Webprogramozás szakkör

Előadás 10 (2013.05.21)

Google Maps API

API: Application Programming Interface (alkalmazásprogramozási felület)

Egy programhoz tartozó kiajánlott függvények, változók és konstansok összességét jelenti, melyeket más programok fel tudnak használni, és ezzel saját szolgáltatásokat készíteni az adott program felhasználásával. Az API fogalmába még bele szokták érteni a hozzá tartozó dokumentációt is, amiből megérthető a használata.

A Google Maps esetében a program maga a Google Maps, amit a hétköznapi életben mi is használhatunk, de fejlesztőként felmerülhet az igény térképet használó alkalmazás készítésére. Saját magunknak interaktív térképet készíteni szinte lehetetlen feladat lenne, de a Google Maps-hez közzétettek fejlesztők számára egy JavaScript nyelvű API-t. Ennek a függvényei, eljárásai lehetővé teszik, hogy a Google Maps szolgáltatásra épülve, saját igényeinknek megfelelően fejlesszünk programokat.

Minden API más! Más objektumokat, függvényeket, változókat, konstansokat használ, így azt nem tudjuk megtanulni, hogyan lehet mindet használni. Azt viszont elsajátíthatjuk, hogyan álljunk neki egy API-ra épülő alkalmazás fejlesztéséhez. Az előző alkalommal a jQuery-ről volt szó, amit önmagában is érdemes megtanulni, hiszen széleskörűen használható programozás technikai szempontból. A mostani példa egy hasznos API lesz, ahol azt az utat járjuk, hogy a dokumentációból megpróbáljuk megismerni a működést. Hogy ne a keresgéléssel teljen rengeteg idő, így minden feladatnál a Tippek segítenek, hol találunk leírást vagy példakódot az adott feladat megoldásához.

Feladat 0 – Ismerjük meg a Google Maps API elérhetőségét, dokumentációját
https://developers.google.com/maps/documentation/javascript/

Fontosabb menüpontok:

- API Reference
- Code Samples

Feladat 1 – Készítsünk egy HTML oldalt egy Google Maps térképpel

Tipp: Code Samples / Basics / Simple map

HTML

A HTML kód szinte végig ugyanaz marad, csak a külön készített JavaScript fájlt kell változtatni.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Google Maps API</title>
<script src="https://maps.googleapis.com/maps/api/js?v=3.exp&sensor=false"></script>
<script src="feladat_01.js"></script>
</head>

<body>
	<div id="map-canvas" style="width:900px;height:500px;"></div>
</body>
</html>
```

JavaScript

```
// JavaScript Document
var map;
function initialize() {
	var mapOptions = {
		zoom: 10,
		center: new google.maps.LatLng(47.49, 19.04),
		mapTypeId: google.maps.MapTypeId.ROADMAP
	};
	map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
}

google.maps.event.addDomListener(window, 'load', initialize);
```

Eddigi ismereteinkből mi került elő?

- HTML + CSS
- JavaScript
  - Változó (*globális* - map, *lokális* - mapOptions)
  - Függvény (initialize())
  - Objektumok
    - mapOptions
    - **new** kulcsszóval való példányosítás (google.maps.Map – globális, a Google Maps API-ban lévő térkép objektum)
  - HTML DOM elérése (document.getElementById())

PÁROS MUNKA: Párokban oldjuk meg a következő feladatokat. A feladatok után zárójelben az adott feladat nehézsége olvasható (1 = könnyű, 2 = közepes, 3 = nehéz). A nehéz feladat kihagyható, ha nem sikerül gyorsan megoldani.

Feladat 2 – A térképre kattintva helyezzünk el jelölőket (Nehézség: 2)

Tipp:

- *API Reference*: Overlays / Marker

- *Code Samples*: Events / Accessing arguments in UI events

```
// JavaScript Document
var map;
function initialize() {
	var mapOptions = {
		zoom: 10,
		center: new google.maps.LatLng(47.49, 19.04),
		mapTypeId: google.maps.MapTypeId.ROADMAP
	};
	map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
	google.maps.event.addListener(map, 'click', function(e) {
		placeMarker(e.latLng);
	});
}

function placeMarker(position) {
	var marker = new google.maps.Marker({
		position: position,
		map: map
	});
	map.panTo(position);
}

google.maps.event.addDomListener(window, 'load', initialize);
```

Feladat 3 – A térképen található vezérlőelemeken (pl. zoom-olás, nézet váltás) változtassunk (Nehézség: 1):

- Tüntessük el az úgynevezett „panControl”-t (ez a bal felső sarokban lévő 4 nyíl)
- A jobb felső sarokban lévő nézetválasztó legördülő menüben jelenjen meg

Tipp:

- *API Reference*: Map / MapOptions / panControl és mapTypeControlOptions
- *Code Samples*: Controls / Adding controls to the map és Controls / Control options

```
var mapOptions = {
	zoom: 10,
	center: new google.maps.LatLng(47.49, 19.04),
	mapTypeId: google.maps.MapTypeId.ROADMAP,
	mapTypeControlOptions: {
		style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
	},
	panControl: false
};
```

Feladat 4/a – A jelölőre való kattintáskor egy információs ablakban jelenítsünk meg tetszőleges szöveget (Nehézség: 2)

Tipp:

- *API Reference*: Overlays / InfoWindow
- *Code Samples*: Events / Using closures in event listeners

Feladat 4/b – A jelölőre való kattintáskor egy információs ablakban jelenítsük meg az adott hely címét, ehhez használjuk a visszafelé való címkódolást (Reverse Geocoding) (Nehézség: 3)

Tipp:

- *API Reference*: Services / Geocoder
- *Code Samples*: Services / Reverse Geocoding

```
// JavaScript Document
var map;
var geocoder;
function initialize() {
	geocoder = new google.maps.Geocoder();
	var mapOptions = {
		zoom: 10,
		center: new google.maps.LatLng(47.49, 19.04),
		mapTypeId: google.maps.MapTypeId.ROADMAP
	};
	map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
	google.maps.event.addListener(map, 'click', function(e) {
		placeMarker(e.latLng);
	});
}

function placeMarker(position) {
	var marker = new google.maps.Marker({
		position: position,
		map: map
	});
	map.panTo(position);
	google.maps.event.addListener(marker, 'click', function() {
		getAddress(marker);
	});
}

function getAddress(marker) {
	geocoder.geocode({'latLng': marker.getPosition()}, function(results, status) {
		if (status == google.maps.GeocoderStatus.OK) {
			if (results[1]) {
				createInfoWindow(marker, results[1].formatted_address);
			} else {
				alert('No results found');
			}
		} else {
			alert('Geocoder failed due to: ' + status);
		}
	});
}

function createInfoWindow(marker, content) {
	var infowindow = new google.maps.InfoWindow();
	infowindow.setContent(content);
	infowindow.open(marker.get('map'), marker);
}

google.maps.event.addDomListener(window, 'load', initialize);
```

Feladat 5 – Cseréljük le a jelölőikont (új ikonkép a kezdőcsomagban van) (Nehézség: 1)

Tipp:

- *API Reference*: Overlays / Marker és MarkerOptions
- *Code Samples*: Overlays / Simple icons

```
function placeMarker(position) {
        var marker = new google.maps.Marker({
                position: position,
                map: map,
                icon: 'images/icon.png'
        });
        map.panTo(position);
        google.maps.event.addListener(marker, 'click', function() {
                getAddress(marker);
        });
}
```

Feladat 6 – Készítsünk animációt a jelölőkhöz (Nehézség: 1)

Tipp:

- *API Reference*: Overlays / Marker és MarkerOptions
- *Code Samples*: Overlays / Marker animations

```
function placeMarker(position) {
        var marker = new google.maps.Marker({
                position: position,
                map: map,
                animation: google.maps.Animation.DROP,
                icon: 'images/icon.png'
        });
        map.panTo(position);

        google.maps.event.addListener(marker, 'click', function() {
                getAddress(marker);
                toggleBounce(marker);
        });
}

function toggleBounce(marker) {
        if (marker.getAnimation() != null) {
                marker.setAnimation(null);
        } else {
                marker.setAnimation(google.maps.Animation.BOUNCE);
        }
}
```

Feladat 7 – Minden egyes újabb kattintás (jelölő letevés) a térképen rajzoljon ki egy útvonalat (Nehézség: 2)

Tipp:

- *API Reference*: Overlays / Polyline
- *Code Samples*: Overlays / Complex polylines

```
var poly;

function initialize() {
        geocoder = new google.maps.Geocoder();
        var mapOptions = {
                zoom: 10,
                center: new google.maps.LatLng(47.49, 19.04),
                mapTypeId: google.maps.MapTypeId.ROADMAP,
                mapTypeControlOptions: {
                  style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
                },
                panControl: false
        };
        var polyOptions = {
                strokeColor: '#000000',
                strokeOpacity: 1.0,
                strokeWeight: 3
        }

        map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
        poly = new google.maps.Polyline(polyOptions);
        poly.setMap(map);

        google.maps.event.addListener(map, 'click', addLatLng);
}

function addLatLng(event) {
        var path = poly.getPath();
        path.push(event.latLng);
        placeMarker(event.latLng);
}
```

Feladat 8 – Jelenítsünk meg a forgalmi adatokat mutató réteget a térképen (Nehézség: 1)

Tipp:

- *API Reference*: Layers / Traffic layer
- *Code Samples*: Layers / Traffic layer

```
function initialize() {
        geocoder = new google.maps.Geocoder();
        var mapOptions = {
                zoom: 10,
                center: new google.maps.LatLng(47.49, 19.04),
                mapTypeId: google.maps.MapTypeId.ROADMAP,
                mapTypeControlOptions: {
                  style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
                },
                panControl: false
        };
        var polyOptions = {
                strokeColor: '#000000',
                strokeOpacity: 1.0,
                strokeWeight: 3
        }

        map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
        poly = new google.maps.Polyline(polyOptions);
        poly.setMap(map);

        var trafficLayer = new google.maps.TrafficLayer();
        trafficLayer.setMap(map);

        google.maps.event.addListener(map, 'click', addLatLng);
}
```

Feladat 9 – Tegyük lehetővé, hogy a jobb egérgomb kattintásra is jelölőket helyezhessünk el, és azokra jelölőkre működjön a fogd és vidd művelet (drag & drop) (Nehézség: 1)

Tipp:

- *API Reference*: Map / Map

```
function initialize() {
        ...

        google.maps.event.addListener(map, 'click', addLatLng);
        google.maps.event.addListener(map, 'rightclick', function(e) {
                placeMarker(e.latLng, true);
        });
}

function addLatLng(event) {
        var path = poly.getPath();
        path.push(event.latLng);
        placeMarker(event.latLng, false);
}

function placeMarker(position, isDragable) {
        var marker = new google.maps.Marker({
                position: position,
                map: map,
                animation: google.maps.Animation.DROP,
                draggable: isDragable,
                icon: 'images/icon.png'
        });
        ...
}
```

Feladat 10 – Bővítsük a HTML kódot egy szövegmezővel és egy gombbal. Minden új jelölő lehelyezésekor írjuk ki a szövegmezőbe az új jelölő koordinátáit. A gombra kattintáskor állítsuk vissza a térkép középpontját az eredeti helyére. (Nehézség: 1)

Tipp:

- *API Reference*: Overlays / Markers; Map / Map; Base / LatLng

HTML

```
<body>
  <div id="map-canvas" style="width:900px;height:500px;"></div>
  <div>
     <input type="text" id="coords" size="75" />
     <input type="button" id="setCenter" value="Set map center" onclick="setMapCenter()" />
  </div>
</body>
```

JavaScript

```
var mapCenter = new google.maps.LatLng(47.49, 19.04);
function initialize() {
        ...
        var mapOptions = {
                zoom: 10,
                center: mapCenter,
                mapTypeId: google.maps.MapTypeId.ROADMAP,
                mapTypeControlOptions: {
                  style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
                },
                panControl: false
        };
        ...
}

function placeMarker() {
        ...
        map.panTo(position);
        document.getElementById('coords').value = position.toString();
        ...
}

function setMapCenter() {
        map.setCenter(mapCenter);
}
```

Teljes JavaScript kód

```
var map;
var geocoder;
var poly;
var mapCenter = new google.maps.LatLng(47.49, 19.04);

function initialize() {
	geocoder = new google.maps.Geocoder();
	var mapOptions = {
		zoom: 10,
		center: mapCenter,
		mapTypeId: google.maps.MapTypeId.ROADMAP,
		mapTypeControlOptions: {
		  style: google.maps.MapTypeControlStyle.DROPDOWN_MENU
		},
		panControl: false
	};
	var polyOptions = {
		strokeColor: '#000000',
		strokeOpacity: 1.0,
		strokeWeight: 3
	}

	map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
	poly = new google.maps.Polyline(polyOptions);
	poly.setMap(map);

	var trafficLayer = new google.maps.TrafficLayer();
	trafficLayer.setMap(map);

	google.maps.event.addListener(map, 'click', addLatLng);
	google.maps.event.addListener(map, 'rightclick', function(e) {
		placeMarker(e.latLng, true);
	});
}

function addLatLng(event) {
	var path = poly.getPath();
	path.push(event.latLng);
	placeMarker(event.latLng, false);
}

function placeMarker(position, isDragable) {
	var marker = new google.maps.Marker({
		position: position,
		map: map,
		animation: google.maps.Animation.DROP,
		draggable: isDragable,
		icon: 'images/icon.png'
	});
	map.panTo(position);
	document.getElementById('coords').value = position.toString();

	google.maps.event.addListener(marker, 'click', function() {
		getAddress(marker);
		toggleBounce(marker);
	});
}
```

```
function getAddress(marker) {
	geocoder.geocode({'latLng': marker.getPosition()}, function(results, status) {
		if (status == google.maps.GeocoderStatus.OK) {
			if (results[1]) {
				createInfoWindow(marker, results[1].formatted_address);
			} else {
				alert('No results found');
			}
		} else {
			alert('Geocoder failed due to: ' + status);
		}
	});
}

function createInfoWindow(marker, content) {
	var infowindow = new google.maps.InfoWindow();
	infowindow.setContent(content);
	infowindow.open(marker.get('map'), marker);
}

function toggleBounce(marker) {
	if (marker.getAnimation() != null) {
		marker.setAnimation(null);
	} else {
		marker.setAnimation(google.maps.Animation.BOUNCE);
	}
}

function setMapCenter() {
	map.setCenter(mapCenter);
}

google.maps.event.addDomListener(window, 'load', initialize);
```

Google Maps API Kivonat

Térkép létrehozása

```
var mapOptions = {
        zoom: 10,
        center: new google.maps.LatLng(47.49, 19.04),
        mapTypeId: google.maps.MapTypeId.ROADMAP
};
var map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
```

Beállításokat egy objektumban adjuk meg (mapOptions), a térképet pedig a google.maps.Map objektum példányosításával hozhatjuk létre.

Eseménykezelés

```
google.maps.event.addListener(MIRE, MILYEN ESEMÉNY, MI TÖRTÉNJEN HA MEGTÖRTÉNIK);

google.maps.event.addListener(map, 'click', fuggvenyhivas());
fuggvenyhivas(event) {
        // amit itt adunk meg, az tortenik a terkepre kattintaskor
        // event parametert kap, ami tartalmazza pl. a kattintás helyét (koordináta)
}
```

Réteg hozzáadása

```
var trafficLayer = new google.maps.TrafficLayer();
trafficLayer.setMap(map);
```

Címkódolás

```
geocoder.geocode({'latLng': marker.getPosition()}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
                if (results[1]) {
                        alert(results[1].formatted_address);
                } else {
                        alert('No results found');
                }
        } else {
                alert('Geocoder failed due to: ' + status);
        }
});
```

Animáció paraméter érték példák

```
google.maps.Animation.BOUNCE
google.maps.Animation.DROP
```