

JavaScript dióhéjban

Siki Zoltán



Böngészőkben futtatható szkript nyelv
A Netscape-nél fejlesztették ki (Brendan Eich)
HTML-be ágyazható
A helyi gép erőforrásait **korlátozottan** érheti el
Prototípus alapú objektum orientált nyelv
Böngészőben bekövetkezett események kezelése (pl. klikk)

Bár a JavaScriptet szabványosították, a különböző böngésző programokban eltérő implementációk készültek (IE!)

A böngészőkhöz kapcsolódó fejlesztőkörnyezetek

FireFox – **FireBug** bővítmény (külön telepítendő), ezt használjuk

Chrome – beépített fejlesztőeszközök

Internet Explorer – beépített fejlesztőeszközök

FireFox – FireBug

Átváltások – Mozilla Firefox

File Edit View History Bookmarks Tools Help

Objektum Orientált ... x Átváltások x +

www.geod.bme.hu/tantargyak/web/atvalt.html

Search

FireBug ikon

Hossz mértékegységek

m	Bécsi öl	Láb	Yard	Mérföld	Tengeri mérföld	Töröl
						Átszámít

Terület mértékegységek

m ²	öl ²	Hektár	Kat. hold	Acre	Töröl
					Átszámít

Szög mértékegységek

Fok	Fok-perc-mp	Újfok	Vonás (6400)	Radián	Töröl
					Átszámít

Firebug 2.0.12

1 Total Firebug (Minimized)

Panel activation status:

Console: On

Net: On

Cookies: On

Script: On

Használat:

Töltse ki a megfelelő mértékegység mezőjét és nyomja meg az Enter gombot vagy kattintson az átszámít gombra.

<http://www.geod.bme.hu/tantargyak/web/atvalt.html>

FireFox – FireBug

The screenshot shows the Mozilla Firefox browser window titled "Átváltások – Mozilla Firefox". The address bar shows the URL "www.geod.bme.hu/tantargyak/web/atvalt.html". The page content includes three sections: "Hossz mértékegységek" (Length units), "Terület mértékegységek" (Area units), and "Szög mértékegységek" (Angle units). Each section has a table of units and input fields for conversion. The "Script" tab in the FireBug developer tool is selected, showing the JavaScript code of the page. The "atvalt.html" file is highlighted in the file list. The "Watch" tab in FireBug shows the current state of the page's variables.

Hossz mértékegységek

m	Bécsi öl	Láb	Yard	Mérföld	Tengeri mérföld	Töröl
						Átszámít

Terület mértékegységek

m ²	öl ²	Hektár	Kat. hold	Acre	Töröl
					Átszámít

Szög mértékegységek

Fok	Fok-perc-mmp	Ízfok	Verés (6400)	Radián	Töröl
					Átszámít

Használat:
Töltse ki a megfelelő mértékegység mezőjét és nyomja meg az Enter gombot vagy kattintson az átszámít gombra.

Az oldalhoz betöltött JavaScriptek

A JavaScript kód nem rejthető el az azavatott szemek elől.

Önálló FireBug ablak

Script

all atvalt.html

```
1 <html><head>
2
3 <title>Átváltások</title>
4 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-2">
5 <script language="JavaScript" type="text/javascript">
6
```

Watch

New watch expression...

window Window atvalt.html

ffclipboard Object { setText=setText() }

fix fix(number, decimals)

Változók

Egyszerű változók szám, szöveg és logikai érték tárolására
Különböző típusú változók közötti műveleteknél automatikus
típus konverzió

Az utasítás után írt ; (pontosvessző) nem kötelező, ha csak
egy utasítás van a sorban

Interaktív JavaScript parancsok bevitele

```
> a = 5;
5
> b=6;
6
> a+b
11
> s = 'alma';
"alma"
> t = "körte";
"körte"
> s + " " + t
"alma körte"
> c = 12.34
12.34
```

Kis és nagybetűk
különböznek!

Mi lesz x és y értéke?

```
v1 = '112';
v2 = 76;
x = v1 + v2;
y = v2 + v1;
```

Logikai konstansok:

```
true
false
v1 == v2;
```

Mi az eredmény?

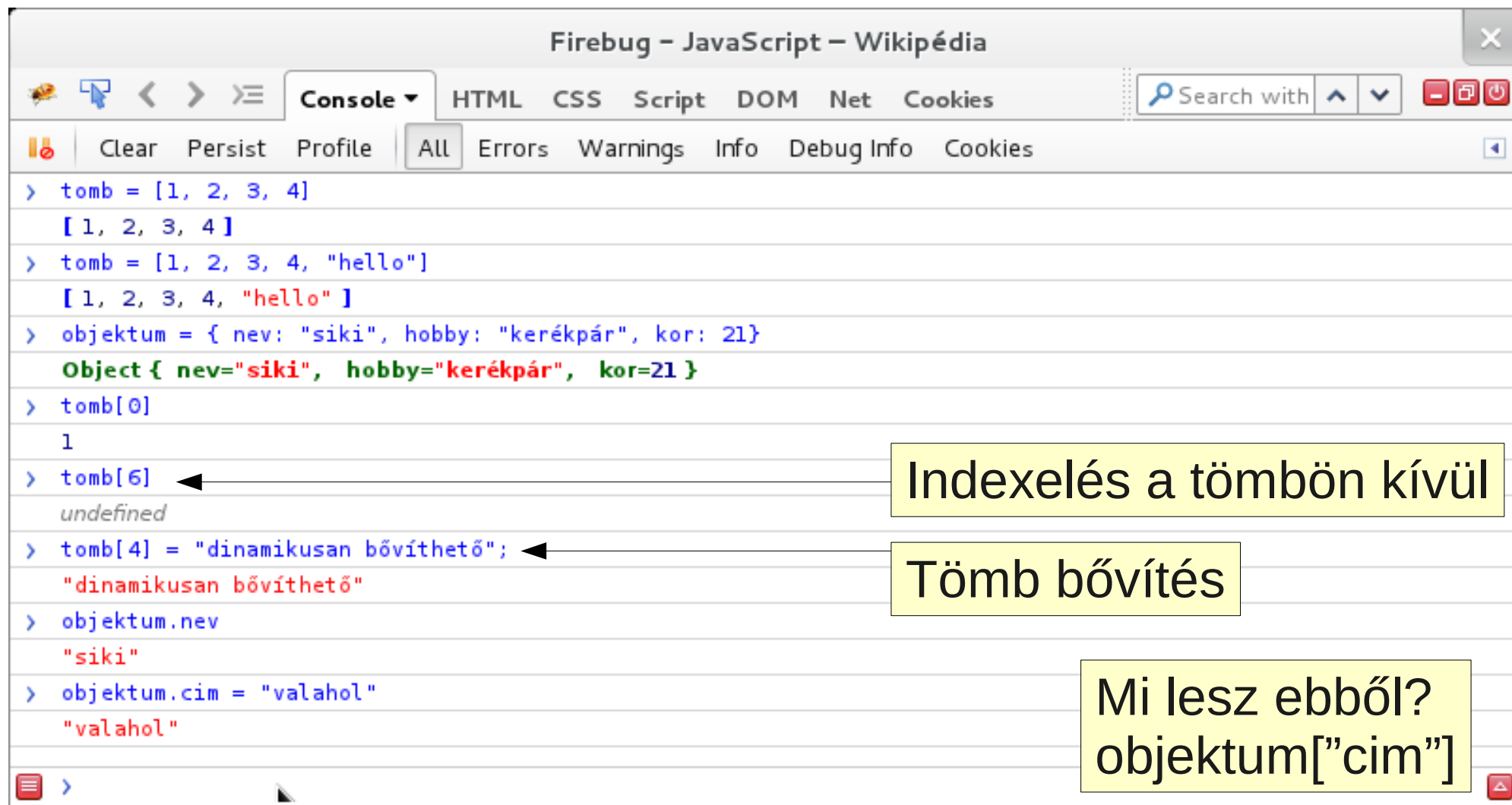
```
6 / 0
```

Összetett adattípusok

Tömb és objektum

Tömb – rendezett adattípus, indexelés 0-tól

Objektum – asszociatív tömb (függvényeket is tartalmazhat)



The screenshot shows the Firebug JavaScript console with the following code and output:

```
> tomb = [1, 2, 3, 4]
[1, 2, 3, 4]
> tomb = [1, 2, 3, 4, "hello"]
[1, 2, 3, 4, "hello"]
> objektum = { nev: "siki", hobby: "kerékpár", kor: 21 }
Object { nev="siki", hobby="kerékpár", kor=21 }
> tomb[0]
1
> tomb[6]
undefined
> tomb[4] = "dinamikusan bővíthető";
"dinamikusan bővíthető"
> objektum.nev
"siki"
> objektum.cim = "valahol"
"valahol"
```

Annotations on the right side of the console:

- Indexelés a tömbön kívül (points to `tomb[6]`)
- Tömb bővítés (points to `tomb[4] = "dinamikusan bővíthető";`)
- Mi lesz ebből? objektum["cim"] (points to `objektum.cim = "valahol"`)

Fontosabb beépített objektumok

Math, Date, RegExp, Function

Függvények (igen, a JavaScriptben a függvények is objektumok)

Szabályos kifejezések

Dátum és időpont kezelés

Matematikai függvények, konstansok

Firebug - Date/Time functions | JavaScript Tutorial

Console HTML CSS Script DOM Net Cookies

Clear Persist Profile All Errors Warnings Info Debug Info Cookies

```
> Math.sin(0.5)
0.479425538604203
> Math.PI
3.141592653589793
> Date()
"Tue Sep 08 2015 22:15:15 GMT+0200 (CEST)"
> d = new Date()
Date { Tue Sep 08 2015 22:15:22 GMT+0200 (CEST) }
> d.getDay()
2
> r = /^a.*x$/
RegExp /^a.*x$/
> "asterix".match(r)
[ "asterix" ]
> "alma".match(r)
null
```

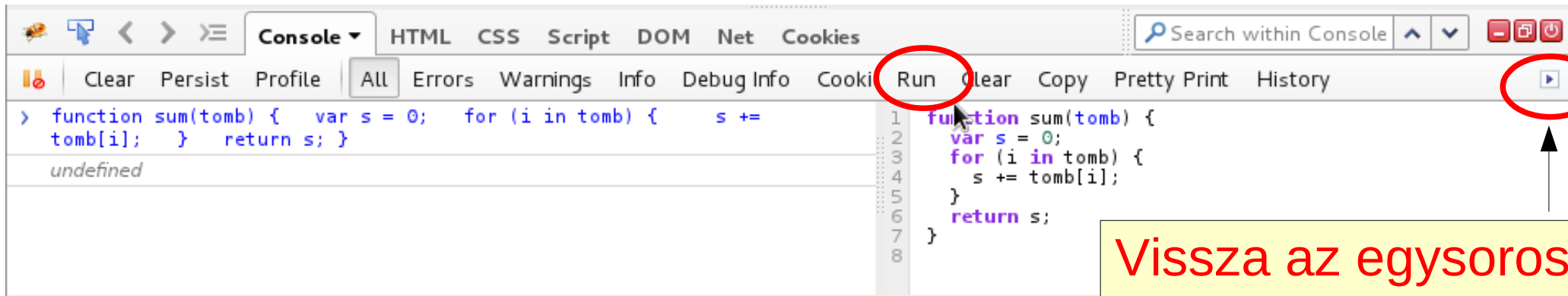
Math statikus objektum

Új objektum létrehozása

Többsoros adatbevitel

Ennek mi lesz az eredménye:
r.test("asterix")

Függvények

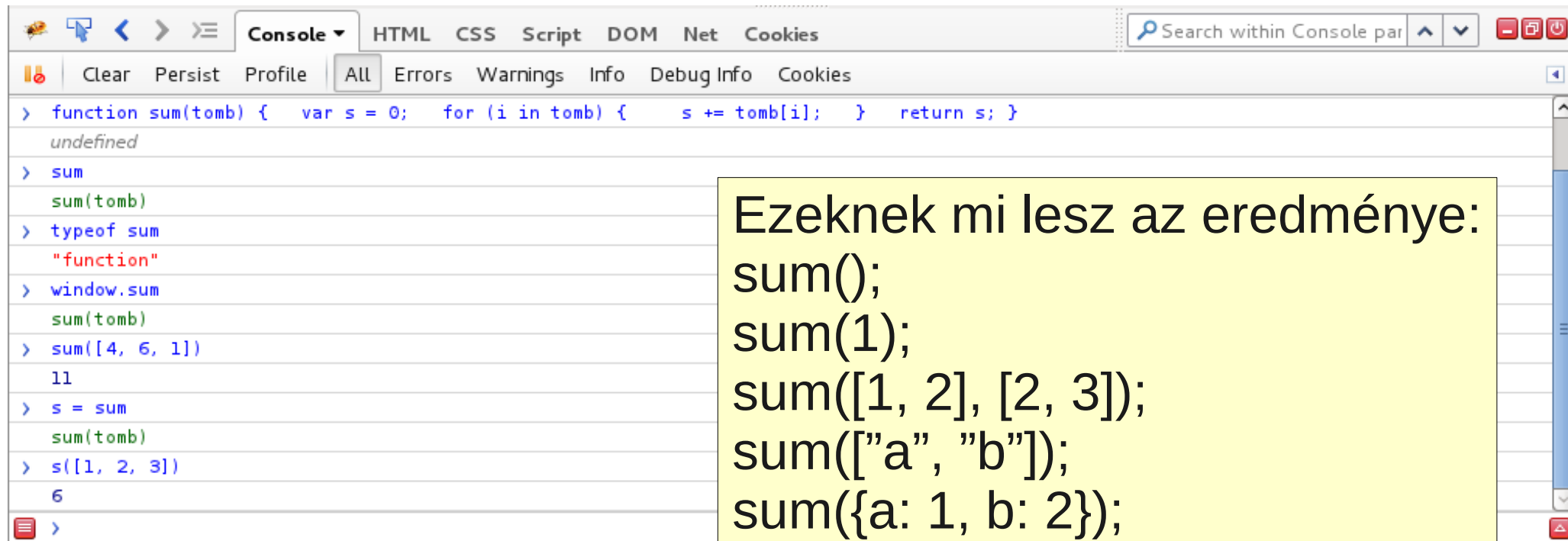


The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays a function definition: `> function sum(tomb) { var s = 0; for (i in tomb) { s += tomb[i]; } return s; }`. The output is `undefined`. The 'Run' button is circled in red. Another red circle is around a button in the top right corner of the console panel.

```
> function sum(tomb) { var s = 0; for (i in tomb) { s += tomb[i]; } return s; }
```

```
1 function sum(tomb) {  
2   var s = 0;  
3   for (i in tomb) {  
4     s += tomb[i];  
5   }  
6   return s;  
7 }  
8
```

Vissza az egysoros adatbevitelhez



The screenshot shows the same web browser's developer console with the 'Console' tab selected. The console displays several function calls and their results: `> function sum(tomb) { var s = 0; for (i in tomb) { s += tomb[i]; } return s; }` (output: `undefined`), `> sum` (output: `sum(tomb)`), `> typeof sum` (output: `"function"`), `> window.sum` (output: `sum(tomb)`), `> sum([4, 6, 1])` (output: `11`), `> s = sum` (output: `sum(tomb)`), and `> s([1, 2, 3])` (output: `6`).

```
> function sum(tomb) { var s = 0; for (i in tomb) { s += tomb[i]; } return s; }
```

```
undefined
```

```
> sum
```

```
sum(tomb)
```

```
> typeof sum
```

```
"function"
```

```
> window.sum
```

```
sum(tomb)
```

```
> sum([4, 6, 1])
```

```
11
```

```
> s = sum
```

```
sum(tomb)
```

```
> s([1, 2, 3])
```

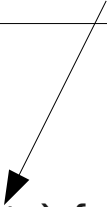
```
6
```

Ezeknek mi lesz az eredménye:
`sum();`
`sum(1);`
`sum([1, 2], [2, 3]);`
`sum(["a", "b"]);`
`sum({a: 1, b: 2});`
Miért?

Függvények 2

A sum függvény másik megfogalmazása
Változó hosszúságú paraméterlista kezelése

```
function sum() {  
  var s = 0;  
  var i;  
  for (i in arguments) {  
    if (typeof arguments[i] == 'number') {  
      s += arguments[i];  
    }  
  }  
  return s;  
}
```



Rekurzió

```
function fact(n) {  
  var f;  
  if (n == 0) {  
    return 1;  
  }  
  return n * fact(n-1);  
}
```

Hasznos további beépített függvények:

parseInt("123")

parseFloat("1.123")

isNaN(NaN)

isFinite(Infinity)

eval(1+2)

Feladat:

Írj egy függvényt az első n természetes szám összegére

Form és JavaScript

<http://www.agt.bme.hu/tantargyak/web/formpelda.htm>

Oldal forrás
illetve
FireBug szkript fül

HTML FORM mintapélda – Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML FORM mintap... x +

www.agt.bme.hu/tantargyak/web/formpelda.htm Search »

HTML FORM mintapélda

Töltse ki az űrlapot az összefoglaló tárolásához az adatbázisban.

1. Adja meg a nevét és jelszavát

Név: Jelszó:

2. Válassza ki az témaköröket

Geodézia:
Ipari geodézia
Kozmikus geodézia

Informatika:
GIS
Adatbáziskezelés

3. Adja meg az összefoglalót:

Ezt a szöveget törölje ki és írja be ide az összefoglalóját

4. Küldje el az adatokat vagy kezdje előről

2015 Szeptember 9, Szerda

Form és JavaScript 2.

```
<HTML>
<HEAD>
<TITLE>HTML FORM mintapélda</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
<script language="JavaScript">
```

```
function check_form() {
    var l = document.minta.nev.value.length;
    if (l == 0) {
        alert("Adja meg a nevét!"); return;
    }
}
```

Eseményre végrehajtott függvény

```
</script>
</head>
<BODY>
<H2>HTML FORM mintapélda</H2>
<P>
<FORM name=minta action="javascript:check_form()">
    <B>Név:</B>
    <INPUT type="text" value=vendég name=nev><br />
    <INPUT type="submit" name="kuld" value="küld" >
</FORM>
<HR>
<SCRIPT language="JavaScript" type="text/javascript">
```

```
// nap és hónap nevek tömbje
```

```
var dayNames = new Array("Vasárnap","Hétfő","Kedd","Szerda","Csütörtök","Péntek","Szombat");
```

```
var monthNames = new Array("Január","Február","Március","Április","Május","Június","Július",
    "Augusztus","Szeptember","Október","November","December");
```

```
var dt = new Date();
```

```
var y = dt.getFullYear();
```

```
document.write(y + " " + monthNames[dt.getMonth()] + " " + dt.getDate() + ", " + dayNames[dt.getDay()]);
```

```
// -->
```

```
</SCRIPT>
```

```
</BODY>
```

```
</HTML>
```

Tömbök statikus értékekkel

Dátum objektum kezelése

Időzítés JavaScripttel

<http://www.agt.bme.hu/tantargyak/web/timer.html>

```
counter = 0; // kiírandó rész pozíciója msg-ben
width = 70; // mező szélessége a szöveghez
var spc = " *** "; // elválasztó a hírek közé
var msg = "Javascript programozás" + spc + // görgetendő szöveg
        "Weblap tervezés/készítés" + spc +
        "MySQL/Postgres adatbázisok" + spc +
        "Internetes térképek" + spc;
```

Változók inicializálása
Az oldal HEAD részében

```
function gorget()
{
```

onLoad eseményre

Globális változó

```
    var out;
    counter++;
    if (counter > msg.length) counter = 0;
    out = msg;
    while (out.length <= width) {
        out += spc + msg;
    }
```

Lokális változó

Rövid szöveg ismétlése

```
    out = out.substr(counter, width);
    document.getElementById("banner").innerHTML = out;
    window.setTimeout(gorget, 500); // 0.5 másodperc után újra
```

Banner id-jű div-be írás

```
}
```

Időzítés beállítása újra

További egyszerű példák

<http://www.agt.bme.hu/tantargyak/web/prim.html>

http://www.geod.bme.hu/on_line/hossztorzulas.html

<http://www.agt.bme.hu/tantargyak/web/animacio.html>

<http://www.agt.bme.hu/tantargyak/web/eov.html>

<http://www.agt.bme.hu/tantargyak/web/eovxy.htm>

<http://www.w3schools.com/js/>



jQuery & form (HTML)

```
<html>
<head>
<title>HTML FORM mintapélda</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.js"></script>
<script language="JavaScript" src="form2.js"></script>
</head>
<body>
<h2>HTML FORM mintapélda 2</h2>
<p>
<form name=minta id="minta">
  <B>Név:</B>
  <input type="text" value="vendég" name="nev" id="nev" /><br />
  <input type="button" name="kuld" value="Küld" id="kuld" />
</form>
<hr>
<div id="datum"></div><br />
<div id="hello"></div>
</body>
</html>
```

jQuery & form (JavaScript)

```
<script language="JavaScript">
// nap és hónap nevek tömbje
var dayNames = new Array("Vasárnap", "Hétfő", "Kedd", "Szerda", "Csütörtök",
    "Péntek", "Szombat");

var monthNames = new Array("Január", "Február", "Március", "Április", "Május",
    "Június", "Július", "Augusztus", "Szeptember", "Október", "November",
    "December");

$(document).ready(function() {
    $("#minta").click(function() {
        if ($("#nev").val().length == 0) {
            alert("Adja meg a nevét!");    return;
        } else {
            $("#hello").html("Hello " + $("#nev").val());
        }
    });
    var dt = new Date();
    var y = dt.getFullYear();
    // trick for mozilla
    if (y < 1000) y += 1900;
    $("#datum").html(y + " " + monthNames[dt.getMonth()] + " " +
        dt.getDate() + ", " + dayNames[dt.getDay()]);
});
</script>
```

jQuery és AJAX

http://www.agt.bme.hu/tantargyak/web/tipp_jquery.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-2">
<title>AJAX példa</title>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.js"></script>
<script type="text/javascript" language="javascript">
```

\$(...) speciális jQuery hívás, lehetne jQuery(...) is

```
$(document).ready(function() {
```

Amikor a dokumentum betöltődött

```
    $("#txt1").keyup(function() {
        var str = $("#txt1").val();
        if (str.length == 0) {
            $("#txtHint").html("");
            return;
        }
        $.ajax({
            url: "nevek.php",
            data: { part: str }
        }).done(function(data) {
            $("#txtHint").html(data);
        });
    });
</script>
</head>
<body>
<form>
Név:
<input type="text" id="txt1">
</form>

<p>Tippek: <div id="txtHint"></div></p>
</body>
</html>
```

Amikor a txt1 ID-jű elemben megnyomtunk egy billentyűt

AJAX kérés a szerver felé

Szerver válaszána megjelenítése

**jQuery előnye, hogy a böngészők közötti különbséggel nem kell foglalkoznunk, jobban elválik a tartalom és a viselkedés, a saját kód jóval rövidebb lesz
vö: <http://www.agt.bme.hu/tantargyak/web/tipp.html>**

Egyszerű példa

Hozzunk létre egy programot időpont átalakítására
GPS hét és másodpercbe (Takács Bence ötlete)

Tervezés (űrlap)

The diagram shows a light blue rectangular form representing a web page. Inside the form, the text 'Időpont:' is followed by a text input field containing '2015-09-16T08:30:00'. Below this, the text '1862 hét 331200 másodperc' is displayed. At the bottom left of the form is a button labeled 'Átszámít'. Arrows point from external text boxes to the input field, the result text, and the button.

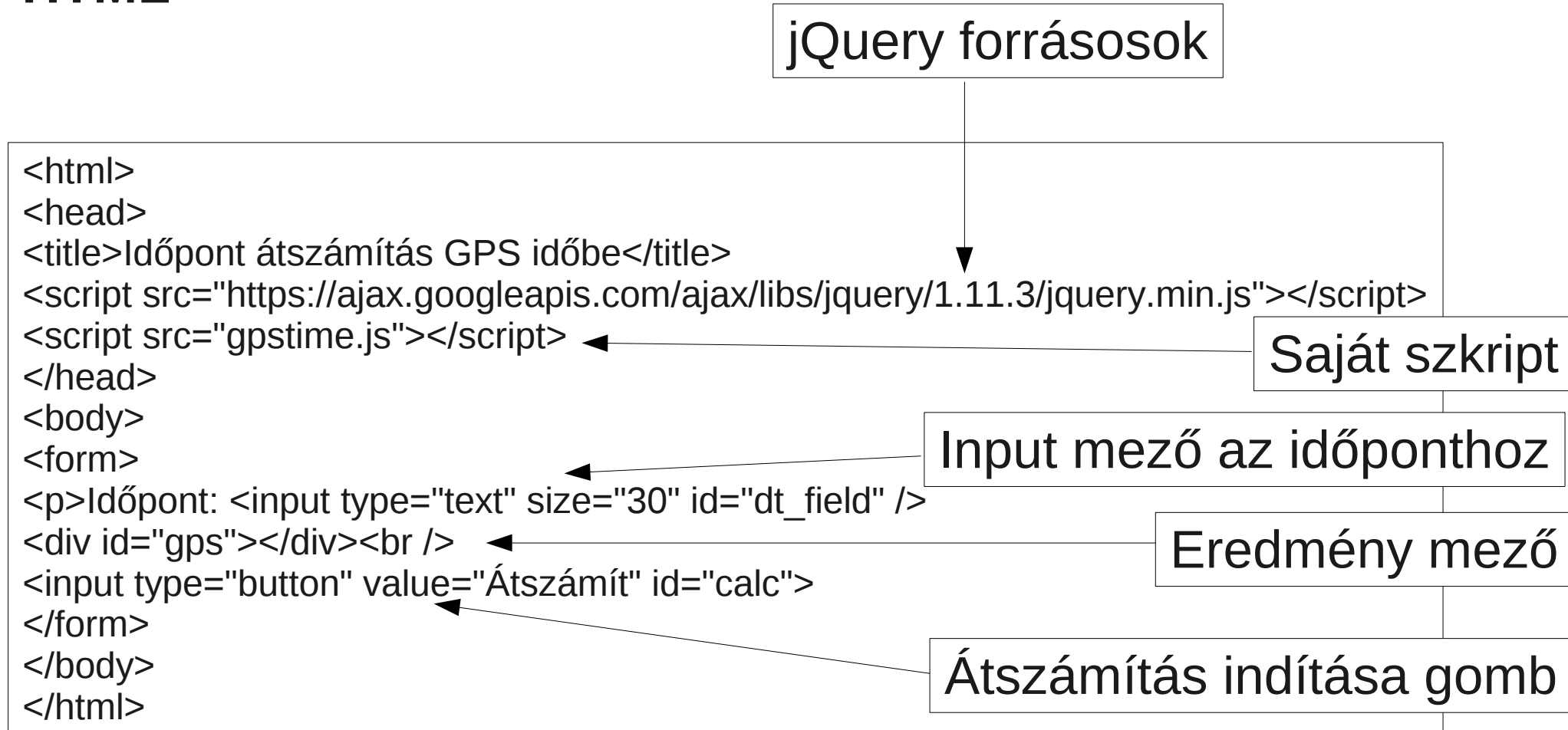
Adatbeviteli mező
indulásnál az aktuális idő
id="dt_field"

A számítás eredménye (div)
id="gps"

Számítás indítása gomb
id="calc"

Egy lehetséges megoldás

HTML



Egy lehetséges megoldás

JavaScript

```
/* Date time to GPS time  
 * based on http://www.ngs.noaa.gov/gps-toolbox/bwr-c.txt */
```

```
var JAN11901 = 15385; // no real constant uppercase used as convention  
var JAN61980 = 44244;  
var SEC_PER_DAY = 86400;  
// days to the start of the month (general year and leap year  
var month_day = [[0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334],  
                 [0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335]];
```

Konstansok a számításhoz

```
// page initialization  
$(document).ready(function() {  
    // add actual date and time  
    var dt = new Date();  
    var dt_str = dt.getFullYear().toString() + "-" +  
        pad(dt.getMonth() + 1, 2) + "-" +  
        pad(dt.getDate(), 2) + "T" +  
        pad(dt.getHours(), 2) + ":" +  
        pad(dt.getMinutes(), 2) + ":" +  
        pad(dt.getSeconds(), 2);
```

HTML dokumentum betöltése
után végrehajtandó függvény

Aktuális időpont szöveglánc

```
$("#dt_field").val(dt_str);  
// click event handler for calculate button  
$("#calc").click(function() {  
    var d = new Date($("#dt_field").val());  
    var gps = ymdhms_to_gps(d);  
    $("#gps").html(gps[0].toString() + " hét " + gps[1].toString() + "  
másodperc");  
});  
});
```

Input mező feltöltése

Átszámítás gomb
esemény kezelője

Egy lehetséges megoldás

JavaScript (folytatás)

```
/* pad number with leading zeros
 * num - the number
 * size - field width (max 10)
 */
function pad(num, size){
    return ('0000000000' + num).substr(-size);
}

/* Calculate GPS week and seconds form date_time (JavaScript Date object)
 * dt - the date and time
 */
function ymdhms_to_gps(dt) {
    var year = dt.getFullYear();
    var leap = year % 4 == 0 ? 1 : 0;
    var yday = month_day[leap][dt.getMonth()] + dt.getDate();
    var mjd = Math.floor((year - 1901) / 4) * 1461 +
        (year - 1901) % 4 * 365 + yday - 1 + JAN11901;
    /* correct timezone offset (convert to UTC)
    */
    var gmthours = -dt.getTimezoneOffset()/60;
    var fmjd = ((dt.getSeconds() / 60 + dt.getMinutes()) / 60 + dt.getHours() - gmthours ) / 24;
    var gps_week = Math.floor((mjd - JAN61980) / 7);
    var sec_of_week = ((mjd - JAN61980) - gps_week * 7 + fmjd) * SEC_PER_DAY;
    return [gps_week, sec_of_week]
}
```

Objektumok

Nincsenek osztályok csak objektumok (prototípus alapú)

Összezártság (függvények és adatok)

Öröklődés (prototípuson keresztül, trükkös)

Többértelműség (nincs paraméter stamp)

```
function személy(first, last, birth) {  
  this.firstName = first;  
  this.lastName = last;  
  this.birth = birth;  
  this.age = function () {  
    return new Date().getFullYear() - this.birth;  
  }  
}
```

Konstruktor függvény (prototípus)

Objektum literális

```
me = { firstName: 'Zoltán',  
      lastName: 'Siki', age: 1958,  
      age: function () {  
        return new Date().getFullYear() - this.birth;  
      }  
}
```

```
me = new személy('Zoltán', 'Siki', 1958);  
me.age();
```

Névtelen függvény

```
személy.prototype.fullname = function() {  
  return this.lastName + ' ' + this.firstName;  
}
```

objektum példány létrehozás

Mi a különbség a konstruktor függvény és az objektum literális között?

minden példány bővítése új függvényekkel!

Komplex szám objektum példa

```
* komplex számok osztálya */
```

```
/* konstruktor */
```

```
function Complex(r, i) {  
    if (r) { this.real = r; } else { this.real = 0.0; }  
    if (i) { this.imag = i; } else { this.imag = 0.0; }
```

Opcionális paraméter
kezelés

```
/* metódusok */
```

```
this.setReal = function(r) { this.real = r; };  
this.setImag = function(i) { this.imag = i; };  
this.getReal = function() { return this.real; };  
this.getImag = function() { return this.imag; };  
this.abs = function() {  
    return Math.sqrt(this.real * this.real + this.imag * this.imag);
```

Getter, setter metódusok

```
};  
this.conj = function() {  
    var w = this.real; this.real = this.imag; this.imag = w;
```

Eredeti objektum
módosítása

```
};  
this.inc = function(c) {  
    this.real += c.real; this.imag += c.imag;  
}  
this.toString = function() {  
    return this.real + '+' + this.imag + 'i';
```

Eredeti objektum
módosítása

```
}
```

```
}
```

Komplex szám objektum használata

`/* komplex osztály használata */`

```
var a = new Complex();  
document.write('a=' + a.toString() + '<br>');  
var b = new Complex(1.0);  
document.write('b=' + b.toString() + '<br>');  
var c = new Complex(0.5, 2.4);  
document.write('c=' + c.toString() + '<br>');  
var d = new Complex();  
d = c; d.conj();    // vigyázat ez elrontja c értékét is! REFERENCIA  
                    // d = Complex(c.real, c.imag) esetén c nem változik!  
document.write('d=' + d.toString() + '<br>');  
document.write('de !!! c=' + c.toString() + '<br>');  
var e = new Complex();  
e.setReal(4.1); e.setImag(1.8);  
document.write('e=' + e.toString() + '<br>');  
d.inc(e);  
document.write('d=' + d.toString() + '<br>');
```

Megoldható olyan conjugalt függvény készítése, mely az eredeti objektumot nem rontja el?

Megoldható olyan összeg függvény készítése, mely egy új objektumba teszi két komplex szám összegét?

Objektumon kívüli megoldás

```
function conjugalt(c) {  
    return new Complex(c.getImag(), -c.getReal())  
}
```

Mi lesz, ha a paraméter
nem Complex típusú objektum?

```
function add(c1, c2) {  
    return new Complex(c1.getReal() + c2.getReal(), c1.getImag() + c2.getImag());  
}
```

Objektumon belüli megoldás

```
Complex.prototype.conjugate = function () {  
    return new Complex(this.getImag(), -this.getReal())  
}
```

Itt nem lehet paraméter típus
probléma

```
Complex.prototype.add = function (c) {  
    if (typeof c == "Number") { return new Complex(this.getReal() + c, this.getImag()); }  
    if (typeof c == "Object" && c.constructor == Complex) {  
        return new Complex(this.getReal() + c.getReal(), this.getImag() + c.getImag());  
    }  
    return new Complex(this.getReal(), c.getImag());  
}
```

Complex + szám

Complex + Complex

Minden más esetben az eredeti Complex

OpenLayers



2006 óta

Nyíl forráskódú, BSD 2 licenz, OSGeo projekt

Vékony kliens, JavaScript könyvtár, AJAX

Szabványos térkép szolgáltatások (WMS, WMTS, WFS, ...)

Nem szabványos szolgáltatások (Google)

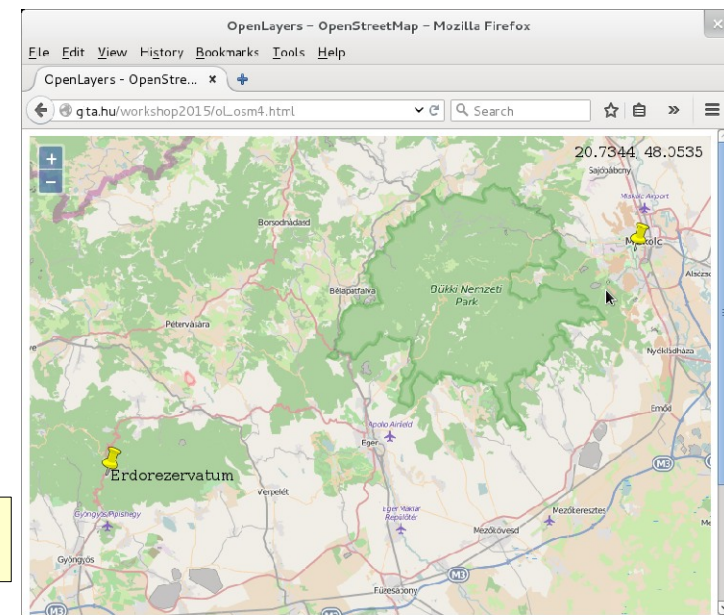
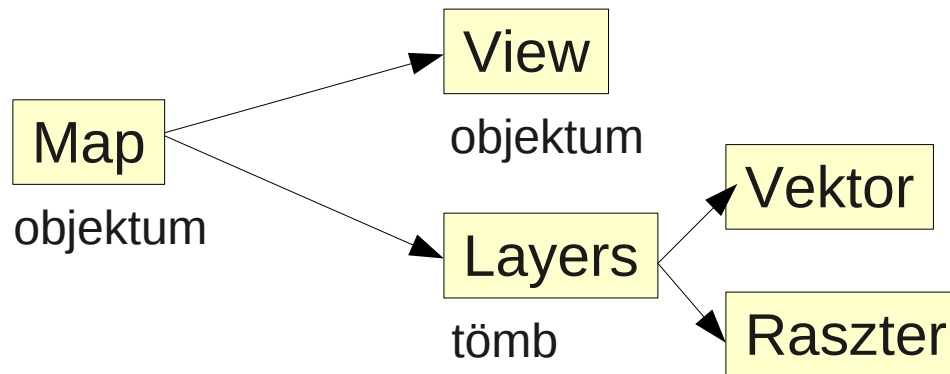
Vektor rétegek (KML, GPX, GeoJSON, GML)

<http://openlayers.org/en/v3.8.2/examples/>

<http://openlayers.org/en/v3.8.2/apidoc/>

<http://openlayers.org/en/v3.8.2/doc/>

Logikai
struktúra



OpenLayers példa

```
<!doctype html>
```

```
<head>
```

```
<title>OpenLayers - OpenStreetMap</title>
```

```
<link rel="stylesheet" href="http://openlayers.org/en/v3.9.0/css/ol.css" type="text/css" />
```

```
<!-- OpenLayer letöltése -->
```

```
<script src="http://openlayers.org/en/v3.9.0/build/ol.js"></script>
```

```
</script>
```

```
// a térkép inicializálás
```

```
function initialize() {
```

```
  // OSM réteg
```

```
  var osmLayer = new ol.layer.Tile({  
    source: new ol.source.OSM()  
  });
```

```
  // pont a térkép középpontjához
```

```
  var eger = ol.proj.transform([20.376004, 47.897257], 'EPSG:4326', 'EPSG:3857');
```

```
  // nézet a térképhez
```

```
  var view = new ol.View({  
    center: eger, // középpont  
    zoom: 14     // nagyítási szint  
  });
```

```
  // térkép
```

```
  var map = new ol.Map({  
    target: 'map' // térképet tartalmazó htm elem  
  });
```

```
  // réteg hozzáadása a térképhez
```

```
  map.addLayer(osmLayer);
```

```
  // nézet hozzáadása a térképhez
```

```
  map.setView(view);
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onLoad="initialize()">
```

```
<!-- térképet tartalmazó elem -->
```

```
<div id="map" class="map"></div>
```

```
</body>
```

```
</html>
```

Stílusok

Forrás

Tile réteg létrehozás (OSM)

WGS84 -> Mercator

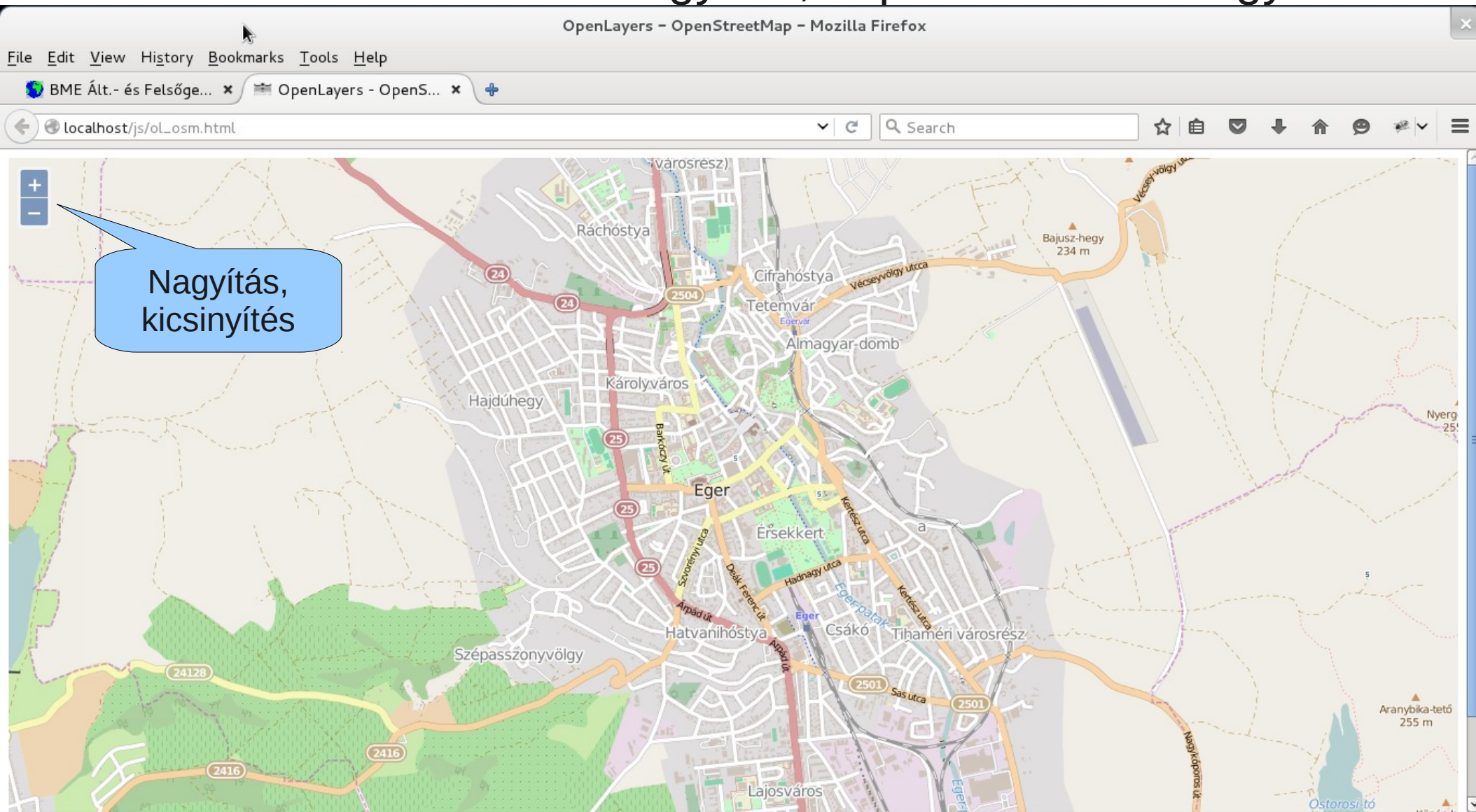
Kompakt megoldás

```
var map = new ol.Map({  
  target: 'map',  
  layers: [  
    new ol.layer.Tile({  
      source: new ol.source.OSM()  
    })  
  ],  
  view: new ol.View({  
    center: ol.proj.transform([20.376004, 47.897257],  
                              'EPSG:4326', 'EPSG:3857'),  
    zoom: 14  
  })  
});
```

Alapfunkciók

Egérműveletek: egérgörgő – nagyítás/kicsinyítés
megragadás és húzás – térkép eltolás

shift + húzás – ablakos nagyítás, dupla kattintás - nagyítás



Billentyűzetről vezérlés

Módosítsuk a map div-et:

```
<div id="map" class="map" tabindex="0">
```

Ettől szelektálható lesz a térkép

Az oldal újratöltése után kattintson a térképre (fókusz)

OpenLayers - OpenStreetMap - Mozilla Firefox

File Edit View History Bookmarks Tools Help

keyboard - Google ke... x OpenLayers - OpenS... x

localhost/js/ol_osm4.html

Search

20.1949, 47.9890

Sajóabony

Borsodnádásd

Salgótarján

Szécsény

Bátontéri Válaszpuszta sportrepülőtér

Bátontéri

Pásztó

Mátra

Verpelet

Eger

Apollo Airfield

Eger Miklós Repülőtér

Mezőkeresztes

Mezőkövesd

Füzesabony

Mezőcsát

Tiszaújváros

Polgár

M3

M35

Tiszacsege

Kesznyéteni Távvédelmi Körzet

Szerencs

Érintő képernyő

Térkép forgatás

Nagyítás, kicsinyítés

Nagyítás, kicsinyítés

Térkép mozgatás

KML réteg

Az előző kódot bővítjük

[Http://www.geod.bme.hu/siki/H-BotanicalGardens.kml](http://www.geod.bme.hu/siki/H-BotanicalGardens.kml) letölteni

Új vektor réteg

```
var projection = ol.proj.get('EPSG:3857');  
var kmlLayer = new ol.layer.Vector({  
  source: new ol.source.Vector({  
    projection: projection,  
    format: new ol.format.KML(),  
    url: 'H-BotanicalGardens.kml'  
  })  
});
```

```
map.addLayer(kmlLayer);
```

Próbáljuk ki!

Találd ki hová kerüljenek a fenti program sorok!

A KML mellett további fájl formátumok is használhatók, pl. GPX, GeoJSON

Koordináta kiírás

```
<div id="map" class="map">  
  <div id="popup"></div>  
</div>
```

```
var element = document.getElementById('popup');  
var popup = new ol.Overlay({  
  element: element  
});  
map.addOverlay(popup);  
  
map.on('click', function(evt) {  
  var coordinate = evt.coordinate;  
  popup.setPosition(coordinate);  
  element.innerHTML = coordinate[0].toFixed(0) +  
    ',' + coordinate[1].toFixed(0);  
});
```

Próbáljuk ki ezt is!

Attribútum kiírás

Az előző kódot bővítjük, koordináta kiírás helyett legyen az arborétum neve felirat

// függvény a felirat megjelenítéshez

```
var displayFeatureInfo = function(pixel, coordinate) {  
  var features = [];  
  // a kattintas kozeleben levo elemek  
  map.forEachFeatureAtPixel(pixel, function(feature, layer) {  
    features.push(feature);  
  });  
  if (features.length > 0) {  
    var info = [];  
    // megtalalt elemek attributumainak osszefuzese  
    for (var i = 0, ii = features.length; i < ii; ++i) {  
      info.push(features[i].get('name'));  
    }  
    // kiiras a terkepre  
    element.innerHTML = info.join(', ') || '(ismeretlen)';  
    // felirat pozicionalasa  
    popup.setPosition(coordinate);  
  } else {  
    // korabbi felirat torlese  
    element.innerHTML = '&nbsp;';  
  }  
};  
// az eger kattintas es a felirat megjelenites osszekapcs.  
map.on('click',function(evt){  
  var coordinate = evt.coordinate;  
  displayFeatureInfo(evt.pixel, coordinate);  
});
```

// szelekcio bealitasa a KML retegre

```
var selectClick = new ol.interaction.Select({  
  layers: [kmlLayer]  
});  
map.addInteraction(selectClick);
```

Talált elemek hozzáadása a tömbhöz

name attribútumok beszerzése

div tartalom módosítása

popup pozíció beállítása

Vezérlők

Az előző kódot bővítjük

```
// terkep vezerlok  
// koordinata kijelzes a jobb felső sarokban  
var mousePosition = new ol.control.MousePosition({  
  coordinateFormat: ol.coordinate.createStringXY(4),  
  projection: 'EPSG:4326'  
});  
map.addControl(mousePosition);  
// lépték bal alsó sarok  
map.addControl(new ol.control.ScaleLine());  
// toloka nagyitas  
map.addControl(new ol.control.ZoomSlider())  
// attekinto terkep  
map.addControl(new ol.control.OverviewMap({  
  layers: [osmLayer]  
}));
```

A vezérlők látható
elemek

Egy kis CSS

```
<style>
  #popup {
    background: yellow;
  }

  .ol-scale-line {
    position: absolute;
    left: 50px;
  }
</style>
```

Módosítsuk az arborétum nevek megjelenítését, legyen sárga a háttér.

A stílus beállítások kerüljenek a ol.css betöltése után!

A lépték vonalzó és az áttekintő térkép ikonja eltakarja egymást. Mozdítsuk el a lépték vonalzót !
(*.ol-scale-line az OpenLayers-ben létrehozott stílus osztály*)

Az eredmény

3.1. Displaying a Scale Li... x OpenLayers - OpenS... x CSS Layout - Horizo... x +

localhost/js/ol_osm4.html ol.control.ScaleLine

20.0384, 47.8316

Egér pozíció WGS84

Nagyítás tolóka

Erdőrezervátum

Áttekintő és lépték

Feladatok

- A http://www.geod.bme.hu/on_line oldalon található JavaScriptek átírása jQuery használatára
- A GPS hét és másodperc átszámító program kibővítése az visszafelé történő átszámítással
- Honlapba beágyazható digitális vagy analóg óra
- Szétnyitható/összezárható lista/al-lista (jQuery hide)
- Egyszerű (hasznos) számítás űrlap adatokból
pl. hőmérséklet átszámítás (Celsius, Fahrenheit, Kelvin)
- Navigációs GPS-ről letöltött GPX állomány megjelenítése OSM térképen
- Homogén koordinátás síkbeli transzformációk grafikus megjelenítéssel (svg vagy WebGL)