

Neden JavaScript

M Akif Eyller
Mlgros E-commerce
Şubat 2020

Kullandığım Diller

- 1971 Mainframe Fortran
- 1975 Terminal Fortran ⇨ Pascal
- 1985 IBM PC Turbo Pascal
- 1990 Macintosh Object Pascal
- 2002 Diz üstü Java
- 2018 Mobil cihazlar JavaScript

1971 Oda içi -- Mainframe



FORTRAN

Program listing

```
1      K=1
2      6      IF (K.EQ.11) GO TO 8
3          READ,I,J
4          IF (J.GT.I) GO TO 65
5          GO TO 66
6      65     WRITE(6,6002)J,I
7      6002   FORMAT(' ',I3,' IS GREATER THAN ',I3)
8          K=K+1
9          GO TO 6
10     66     WRITE(6,6001)I,J
11     6001   FORMAT(' ',I3,' IS GREATER THAN ',I3)
12         K=K+1
13         GO TO 6
14     8      CALL EXIT
15         END
```

1985 Mac & PC



Neden JavaScript

- her platformda çalışıyor
- daha kolay ve daha eğlenceli
- web tasarımında vazgeçilmez

<https://eyler.blogspot.com/2018/06/neden-javascript.html>

Name, Value, Type

```
let user = 'Ali'           //string
let count = 234           //number
let time = new Date()     //object

user.length               //property
3

user.toLowerCase()        //method
"ali"

count.toFixed(2)
"234.00"

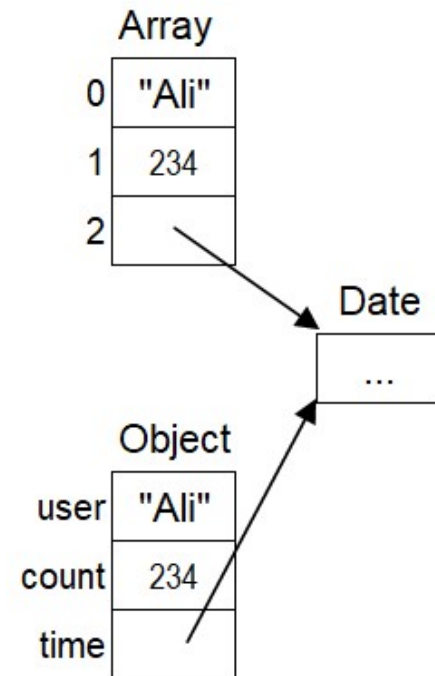
time.getFullYear()
2020

time.toString()
"Wed Feb 05 2020"
```

Array and Object

```
a = [user, count, time]
["Ali", 234, Wed Feb 05 2020 09:13:17]
a.length //3
a[1] //234
a.indexOf(234) //1
a.filter(x => typeof x == 'string')
["Ali"]

p = {user, count, time}
{user: "Ali", count: 234, time: Wed ...}
p.user // "Ali"
p.count //234
```



Array, String, JSON

a

```
["Ali", 234, Wed Feb 05 2020 09:13:17]
```

```
s = a.join(',')
```

```
"Ali, 234, Wed Feb 05 2020 09:13:17"
```

```
b = s.split(',')
```

```
["Ali", "234", "Wed Feb 05 2020 09:13:17"]
```

```
a == b //false
```

```
b.join(',') == s //true
```

```
JSON.stringify(a)
```

```
' ["Ali",234,"2020-02-05T06:13:17.026Z"] '
```

Numbers are double

```
0.1 + 0.2 == 0.3 //false - also in C and in Java
0.1 + 0.2 - 0.3 //5.551115123125783e-17
Number.EPSILON //2.220446049250313e-16
0.1 + 0.2 - 0.3 < Number.EPSILON //true
(0.1+0.2).toFixed(1) //"0.3" as desired
Number.MAX_SAFE_INTEGER //9007199254740991
Number.MAX_SAFE_INTEGER+2 //9007199254740992
"9007199254740991".length //16
```

- * Do all monetary operations in cents (integer)
- * Never compare numbers for equality (use epsilon)
- * Do not print numbers unformatted

Callback

```
L = console.log //simplest callback
```

```
navigator.geolocation //returns Geolocation
```

```
navigator.geolocation.getCurrentPosition(L)
```

```
navigator.geolocation.getCurrentPosition(x => p=x.coords)  
//p.latitude, p.longitude
```

```
navigator.getBattery() //returns Promise
```

```
navigator.getBattery().then(L)
```

```
navigator.getBattery().then(x => b=x)  
//b.level, b.dischargingTime
```

Short-circuit

```
t2s = (t, u) =>  
  t < 1 ? '' : t.toPrecision(2) + ' ' + u
```

```
timeString = (t) =>  
  t2s(t/86400, 'days')  
  || t2s(t/3600, 'hours')  
  || t2s(t/60, 'minutes')  
  || t2s(t, 'seconds')  
  || (1000*t).toFixed() + ' msec'
```

```
timeString(543) // "9.1 minutes"
```

```
timeString(54345) // "15 hours"
```

fetch(url).then(cb)

<https://maeyler.github.io/JS/hard/Fetch%20remote.html>

```
u = "https://ipinfo.io/json"
```

```
fetch(u).then(r => r.blob()).then(x => b=x)
```

```
//b.size, b.type
```

```
fetch(u).then(r => r.json()).then(x => p=x)
```

```
//p.ip, p.city, ...
```

```
fetch(u).then(r => r.text()).then(t => a=t.split('\n'))
```

```
//a is Array of lines in remote text file
```

Array of Objects

```
u = "https://maeyler.github.io/JS/data/Students.txt"
fetch(u).then(r => r.text()).then(t => a=t.split('\n'))
//input data: id, name, gpa separated with TAB on each line
s = a[0]
s.split('\t')
[id,name,gpa] = s.split('\t')
D = a.map(s => {
  [id,name,gpa] = s.split('\t')
  return {id,name,gpa}
})
//output data: {id, name, gpa}
D.filter(d => d.name == "GÜL")
D.filter(d => d.gpa > 3.98)
```

Tools

Chrome



GitHub



GitKraken



VS Code – Live Server extension

JavaScript Örnekleri: <https://maeyler.github.io/JS/>

Eloquent JavaScript: <http://eloquentjavascript.net/>

What Web Can Do Today: <https://whatwebcando.today/>

Simple Cloud Storage

Forms to Sheets:

<https://maeyler.github.io/JS/data/>

MVC application:

<https://github.com/taniarascia/mvc/>

MVC Todo App:

<https://maeyler.github.io/mvc/>

<https://maeyler.github.io/>

MVC Todo Form

User (Unique id -- not verified)

Topic (Todo item to be added)

Marks (C:completed, D:deleted)