JavaScript dióhéjban

Siki Zoltán

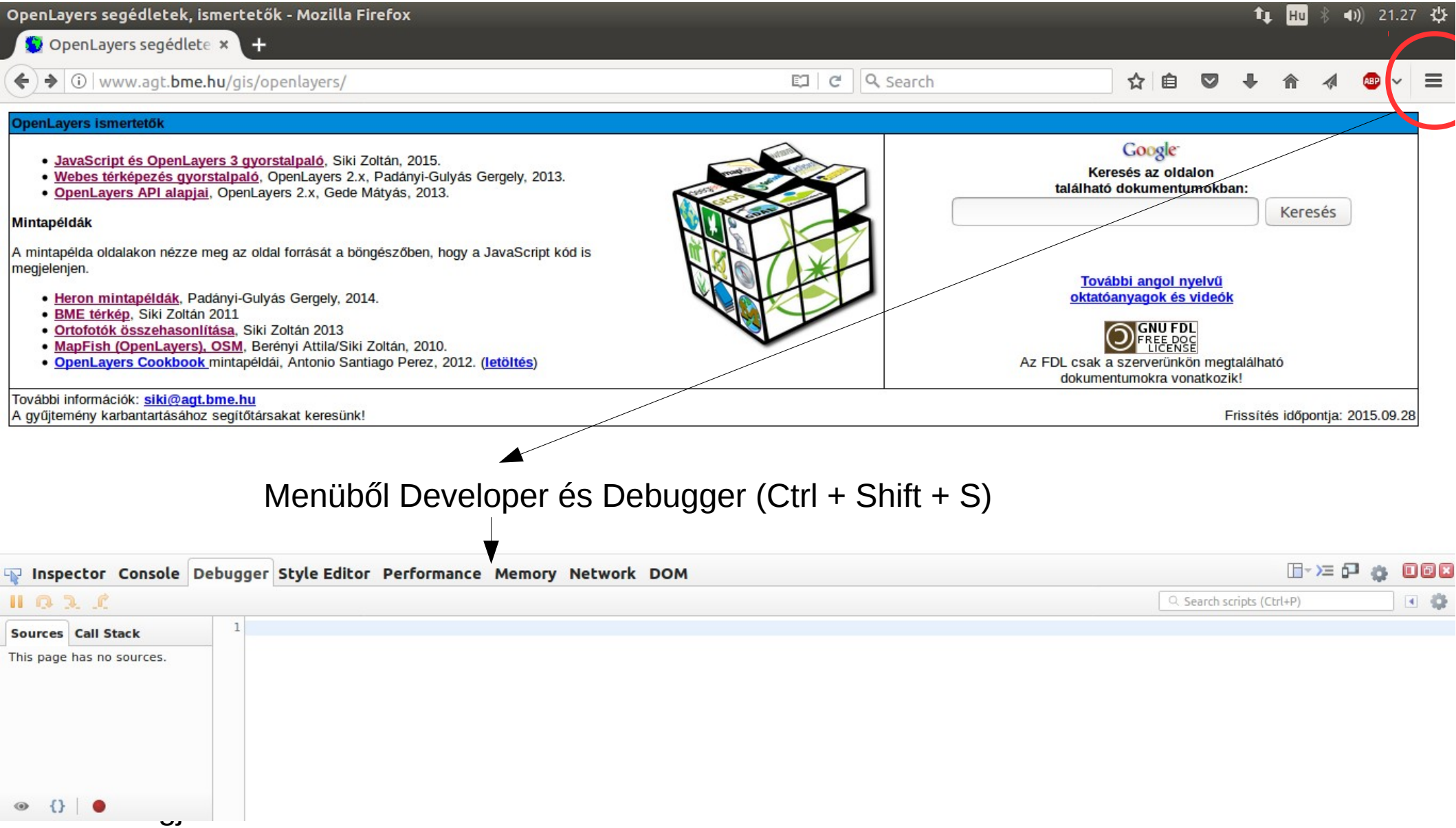
Böngészőkben futtatható szkript nyelv
A Netscape-nél fejlesztették ki (Brendan Eich)
HTML-be ágyazható
A helyi gép erőforrásait **korlátozottan** érheti el
Prototípus alapú objektum orientált nyelv
Böngészőben bekövetkezett események kezelése (pl. klikk)

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

A böngészőkhöz kapcsolódó fejlesztőkörnyezetek

FireFox
Chrome
Internet Explorer

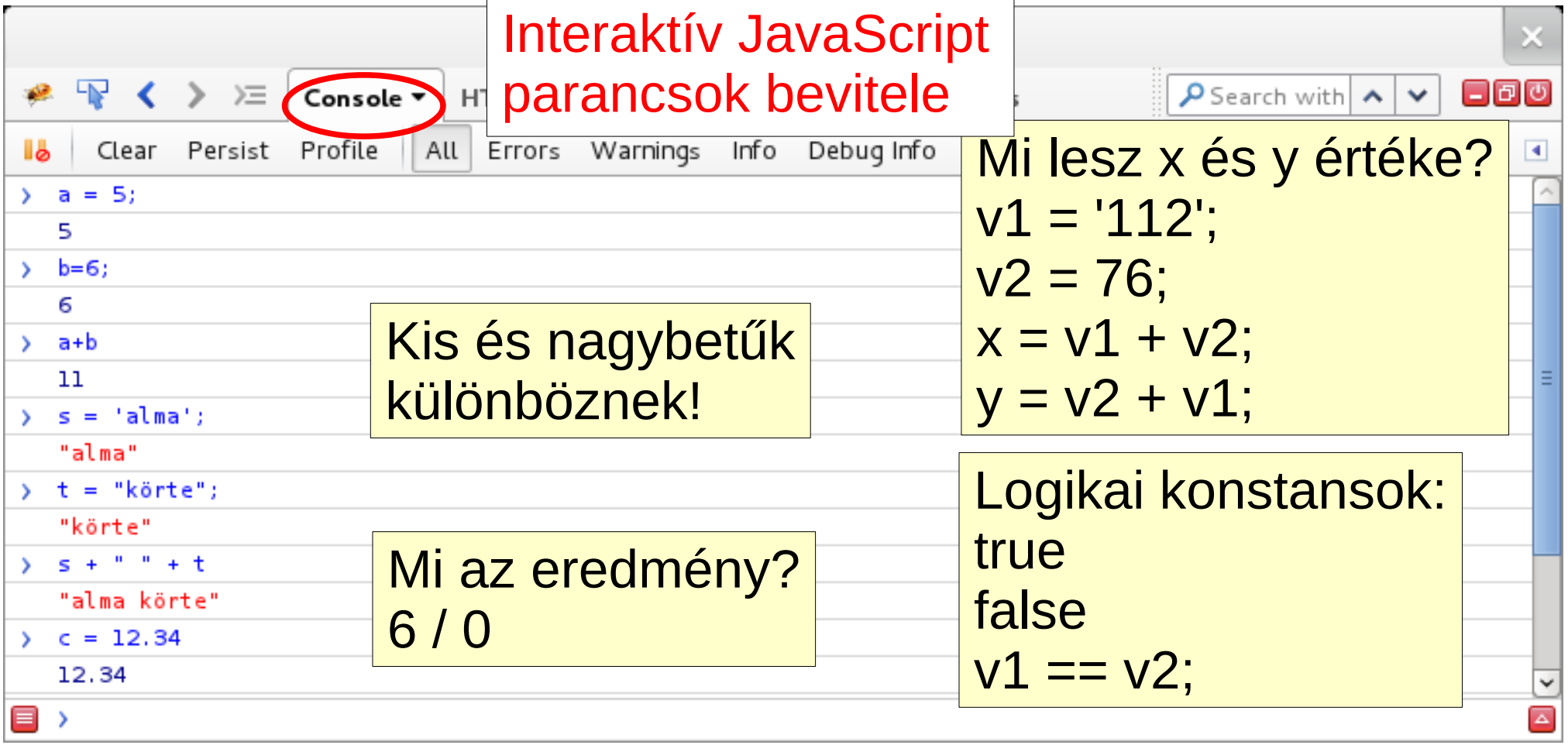
beépített fejlesztőeszközök, debugger



Menüből Developer és Debugger (Ctrl + Shift + S)

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

Kis és nagybetűk különböznek!

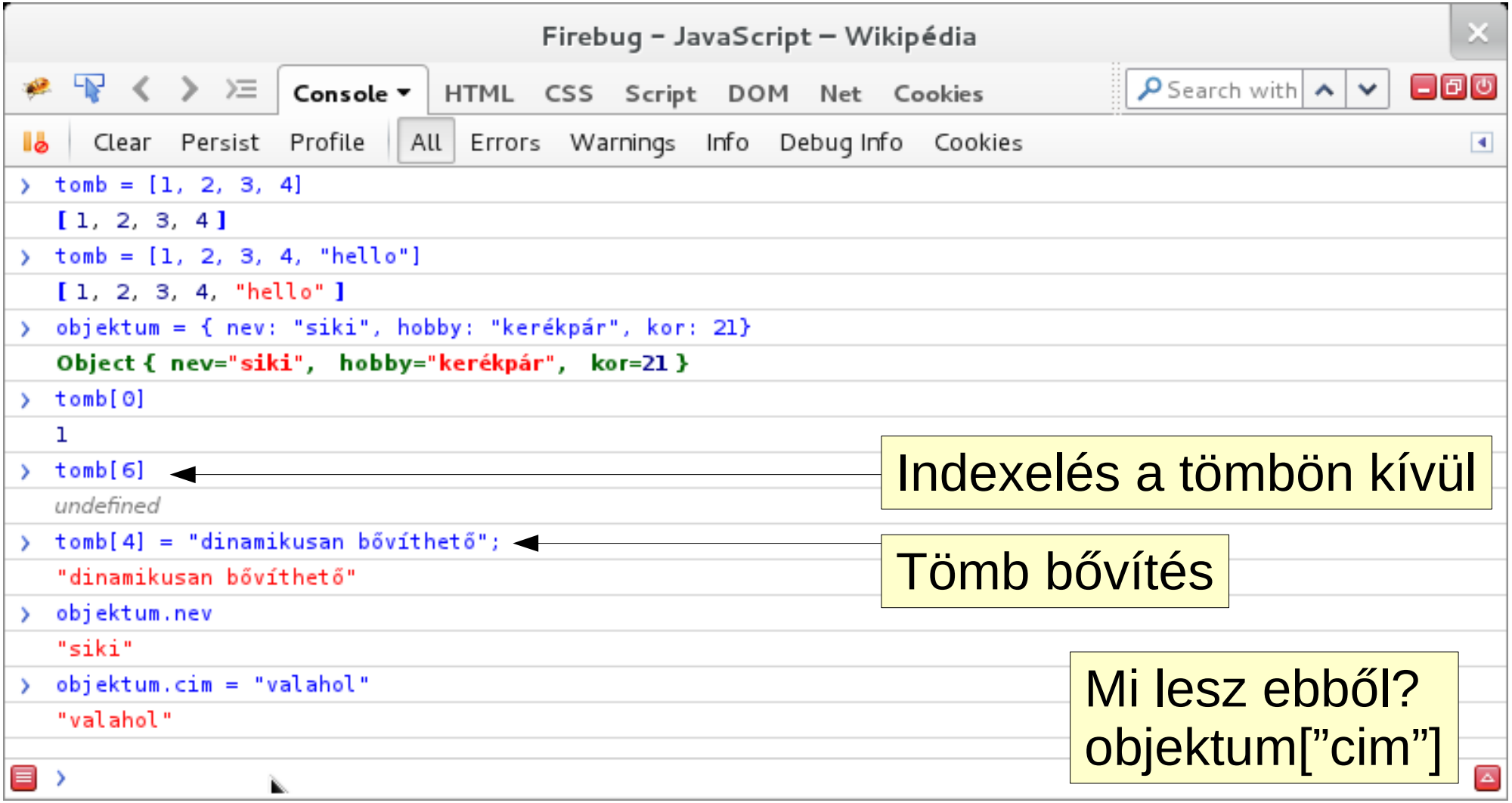
Mi az eredmény?
6 / 0

Mi lesz x és y értéke?
v1 = '112';
v2 = 76;
x = v1 + v2;
y = v2 + v1;

Logikai konstansok:
true
false
v1 == v2;

Összetett adattípusok

Tömb és objektum
Tömb – rendezett adattípus, indexelés 0-tól
Objektum – asszociatív tömb (függvényeket is tartalmazhat)



Fontosabb beépített objektumok

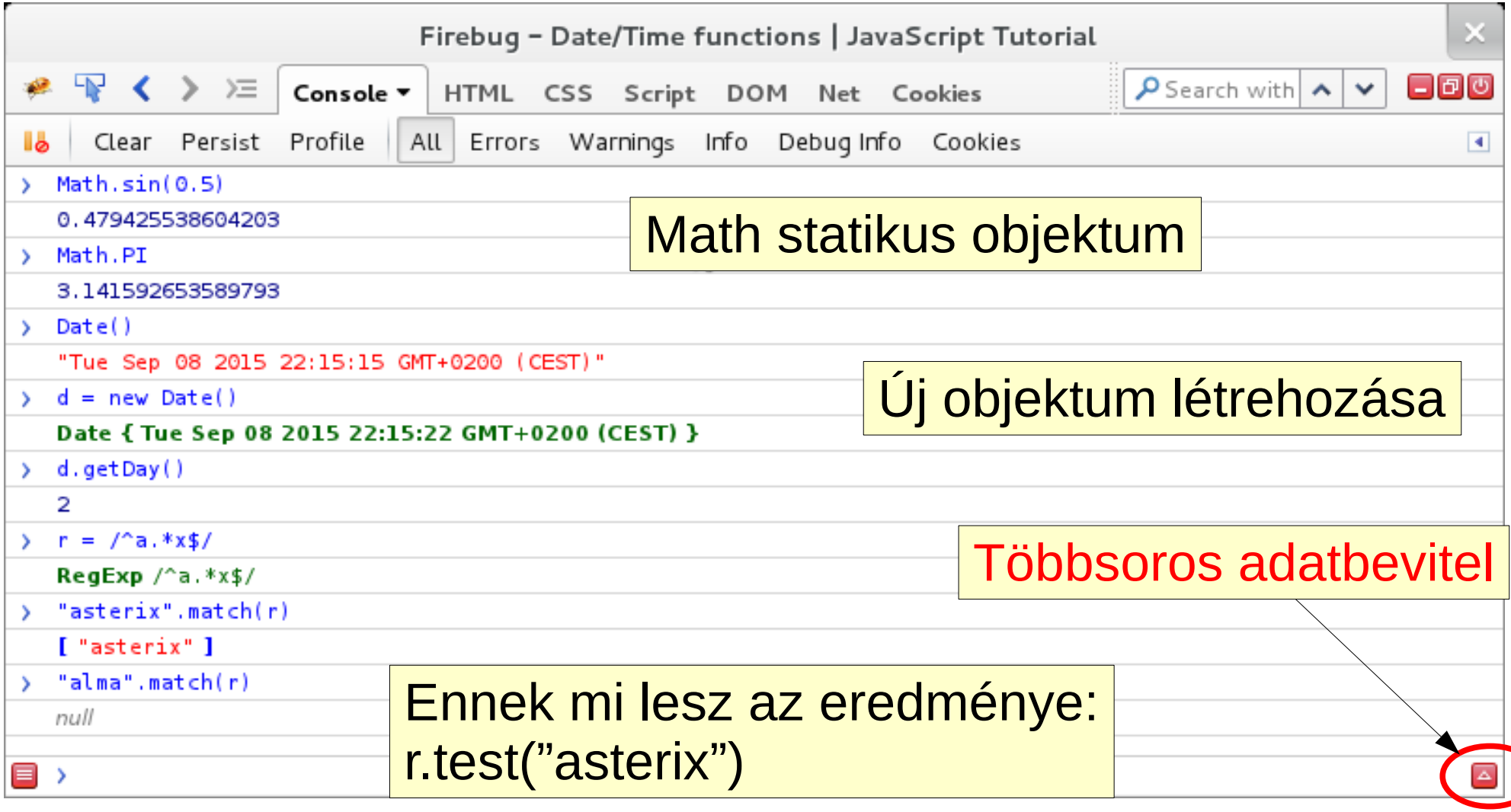
Math, Date, RegExp, Function

Függvények (igen, a JavaScriptben a függvények is objektumok)

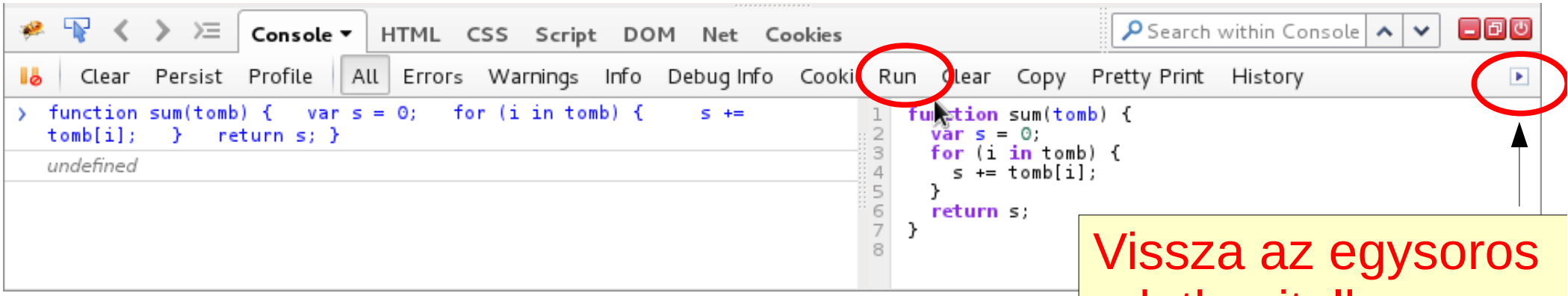
Szabályos kifejezések

Dátum és időpont kezelés

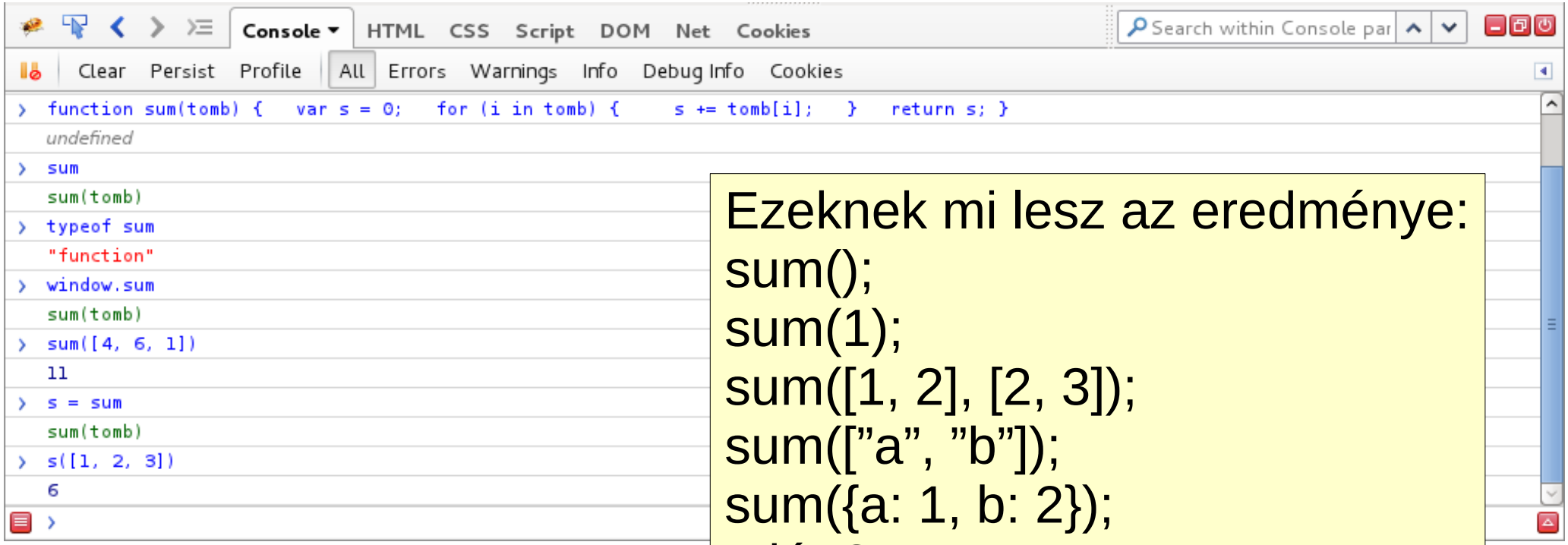
Matematikai függvények, konstansok



Függvények



Vissza az egysoros adatbevitelhez



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

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;

function gorget()
{
    var out;
    counter++;
    if (counter > msg.length) counter = 0;
    out = msg;
    while (out.length <= width) {
        out += spc +  msg;
    }
    out = out.substr(counter, width);
    document.getElementById("banner").innerHTML = out;
     window.setTimeout(gorget, 500);     // 0.5 másodperc után újra
}
```

Változók inicializálása
Az oldal HEAD részében

onLoad eseményre

Globális változó

Lokális változó

Rövid szöveg ismétlése

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/

Objektumok

- Nincsenek osztályok csak objektumok (prototípus alapú)
- Összezártság (függvények és adatok)
- Öröklődés (prototípuson keresztül, trükkös)
- Többértelműség (nincs paraméter stamp)

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

Konstruktor függvény (prototípus)

Névtelen függvény

Objektum literális

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

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

objektum példány létrehozás

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

minden példány bővítése új függvénnyel!

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

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; }

	/* 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);
	};
	this.conj = function() {
		var w = this.real; this.real = this.imag; this.imag = w;
	};
	this.inc = function(c) {
		this.real += c.real; this.imag += c.imag;
	}
	this.toString = function() {
		return this.real + '+' + this.imag + 'i';
	}
}
```

Opcionális paraméter kezelés

Getter, setter metódusok

Eredeti objektum módosítása

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())
}

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

Mi lesz, ha a paraméter nem Complex típusú objektum?

Objektumon belüli megoldás

Itt nem lehet paraméter típus probléma

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

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

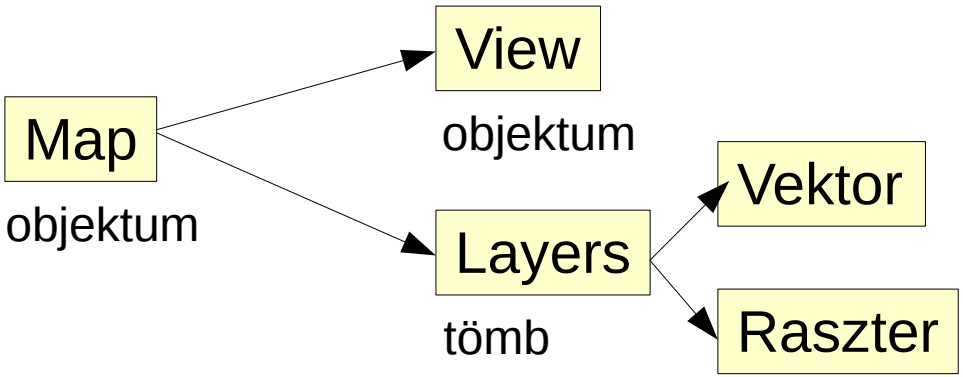
Minden más esetben az eredeti Complex

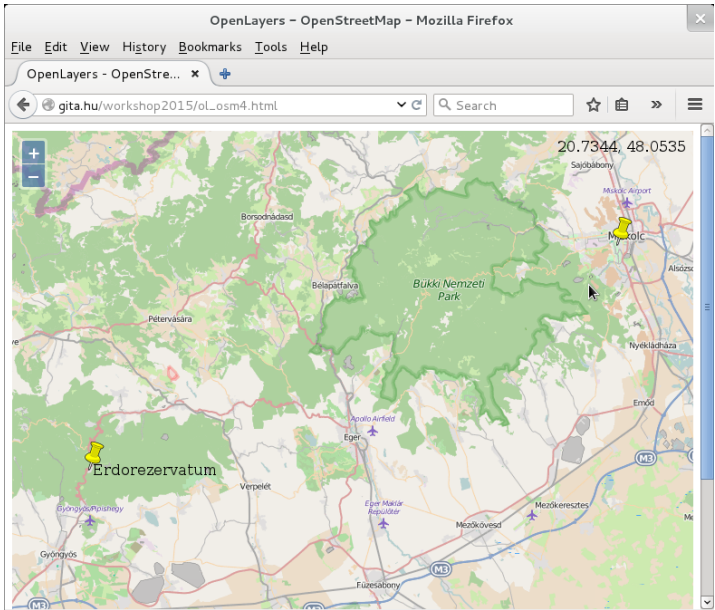
OpenLayers

2006 óta
Nyíl forráskódú, BSD 2 licensz, 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 letoltese -->
    <script src="http://openlayers.org/en/v3.9.0/build/ol.js"></script>
    <script>
    // a terkep inicializalas
    function initialize() {
      // OSM reteg
      var osmLayer = new ol.layer.Tile({
        source: new ol.source.OSM()
      });
      // pont a terkep kozepponthoz
      var eger = ol.proj.transform([20.376004, 47.897257], 'EPSG:4326', 'EPSG:3857');
      // nezet a terkephez
      var view = new ol.View({
        center: eger,   // kozeppont
        zoom: 14        // nagyitasi szint
      });
      // terkep
      var map = new ol.Map({
        target: 'map'  // terkepet tartamazo htm elem
      });
      // reteg hozzadasa a terkephez
      map.addLayer(osmLayer);
      // nezet hozzadasa a tekephez
      map.setView(view);
    }
    </script>
  </head>
  <body onLoad="initialize()">
    <!-- terkepet tartalazo elem -->
    <div id="map" class="map"></div>
  </body>
</html>
```

Stílusok

Forrás

Tile réteg létrehozás (OSM)

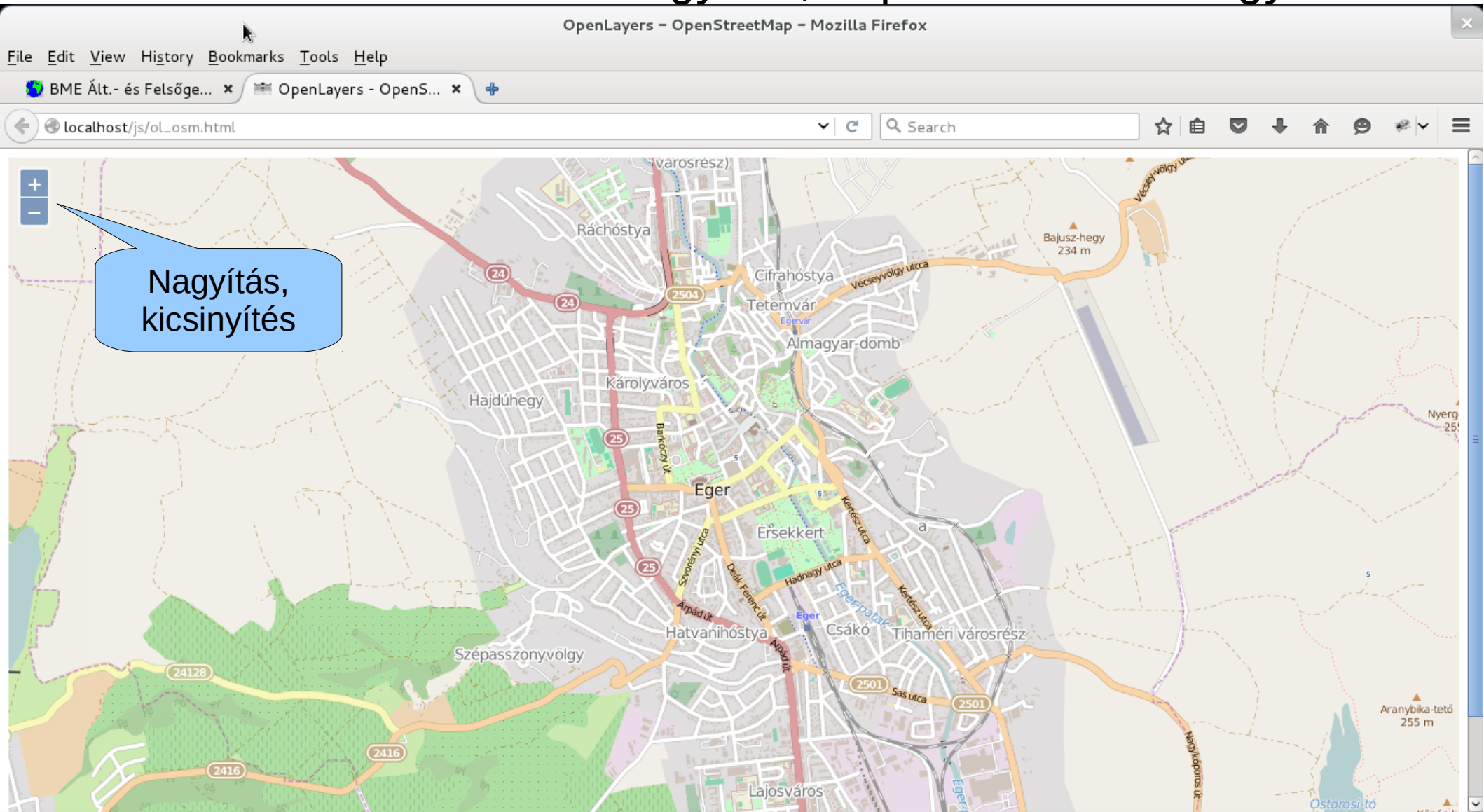
WGS84 -> Mercator

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

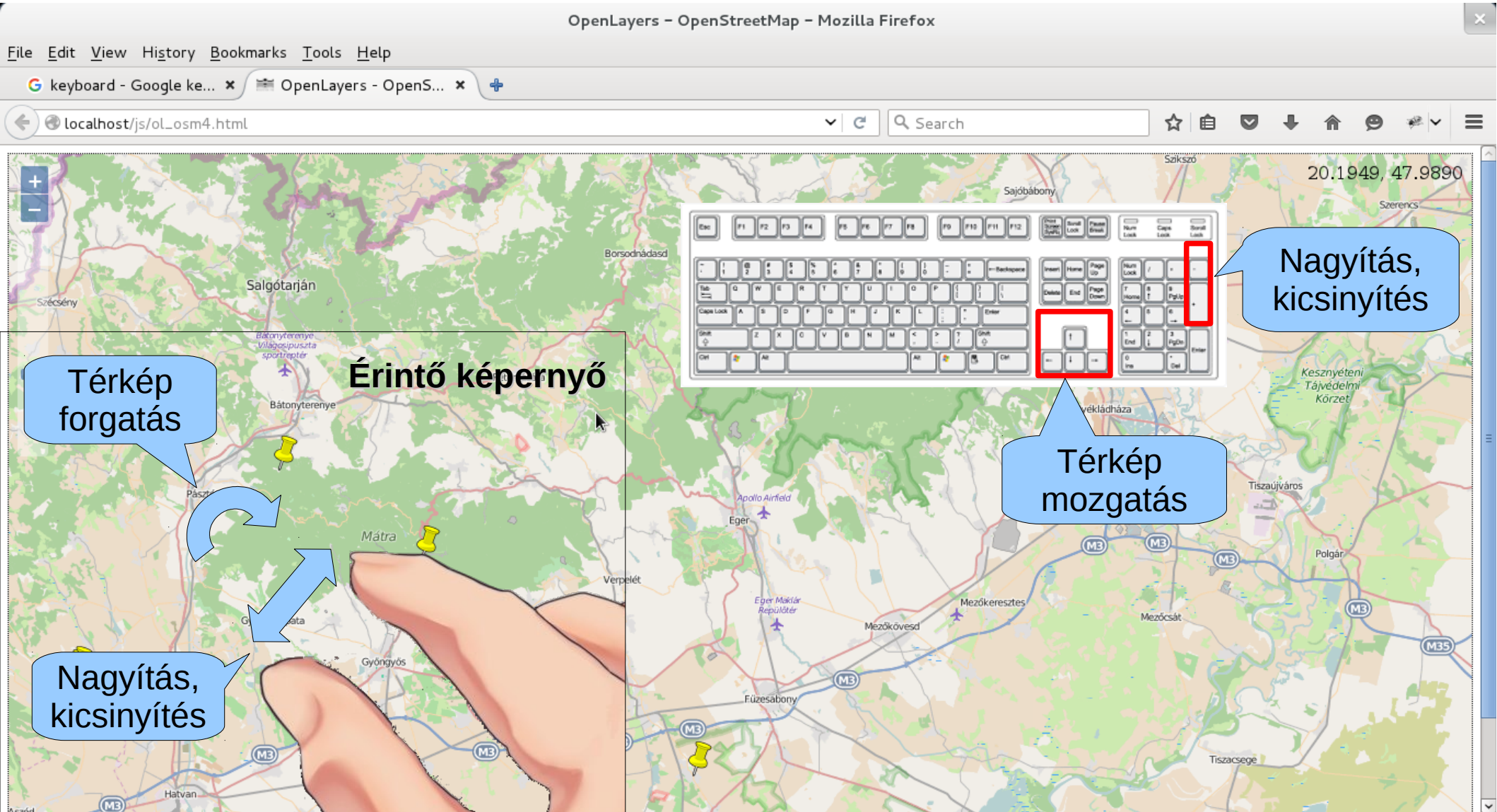
Egérműveletek: egérgörgő – nagyítás/kicsinyítés
megragadás és húzás – térkép eltolás
shift + húzás – ablakos nagyítás, dupla kattintás - nagyítás



Billentyűzetről vezérlés

Módosítsuk a map div-et:
<div id="map" class="map" **tabindex="0"**>
Az oldal újratöltése után kattintson a térképre (fókusz)

Ettől szelektálható lesz a térkép



KML réteg

Az előző kódot bővítjük

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

```
// fuggveny a felirat megjeleniteshez
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 = ' ';
    }
};
// 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 felso sarokban
var mousePosition = new ol.control.MousePosition({
   coordinateFormat: ol.coordinate.createStringXY(4),
   projection: 'EPSG:4326'
});
map.addControl(mousePosition);
// lépték bal also 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

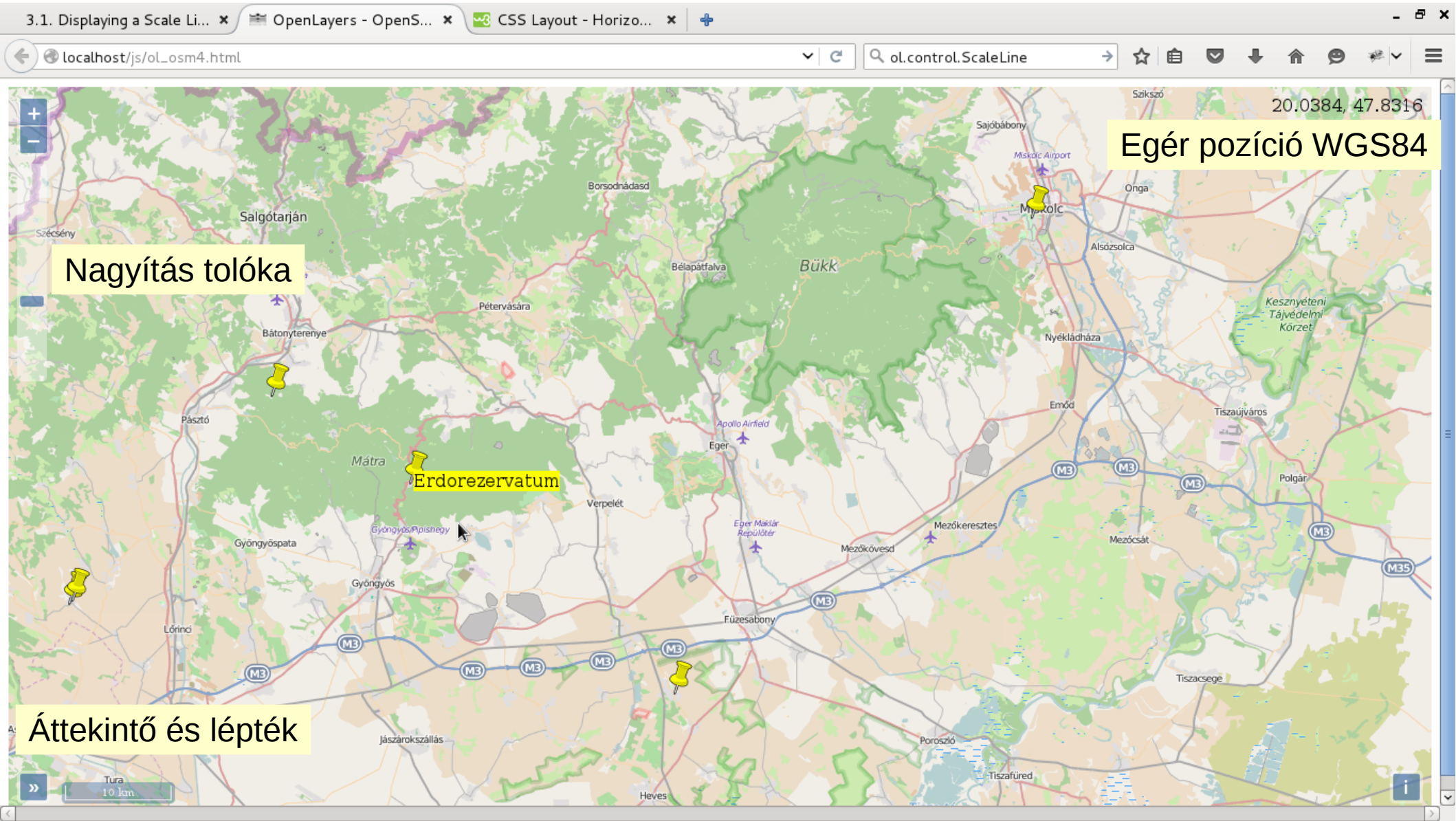
```
<style>
    #popup {
       background: yellow;
    }

    .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...
OpenLayers - OpenS...
CSS Layout - Horizo...
localhost/js/ol_osm4.html
ol.control.ScaleLine
20.0384, 47.8316
Egér pozíció WGS84
Nagyítás tolóka
Áttekintő és lépték
Erdorezervatum
Salgótarján
Bükk
Mátra
Eger
Miskolc
Gyöngyös
Füzesabony
Heves
Tiszafüred
Tiszaújváros
Mezőkövesd
Mezőkeresztes
Tura
10 km

QGIS2WEB modul

- TODO

Feladatok

- A html://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)