



BATCH : B45-46-71
LESSON : **Javascript**
DATE : 08.04.2022
SUBJECT : **Advanced JS**



 techproeducation.com

 info@techproeducation.com

 +1 (917) 768-7466



JavaScript

Advanced JavaScript

- › Error handling
- › Debugging
- › Timeout , Interval
- › Web storage
- › Rest & spread operator
- › Destructuring
- › Callbacks
- › Fetch
- › Modules



Error handling

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



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.



Error handling

PRACTISE

Login

Username:

Password:

Login

[Register](#)

Bir login formu oluřturup, form handling yapınız



Debugging



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



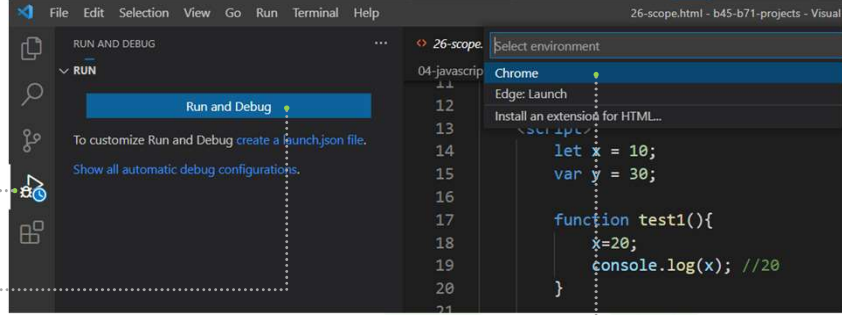
Debugging

1

Debugging bölümü açılır

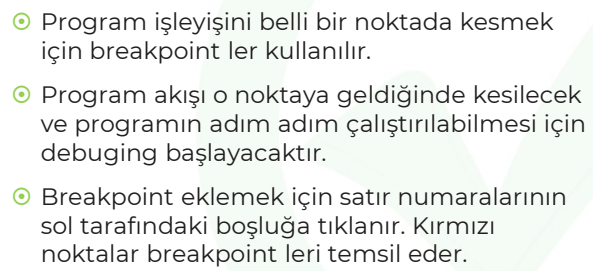
2

Run and Debug butonuna basılır



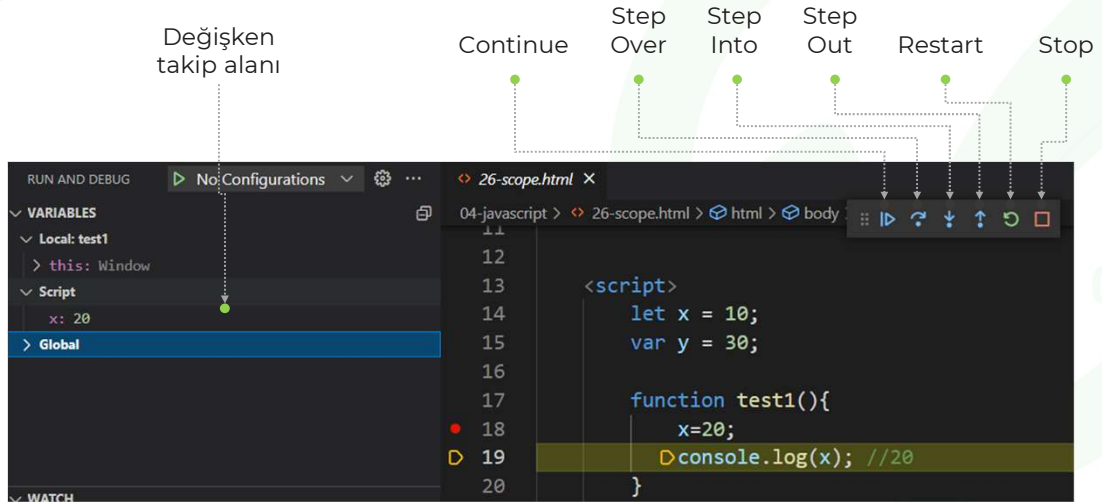
3

Hangi ortamda çalıştırılacağı seçilir





Debugging



Continue: Varsa bir sonraki breakpoint e atlar, yoksa programı sonuna kadar çalıştırır.

Step Over: Fonksiyonlara girmeden adım adım programı ilerletir.

Step Into: Fonksiyonlara da girerek programı adım adım ilerletir.

Step Out: Bir fonksiyondan çıkmayı sağlar

Restart: En baştan programı tekrar başlatır

Stop: Debugging modu durdurur.