi
meyveler.sort();         Harfe göre sıralandı
meyveler.splice(1,0,'Kiraz');  1.İndex e Kiraz eklendi
meyveler.splice(3, 1 ,'Kayısı');  3.İndex e Kayısı eklendi, Muz silindi
```

TECHPRO EDUCATION

Mutator methodlar

HOME WORK

Şekildeki gibi bir uygulama yapınız

Add

Sort | Reverse | Remove Last | Remove First | Remove All | Mix

- Ankara
- Istanbul
- İzmir

TECHPRO EDUCATION

Access methods

RETURN	ARRAY	METHOD
🐶🐶🐶🐱🐱	🐶🐶🐶	.concat(🐱🐱)
true	🐱🐱🐶	.includes(🐶)
1	🐱🐶🐶	.indexOf(🐶)
2	🐶🐶🐶	.lastIndexOf (🐶)
" 🐶 - 🐼 - 🐱 "	🐶🐼🐱	.join("-")
" 🐶 , 🐼 , 🐱 "	🐶🐼🐱	.toString()
🐶🐼	🐶🐼🐱	.slice(0, 2)

TECHPRO EDUCATION

Access methods

```
const hayvanlar = ['fil','kuş','deve','fare','kedi’];
hayvanlar.includes('fil');        true
hayvanlar.includes(3);            false
hayvanlar.includes("at"));        false

console.log(hayvanlar.join());        fil,kuş,deve,fare,kedi
console.log(hayvanlar.join(""));      filkuşdevefarekedi
console.log(hayvanlar.join("-"));     Fil-kuş-deve-fare-kedi

console.log(hayvanlar.slice(2));      ["deve", "fare", "kedi"]
console.log(hayvanlar.slice(2, 4));   ["deve", "fare"]
console.log(hayvanlar.slice(1, 5));   ["kuş", "deve", "fare", "kedi"]

console.log(hayvanlar.toString());    fil,kuş,deve,fare,kedi
```

TECHPRO EDUCATION

Access methods

```
const hayvanlar = ['fil','kuş','deve','kuş','kedi','kuş'];

console.log(hayvanlar.indexOf('kuş'));
console.log(hayvanlar.lastIndexOf('kuş'));
console.log(hayvanlar.indexOf('at'));
```

```
1
5
-1
```

```
const harfler = ['a', 'b', 'c'];
const rakamlar = [1, 2, 3];
const birlesik = harfler.concat(rakamlar, 4, [5,6]);
console.log(birlesik);
```

```
['a', 'b', 'c', 1, 2, 3, 4, 5, 6];
```

TECHPRO EDUCATION

Iteration - for

PRACTISE

ÖRNEK:
Bir dizideki sayıların toplamını hesaplayan uygulamayı For döngüsü ile yazınız.

```
const rakamlar = [-5, 15, 22, -4, 45, 78, -25];

let toplam = 0;

for (let i = 0 ; i < rakamlar.length ; i++) {
  toplam += rakamlar[i];
}

console.log(toplam);
```

TECHPRO EDUCATION

Iteration - for

PRACTISE

Bir dizideki pozitif ve negatif sayıların toplamını hesaplayıp, bu toplamların farkını bulan uygulamayı for döngüsü ile yazınız.

hesapla adında bir fonksiyonda hesaplamayı yapınız.

TECHPRO EDUCATION

Iteration – for in

For döngüsünün kısaltılmış halidir. (sayaç ve koşul kullanmaya gerek yok.) Özellikle dizi ve nesnelerin iterasyonu için geliştirilmiştir. Değişken içine otomatik olarak elemanın indis numarası gelir.

```
for ( degisken in diziAdi) {
    // Döngü içi
}
```

TECHPRO EDUCATION

Iteration – for in

```
const adlar = ["Ahmet", "Can","Mustafa","Ayşe","Elif"];
const soyAdlar= ["Öztürk", "Yılmaz", "Arı", "Çalı", "Yazı"];

const birlestir = (x,y) => {
  let adVeSoyadlar = [];
  for (let i in x) {
    adVeSoyadlar[i] = `${x[i]} ${y[i]}`;
  }
  return adVeSoyadlar;
};

console.log(birlestir(adlar, soyAdlar));
```

ÖRNEK:
İki ayrı dizideki eşleşen indis elemanları birleştirerek ayrı bir diziye saklayan uygulamayı FOR IN ile yazınız.

TECHPRO EDUCATION

Iteration – for of

ES 6

FOR OF, bir çok veri yapısı üzerinde çalışabildiği için FOR IN'e göre daha geniş kullanım alanına sahiptir. Diziler, Stringler v.b bir çok veri yapısında kullanılabilir. Değişken içine otomatik olarak elemanın değeri gelir.

```
for ( degisken of diziAdi) {
    // Döngü içi
}
```

TECHPRO EDUCATION

Iteration – for of

```
let arabalar = ["BMW", "Volvo", "Mini"];
let yazi = "";

for (let x of arabalar) {
  yazi += x + " ";
}
console.log(yazi);
```

ÖRNEK:
Dizideki elemanları birleştirerek tek bir String haline getiren uygulamayı FOR OF ile yazınız.

TECHPRO EDUCATION

For - of

PRACTISE

Harici ülke listesi datasını alarak bir select içinde ülkeleri gösteriniz.

TECHPRO EDUCATION

Iteration – for each

FOR OF döngüsüne benzemektedir. Sadece diziler üzerinde çalışır. For of' tan sonra popülerliğini yitirmeye başlamıştır.

```
dizi.forEach( (item, index)=> {
    // Döngü içi
});
```

TECHPRO EDUCATION

Iteration – for each

```
const dizi = [-5, 24, -13, 7];
const yeniDizi = [];

dizi.forEach((item, index) => {
     yeniDizi[index] = item * 5;
});
console.log(yeniDizi);
```

ÖRNEK:
Belirtilen dizinin her bir elamanının 5 katını alarak ayrı bir dizide saklayan uygulamayı forEach() metodu ile yazınız.

TECHPRO EDUCATION

Iteration – map

Mevcut bir dizi içinde iterasyon yapıp, orijinal dizinin kopyası üzerinde elemanlarının değerlerini değiştirmeyi sağlar. Değişiklik yapılmış diziyi geri verir. Map methodu orijinal diziyi değiştirmez.

```
dizi.map( (item, index) => item * 5);
```

TECHPRO EDUCATION

Iteration – map

```
const isimler = ["Mustafa", "Murat", "Ahmet", "Mustafa", "Ayşe",
"canan"];

const buyukIsimler = isimler.map((x) => x.toUpperCase());

console.log(buyukIsimler);
```

ÖRNEK:
Bir dizideki tüm isimleri BÜYÜK harfe dönüştüren uygulamayı yazınız.

TECHPRO EDUCATION

Iteration – map

tlFiyatlar dizisindeki fiyatların Euro ve dolar karşılıklarını hesaplatarak yeni dizlere kaydediniz.

```
const euro = 21.10;
const dolar = 20.35;
const tlFiyatlar = [100, 150, 100, 50, 80];
```

TECHPRO EDUCATION

Iteration

ÖRNEK:

tlFiyatlar dizidekisindeki ürünlere fiyatı 100 TL den fazla olanlara %10 zam, 100 TL den az olanlara ise %15 zam yapılmak isteniyor.

Ayrıca, zamlı olan yeni değerleri yeni bir diziye saklamak istiyoruz.

Iteration – filter

Mevcut bir dizi içinde iterasyon yapıp, orijinal dizinin kopyası üzerinde filtreleme yapar. Filtrelenmiş diziyi geri verir. Filter methodu orijinal diziyi değiştirmez.

```
dizi.filter( (item, index) => item > 5);
```

TECHPRO EDUCATION

Iteration – filter

ÖRNEK:

Koordinatlar dizisindeki negatif koordinatları alıp yeni bir dizide saklayan uygulamayı filter() ile yapınız.

```
const koordinatlar = [-100, 150, -32, 43, -20];
const negatifKoordinatlar = koordinatlar.filter((x) => x < 0);
console.log(negatifKoordinatlar);
```

TECHPRO EDUCATION

Iteration – filter

```
const bireyler = ["Mustafa", "Murat", "Ahmet", "mustafa", "Ayşe",
"Canan"];

const filtrele = function (harf) {
    const h = harf.toUpperCase();
    const filtrelenmiş = bireyler.filter((t) => t.startsWith(h));
    return filtrelenmiş;
};

console.log(filtrele("m"));
console.log(filtrele("A"));
```

ÖRNEK:
Bireyler dizisindeki kişilerden adı "Belirtilen" harf ile başlayanları seçerek ayrı bir diziye saklayan uygulamayı yazınız.

TECHPRO EDUCATION

Iteration – filter

ÖRNEK:

Koordinatlar dizisindeki negatif koordinatları seçerek bunları pozitife çevirip alt alta konsola bastıran uygulamayı yazınız.

```
const koordinatlar = [-100, 150, -32, 43, -20];
koordinatlar.filter(x => x < 0)
.map(t => t * -1)
.forEach(y => console.log(y));
```

Iterasyon methodlarının peşpeşe kullanılmasına pipeline denir

TECHPRO EDUCATION

Filter

PRACTISE

Tablo içine yerleştirilen ülkelerden biri seçildiğinde seçilen ülkeye ait bilgileri gösteren uygulamayı filter ile yapınız.

TECHPRO EDUCATION

Iteration – reduce

Mevcut bir dizi içinde iterasyon yapıp, yapılan işlemlere göre tek bir hesaplanmış değer döndürür.

```
dizi.reduce( (total, x) => total + x, 0 );
```

Her bir işlem sonucunu saklayacak olan değer

Her bir dizi elamanı

İlk değer

TECHPRO EDUCATION

Iteration – reduce

```
const koordinatlar = [-100, 150, -32, 43, -20];

const toplam = koordinatlar.reduce( (tp, koordinat ) => {
   console.log(`iterasyon ${tp} ${koordinat}`);
   return tp + koordinat;
},0);
console.log(toplam);
```

ÖRNEK:
Koordinatlar dizisindeki değerlerin toplamını, ara değerleri de göstererek konsola bastıran uygulamayı reduce() ile yazınız.

TECHPRO EDUCATION

Iteration – reduce

ÖRNEK:

Koordinatlar dizisindeki değerlerin ortalamasını hesaplayarak konsola bastıran uygulamayı reduce() ile yazınız.

```
const koordinatlar = [-100, 150, -32, 43, -20];

let ortalama = koordinatlar.reduce((toplam, koordinat) => toplam + koordinat );
ortalama /=  koordinatlar.length;

console.log("Koordinatların Ortalaması:" + ortalama);
```

TECHPRO EDUCATION

Iteration – reduce

ÖRNEK:

Bir Firma, 3000 TL den az olan maaşlara %10 zam yapmak istiyor ve zam yapılan bu kişilere toplam kaç TL ödeneceğini bilmek istiyor. İlgili programı yazınız.

```
const maaslar = [3000, 2891, 3500, 4200, 7000, 2500];

const zamliToplam = maaslar
.filter((maas) => maas < 3000)
.map((maas) => maas * 1.1)
.reduce((toplam, maas) => toplam + maas);

console.log(zamliToplam.toFixed(2));
```

TECHPRO EDUCATION

Reduce

PRACTISE

Ülke listesinin altına ülke toplam yüzölçümlerini reduce kullanarak yazdırınız.

TECHPRO EDUCATION

Iteration – every

Mevcut bir dizi içinde tüm elemanların istenilen kriteri sağlayıp sağlamadığını döndürür. Sağlıyorsa true, değilse false döner.

```
dizi.every( (item) => item > 33 );
```

TECHPRO EDUCATION

Iteration – some

Mevcut bir dizi içinde en az bir elemenın istenilen kriteri sağlayıp sağlamadığını döndürür. Sağlıyorsa true, değilse false döner.

```
dizi.some( (item) => item > 33 );
```

TECHPRO EDUCATION

JavaScript

DOM
(Document Object Model)

- DOM nedir?
- DOM yapısı
- DOM erişim
- DOM güncelleme

DOM Nedir?

DOM (Document Object Model), dokümanlarının stillerinin, yapısının, içeriğinin erişilmesine ve güncellenmesine olanak sağlayan bir nesne modelidir.

Objects	Properties	Styles	Methods	Events
HTML elemanları	HTML elemanlarının özellikleri (attribute)	HTML elemanlarının CSS özellikleri	HTML elemanlarına erişmeyi sağlayan methodlar	HTML elemanları üzerinde gerçekleşen olaylar

TECHPRO EDUCATION

DOM Yapısı

- Bir web sayfası yüklendiğinde tarayıcı, ilgili HTML sayfasının DOM'unu oluşturur.
- DOM ağaç yapısı ise HTML dosyası içerisinde bulunan tüm nesnelerin (node) hiyerarşik bir biçimde gösterimidir.

DOM Yapısı

```html
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>DOM</title>
</head>
<body>
  <h1>DOM Temelleri</h1>
  <section>
    <img src="./img/js_logo.png" alt="Javascript">
    <p>Javascript öğrenmek için link
       <a href="www.javascript.org">Javascript</a>
    </p>
    <button>Tıkla</button>
  </section>
  <script src="main.js"></script>
</body>
</html>
```

TECHPRO EDUCATION

DOM'a Erişim

INTERVIEW

Method	Açıklama
getElementById	Bir elemanı id sinden bulur
getElementsByTagName	Bir elemanı Tag (Etiket) adına göre bulur
getElementsByClassName	Bir elemanı Class (Sınıf) adına göre bulur
querySelector	Bir elemanı id, class, özellik, tür ve değere göre seçmek için kullanılır. Eşleşen ilk elemanı seçer.
querySelectorAll	Bir elemanı id, class, özellik, tür ve değere göre seçmek için kullanılır. Eşleşen elemanların listesini döndürür.

TECHPRO EDUCATION

DOM erişim

PRACTISE

Adı	Puanı
Ali	70
Veli	30
Ayşe	5
Fatma	90

Puanı belli bir değerden aşağıda olan kişilerin rengini kırmızı yapan programı yapınız.

TECHPRO EDUCATION

DOM'a Erişim

Method	Açıklama
closest	En yakın elemanı seçer, yukarı doğru çalışır
children	Bir elementin çocuklarını seçer
firstElementChild, lastElementChild	Bir elementin ilk çocuğunu veya son çocuğunu seçer.
parentElement	Bir elemanın parent ını seçmek için kullanılır.
nextElementSibling, previousElementSibling	Bir elemanın önceki veya sonraki kardeşini (aynı seviye) seçmek için kullanılır.

TECHPRO EDUCATION

DOM erişim

PRACTISE

Her satırın sonundaki Sil butonlarına basıldığında satır başındaki ismi alarak uyarı veren programı yazınız.

TECHPRO EDUCATION

DOM güncelleme

Method	Açıklama
innerText	Bir elemanın içindeki metni getirir veya içine metin yerleştirir.
innerHTML	Bir elemanın içindeki html i getirir veya içine HTML yerleştirir.
textContent	innerText ile aynıdır. Tek farklı stil özelliklerinden dolayı gizlenmiş olan text leri de getirir

TECHPRO EDUCATION

DOM Styles

Method	Açıklama
classList	Bir elemanın class listesine erişim sağlar.
add("class")	classList e bir class ekler
remove("class")	classList teki bir class ı siler
toggle("class")	classList te yoksa class ı ekler, varsa class ı siler
contains("class")	classList te bir class ın olup olmadığını kontrol eder.
replace("class1", "class2")	classList te class1 i siler, yerine class2 yi koyar

TECHPRO EDUCATION

DOM Styles

PRACTISE

Bir tablonun satırlarına bir kere tıklandığında seçili hale getiren tekrar aynı satıra tıkladığında seçili olmaktan çıkaran programı yapınız.

İsim	Puan	#
Ayşe	80	Sil
Ali	50	Sil
Veli	30	Sil
Fatma	20	Sil
Aslı	55	Sil

TECHPRO EDUCATION

DOM Attributes

Method	Açıklama
getAttribute("attribute")	Bir elemanın attribute unu getirir.
setAttribute("attribute", "value")	Bir elemana attribute ekler/değiştirir
removeAttribute("attribute")	Bir elemanın attribute unu siler
attributes	Bir elemanın tüm attribute larını getirir.

TECHPRO EDUCATION

DOM Attributes

PRACTISE

Default olarak disabled olan bir buton, textbox içine bir şey yazılınca aktif, silinince pasif olacak.

TECHPRO EDUCATION

DOM güncelleme

Method	Açıklama
createElement("h1")	Yeni bir element oluşturur. Document objesinin bir methodudur.
appendChild(element)	Bir elemanın içinde en sona başka bir eleman ekler
append(element, "metin")	Bir elemanın içinde en sona başka bir elemanı veya text i ekler. Tek seferde birden fazla ekleme yapabilir.
prepend(element, "metin")	Bir elemanın içinde en başa başka bir elemanı veya text i ekler. Tek seferde birden fazla ekleme yapabilir.
remove()	Bir elemanı siler

TECHPRO EDUCATION

DOM

PRACTISE

Adı	Puanı	
Fatma	90	Sil
Ali	70	Sil
Veli	30	Sil
Ayşe	5	Sil

Tablodaki satırların sonundaki sil butonuna basıldığında satırı silen programı yapınız.

TECHPRO EDUCATION

DOM

HOME WORK

Type name... | Point | + Add

#	Name	Point	
1	Ali Gel	80	
2	Ayşe Koş	70	
3	Mehmet Git	90	
	Average	70	

Dizi kullanılmadan yapılacak

TECHPRO EDUCATION

JavaScript

Advanced JavaScript

- Error handling
- Debugging
- Timeout , Interval
- Web storage
- Rest & spread operatör
- Destructuring
- Syncrounous & Asyncronous
- Callbacks
- Modules
- Promises

Error handling

INTERVIEW

Hata olması muhtemel kodlarda, hataların programın akışını engellememesini sağlamak ve kullanıcıya daha kontrollü ve anlaşılır uyarılar vermek için yapılan işlemdir.

Genellikle kullanıcıdan bilgi alırken veya bir API dan bilgi çekerken kullanılırlar.

Try – Catch - Finaly

TECHPRO EDUCATION

Error handling

```
try{

}
```

Hata olması beklenen kodlar try bloğu içine yazılır.

```
catch(ex){

}
```

Try bloğu içinde bir hata olduğunda program akışı catch bloğuna geçer ve hata ile ilgili bilgi ex object içinde bulunur.

```
finally {

}
```

Hata olsa da olmasa da çalıştırılan bir bloktur. Kullanılması zorunlu değildir.

TECHPRO EDUCATION

Error handling

PRACTISE

Bir login formu oluşturup, form handling yapınız

TECHPRO EDUCATION

Debugging

INTERVIEW

Programdaki hataların ayıklanması işlemine debugging denir. Debugging için chrome developer tools kullanılabileceği gibi Vscode gibi programlardaki debuger da kullanılabilir.

TECHPRO EDUCATION

Debugging

1

Debugging bölümü açılır

2

Run and Debug butonuna basılır

3

Hangi ortamda çalıştırılacağı seçilir

TECHPRO EDUCATION

Debugging

- Program işleyişini belli bir noktada kesmek için breakpoint ler kullanılır.
- Program akışı o noktaya geldiğinde kesilecek ve programın adım adım çalıştırılabilmesi için debuging başlayacaktır.
- Breakpoint eklemek için satır numaralarının sol tarafındaki boşluğa tıklanır. Kırmızı noktalar breakpoint leri temsil eder.

TECHPRO EDUCATION

Debugging

Değişken takip alanı

Continue

Step Over

Step Into

Step Out

Restart

Stop

RUN AND DEBUG

No Configurations

VARIABLES

Local: test1

this: Window

Script

x: 20

Global

WATCH

26-scope.html

04-javascript > 26-scope.html > html > body

```
12
13      <script>
14          let x = 10;
15          var y = 30;
16
17          function test1(){
18              x=20;
19              console.log(x); //20
20          }
```

TECHPRO EDUCATION

Timeout

INTERVIEW

Gecikmeli olarak çalışması istenilen kodlar, timeout oluşturularak ve belli bir gecikme zamanı belirlenerek çalıştırılabilir.

```
setTimeout( ()=> {
    // Gecikmeli
    // çalışacak
    // kodlar
    // burada
    // olacak
}, 500);
```

setTimeout methodu ile bir zamanlayıcı oluşturulur.

Gecikme zamanı ms cinsinden tanımlanır

clearTimeout() methodu ile timer durdurulabilir.

TECHPRO EDUCATION

Timeout

PRACTISE

Bir metin kutusunda anlık arama yapılırken data sorgusunun geciktirilmesi ve böylece gereksiz sorguların engellenip performansın artırılması hedefleniyor.

Bu şekilde çalışan bir metin kutusu tasarlayınız.

TECHPRO EDUCATION

Interval

Belli periyotlarla çalışması istenilen kodlar, interval oluşturularak ve belli bir zaman belirlenerek çalıştırılabilir.

```
setInterval( ()=> {
    // Belli aralıklarla
    // çalışacak
    // kodlar
    // burada
    // olacak
}, 500);
```

setInterval methodu ile bir zamanlayıcı oluşturulur.

Period ms cinsinden tanımlanır

clearInterval() methodu ile timer durdurulabilir.

TECHPRO EDUCATION

Timeout

PRACTISE

11:17

Çalışan digital saat uygulaması yapınız

TECHPRO EDUCATION

Storing data

Kullanıcı tarafında (browser) data saklayabilmek için iki yol bulunmakadır.

Storege API

Cookies

TECHPRO EDUCATION

Storing data

	Cookie	Storage API
Kapasite	4KB / cookie	5MB / web site
Yaşam süresi	Expire olana veya silinene kadar	Browser kapanana veya silinene kadar
Güvenlik	Çok güvenli değil	Daha güvenli
Erişim	Her yerden erişim	Sadece mevcut siteden erişim
Kota	Kotasız	5MB
Senkronizasyon	Yok	Var
Erişim hızı	Yavaş	Hızlı

TECHPRO EDUCATION

Web Storage API

INTERVIEW

Kullanıcı tarafında data saklayabilmek için çerezler(cookie) veya HTML5 ile birlikte gelen Storage API kullanılır. Veri, browser üzerinde saklanır.

localStorage

Veri, kullanıcı veya app silene kadar saklanır.
Veriye farklı sekmeler ve pencerelerden ulaşılır.

sessionStorage

Veri, oturum bazlı saklanır.
Tarayıcı kapatıldığında veri silinir.
Veriye farklı sekmeler ve pencerelerden ulaşılamaz

TECHPRO EDUCATION

Web Storage API

localStorage ve sessionStorage de data manipülasyonu için kullanılan methodlar

setItem	getItem	removeItem
Veri saklamak için	Veriyi almak için	Veriyi silmek için

```
localStorage.setItem("counter", 1);
```

TECHPRO EDUCATION

Web Storage API

PRACTISE

Sayfa her ziyaret edildiğinde, ziyaretçiye siteye kaçıncı kez girdiğini söyleyen programı yazınız.

TECHPRO EDUCATION

Web Storage API

PRACTISE

Sayfaya girildiğinde alt taraftan offcanvas şeklinde açılacak bir privacy policy uyarı penceresi gösteriniz. Accept butonuna basılınca kapansın ve bir daha sormasın.

TECHPRO EDUCATION

Rest & Spread Operator

INTERVIEW

ES 6

Rest ve Spread operatörleri 3 nokta ile ifade edilirler ancak birbirinden farklıdırlar.

REST

Farklı değişkenlerden gelen verileri bir dizi içinde toplar

SPREAD

Dizi veya object içindeki verileri ayrı değişkenlere açar.

TECHPRO EDUCATION

Rest Operator

Örnek: Toplama işlemi yapan bir fonksiyon yapınız. Ancak bu fonksiyona farklı miktarda sayı parametre olarak gönderilebilmelidir.

```
function topla(...sayilar){
     return sayilar.reduce( (toplam, x)=> toplam + x);
}

console.log(topla(3,5,6));
console.log(topla(13,5,16,3,56,76,1));
```

Gelen tüm parametreleri sayilar isimli dizi içinde alacaktır.

TECHPRO EDUCATION

Rest Operator

Örnek: Öğrencinin adını ve soyadı ile belirsiz sayıda notu parametre olarak alan ve notların ortalamasını bulup, öğrenci adı ile birlikte ortalamayı döndüren fonksiyonu yazınız.

```
function ortalama(ad, soyad, ...notlar){
     let ort = notlar.reduce( (toplam, x)=> toplam + x);
     return `${ad} ${soyad} ortalaması: ${ort/notlar.length}`
}

console.log(ortalama("Ali","Gel",3,5,6));
console.log(topla(("Ayşe","Koş", 13,5,16,3,56,76,1));
```

İlk iki parametre haricide geri kalan tüm parametreleri bir dizi içine alacaktır

TECHPRO EDUCATION

Spread Operator

Örnek: Bir dizi içindeki sayıları ayrı ayrı parametre olarak topla fonksiyonuna gönderen kodları yazınız.

```
function topla(s1, s2, s3){
     return s1 + s2+ s3;
}

const sayilar = [12,56,17];

console.log(topla(...sayilar));
```

Sayılar dizisini genişletip her bir elemanını ayrı bir değişken haline getirdi.

TECHPRO EDUCATION

Spread Operator

```
let numberStore = [0, 1, 2];
let newNumber = 12;

numberStore = [...numberStore, newNumber, 15];
```

numberStore dizisi başka bir dizi içinde açılarak ve yeni elemanlar eklenerek güncellendi.

TECHPRO EDUCATION

Spread Operator

```
const student = {
    firstName: "Ali",
    lastName: "Gel",
    no:"234",
    phone:"23423423423"
};

let newStudent = {...student, firstName:"Veli"};
console.log(student, newStudent);
```

student nesnesini genişletip, firstName özelliğinin değeri değiştirip yeni bir nesne oluşturuyor.

TECHPRO EDUCATION

Destructuring

INTERVIEW

ES 6

Bir nesnenin değerlerinin değişkenlere açılması işlemine destructuring denir.

```
let { variable1, variable2 } = object;
```

TECHPRO EDUCATION

Destructuring

```
const student = {
    firstName: "Ali",
    lastName: "Gel",
    no:"234",
    phone:"23423423423"
};

const { firstName, lastName } = student;
console.log(firstName);
```

Student nesnesi içindeki, "Ali" ve "Gel" değerlerini ayrı ayrı firstName ve lastName değişkenlerine aktarmış

TECHPRO EDUCATION

Destructuring

```
const student = {
    firstName: "Ali",
    lastName: "Gel",
    no:"234",
    phone:"23423423423"
};

function sayHello({firstName, lastName}){
    console.log(`Hello ${firstName} ${lastName}`);
}

sayHello(student);
```

Fonksiyon parametre olarak nesne alırken, eğer süslü parantezler içinde nesnenin key leri yazılırsa yine destructuring yapılmış olur.

TECHPRO EDUCATION

Modules

INTERVIEW

Moduller, JS kodlarını bölerek farklı farklı dosyalardan çağırmak ve böylece büyük projelerde kod karmaşasının önüne geçmek için kullanılan bir yöntemdir.

export	import
Dışarı aktarma	İçeri alma

Dışarı aktarılan kodlar içeri alınabilir.

TECHPRO EDUCATION

Modules

app.js

```
import {showAlert, showConfirm}
from "./message.js";


showAlert();
```

message.js

```
export const showAlert = () => {

}


export const showConfirm = () => {

}
```

TECHPRO EDUCATION

Modules

app.js

```
import {showAlert, showConfirm}
from "./message.js";


showAlert();
```

message.js

```
const showAlert = () => {

}


const showConfirm = () => {

}

export {showAlert, showConfirm};
```

TECHPRO EDUCATION

Modules

app.js

```
import showAlert, {showConfirm}
from "./message.js";


showAlert();
```

message.js

```
const showAlert = () => {

}


const showConfirm = () => {

}

export default showAlert;
export {showConfirm};
```

TECHPRO EDUCATION

Module

PRACTISE

Aşağıdaki methodları içeren bir modüle yazınız.

sOrderASC()
sOrderDESC()
nOrderASC()
nOrderDESC()

TECHPRO EDUCATION

Synchronous vs Asynchronous

Synchronous

11sn

TECHPRO EDUCATION

Synchronous vs Asynchronous

Asynchronous

5sn

TECHPRO EDUCATION

Synchronous vs Asynchronous

Javscript senkron bir yapıda çalışır. Yani kodlar sırayla çalıştırılmaktadır. Ancak bazı durumlarda asenkron moda geçerek gereksiz beklemelerin önüne geçer.

- Event listeners
- setTimeout
- setInterval
- Promise
- fetch
- XmlHttpRequest
- Stream
- Worker

TECHPRO EDUCATION

Synchronous vs Asynchronous

```
function getData(){
    setTimeout( ()=> {
        console.log("ok2");
    },500);
}

console.log("ok1");
getData();
console.log("ok3");
```

ok1
ok3
ok2

TECHPRO EDUCATION

Callbacks

INTERVIEW

Asenkron çalışan fonksiyonlardan veri döndürmek için callback fonksiyonlar kullanılır.

Callbacks

```
.
.
.
siparisVer( (sonuc)=> {



});
```

```
function siparisVer(callback){
    ...
    callback("lahmacun");
}
```

TECHPRO EDUCATION

Callbacks

```
siparisVer( (sonuc)=> {
    console.log(sonuc);
});
console.log("ayakkabı aldım");
```

```
function siparisVer(callback){
    console.log("sipariş verdim");
    setTimeout( ()=> {
        callback("lahmacun hazır");
    },3000)
}
```

TECHPRO EDUCATION

Callback Chains

Asenkron yapıların birbirini bekleyere çalıştırmak için callback fonksiyonlardan oluşan bir zincir yapılır. Buna callback chain denir.

```
siparisVer( (sonucSiparis)=> {
    yemekYe( (sonucYemek)=>{
        sinemadaFilmIzle( (sonucFilm)=> {

        });
    });
});
console.log("ayakkabı aldım");
```

TECHPRO EDUCATION

Promises

INTERVIEW

Promise yapıları, callback ler gibi asenkron sonuçları işleyebilmek için kullanılan, callback lerdeki bazı sorunların da ortadan kaldırıldığı daha güvenilir yapılardır.

İç içe callback çağrıldığında oluşan callback sorunları

- Callback'in beklenenden erken çağırılması
- Callback'in hiç çağırılmaması
- Callback'in beklenilenden az veya çok çağırılması
- Gerekli parametreleri doğru bir şekilde alamaması
- Hataların yutulması

TECHPRO EDUCATION

Promises

Hamburger siparişi verdik. Bize söz verdiler. (Promise)

Indicator, hamburger hazır olunca haber verecek. Bekliyoruz. (Pending)

Hamburger yerine bir sorun oluşursa (rejected)

Hamburger başarılı bir şekilde gelirse (resolved)

TECHPRO EDUCATION

Promises

Defining Promise Function

```
const soz1 = ()=> {
    return new Promise( (resolve, reject)=>{
      if (herseyYolunda) {
        resolve('İşlem tamam!');
      } else {
        reject('Bir sıkıntı var...');
      }
    })
}
```

Calling Promise Function

```
soz1().then( cevap=>{
  console.log(cevap);
}).catch( hata=>{
  console.log(hata);
})
```

Resolve ile dönüşler then tarafında, reject ile dönüşler catch tarafında yakalanır.

TECHPRO EDUCATION

Promise

PRACTISE

Javascript te bulunan confirm method unu async hale getiriniz.

TECHPRO EDUCATION

Promise Chain

Calling Promise Chain

```
soz1().then( (cevap1)=>{
  return soz2(cevap1)
})
.then(cevap2=> {
    return soz3(cevap2);
})
.then(cevap3=> {
    console.log(cevap3);
})
.catch( hata=>{
  console.log(hata);
})
```

- Promise zincirleri peş peşe çalışması gereken asenkron işlemler için kullanılır.
- Bir asenkron işlemin cevabı geldiğinde bir sonraki asenkron işlem çağrılır.

TECHPRO EDUCATION

Awaiting Promises

Promise yapılarında senkron kod üretmek ve then yapılarındaki kod okunabilirliği sorununu gidermek için await keyword u kullanılabilir.

TECHPRO EDUCATION

Awaiting Promises

```
const f1 = () => {
    return new Promise( (resolve, reject)=> {
        setTimeout(() => resolve("hello2"), 1000);
    })
}

const f2 = () => {
    return new Promise( (resolve, reject)=> {
        setTimeout(() => resolve("hello4"), 2000);
    })
}
```

```
const init = async () => {
    console.log("hello1");
    const msg1 = await f1();
    console.log(msg1);
    console.log("hello3");
    const msg2 = await f2();
    console.log(msg2);
}
```

TECHPRO EDUCATION

Tea break...

10:00

Start | Stop | Reset | mins: 10 | secs: 0 | type: Tea

Breaktime for PowerPoint by Flow Simulation Ltd.

Pin controls when stopped ☑