Adding To The Leaf Pile

Leaflet.js Plugin Development

Michael Moore
Flat Rock Geographics

https://github.com/stuporglue/leaflet-spiders

Leaflet in 30 Seconds

- JavaScript web mapping Library
- Free
- Lightweight
  - 9k LOC vs. OpenLayer's 85k
- Many plugins for other use cases
  - Some things are still missing!

Setup

1. Follow a tutorial on leafletjs.com

2. Change leaflet.js to leaflet-src.js

3. Get as close as you can without customizing

Finding What To Change: API

- Good API docs. Start there.
  - Find objects & methods related to task

UI Layers

Marker
Popup

Raster Layers

TileLayer
TileLayer.WMS
TileLayer.Canvas
ImageOverlay

Vector Layers

Path
Polyline
MultiPolyline
Polygon
MultiPolygon
Rectangle
Circle
CircleMarker

Other Layers

LayerGroup
FeatureGroup
GeoJSON

Basic Types

LatLng
LatLngBounds
Point
Bounds
Icon
DivIcon

Controls

Control
Zoom
Attribution
Layers
Scale

Events

Event methods
Event objects

Utility

Class
Browser
Util
Transformation
LineUtil
PolyUtil

DOM Utility

DomEvent
DomUtil
PosAnimation
Draggable

Interfaces

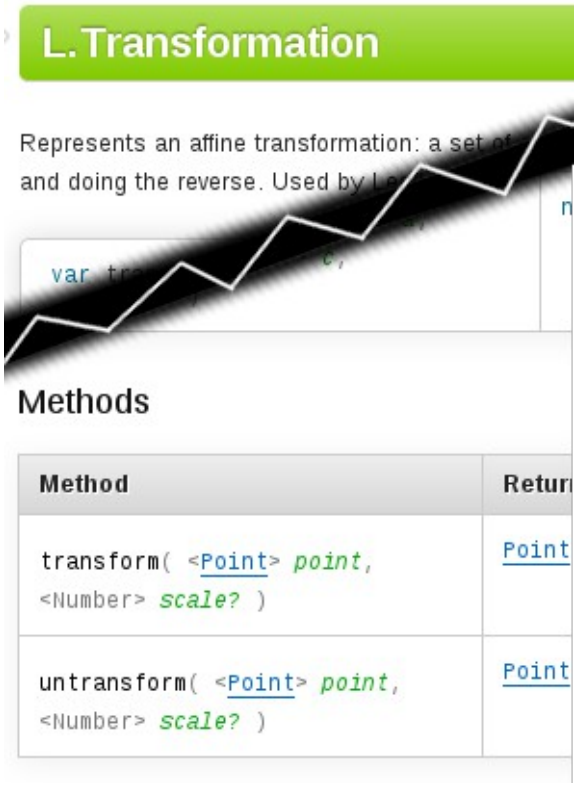
IHandler
ILayer
IControl
IProjection
ICRS

Misc

global switches
noConflict
version

Finding What To Change: API

- Find those objects or in leaflet-src.js
- “L.Point = “
- “L.Point.prototype = ”
- “L.point = “
- “methodname: function(“



Object Types

- L.Object – the object itself
  - Initialization defined here
- L.Object.prototype – the object definition
  - This is what we actually want to modify
  - May be inherited, or may be defined explicitly
    - If inherited, you won't find it in the code
- L.object – convenience method
  - May have extra initialization options

Special functions to know

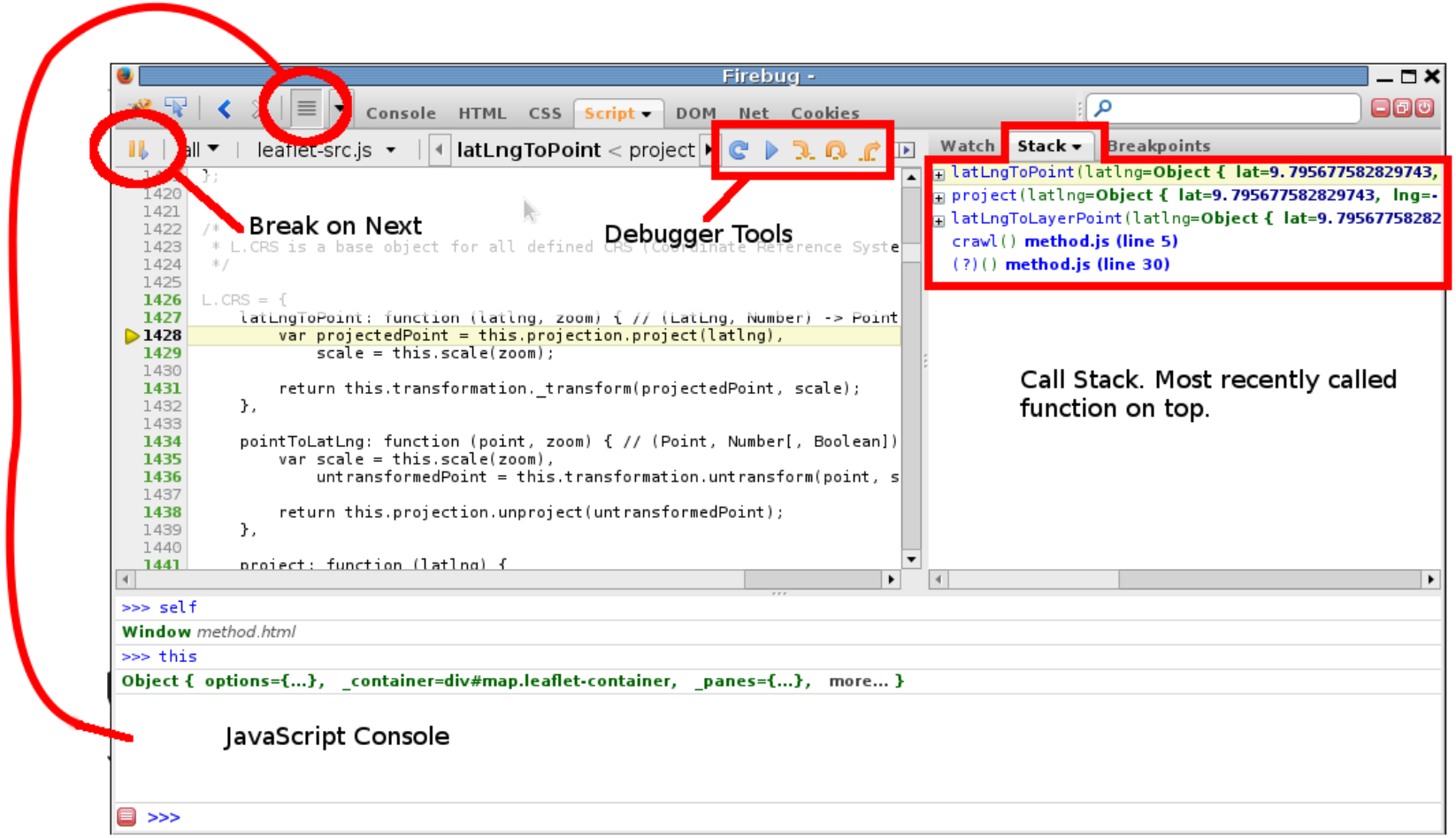
- L.Class
  - Most Leaflet things inherit from this
- L.Class.extend
  - Create a new class by extending another with additional functions
- L.Class.include
  - Set functions in existing class
- L.Class.mergeOptions
  - Set new default option values

```
L.NewClass = L.Marker.extend({funcName: function(){}})
```

Finding What To Change: Debugger

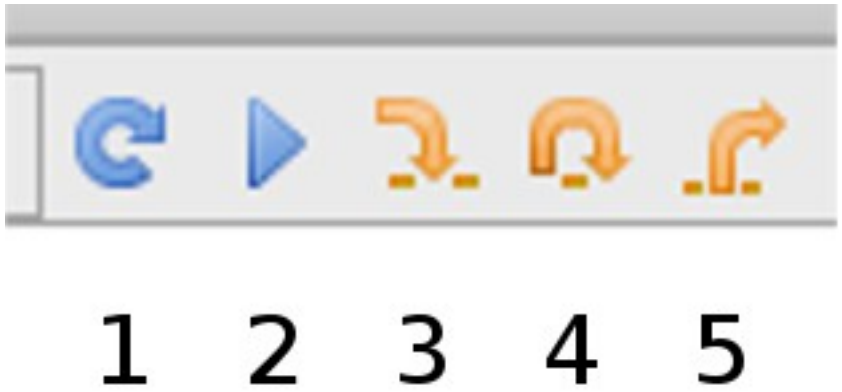
- Explore objects in debugger
  - Expand to see methods and variables
  - Use tab-completion!
- Use debugger on-next to figure out what gets called.
  - Step through to find which function you want to change

Debugger Crash Course



Debugger Tools

1. Reload
2. Continue
3. Step In To
4. Step Over
5. Step Out



Spiders

Let's add some Halloween spiders to the map and make them dance

We will use 3 different methods!

Ways To Bend Leaflet To Your Will

- Add new methods
- Override existing methods
- Create new objects by extending existing

- Not covered
  - New Objects From Scratch
  - Modifying Leaflet Itself

- Not recommended
  - Reaching Inside Leaflet Objects

Static Spiders

Demo

```
// Customization Code
/* NONE*/


// Add some spider code
var spiders = [];
var spider,lat,lng;
var spiderIcon = new L.Icon({iconUrl:'img/spider.png'});

for(var i = 0;i<10;i++){
    lat = Math.random()*180 - 90;
    lng = Math.random()*360 - 180;
    spider = new L.Marker([lat,lng],{icon:spiderIcon});
    spider.addTo(map);
}
```

Adding a New Method

DEMO

```
// Customization Code
// We add a new function called *crawl* to the Marker object

L.Marker.include({crawl: function(){
    // Convert to a pixel and jiggle it up to 5 pixels in any direction
    var pixel = this._map.latLngToLayerPoint(this._latlng);
    pixel.x += Math.round(Math.random()*10) - 5;
    pixel.y += Math.round(Math.random()*10) - 5;

    this._latlng = this._map.layerPointToLatLng(pixel);
    this.update();
}});
```

New Method (cont)

```
// Add some spiders code
var spiders = [];
var spider,lat,lng;
var spiderIcon = new L.Icon({iconUrl:'img/spider.png'});

for(var i = 0;i<10;i++){
    lat = Math.random()*180 - 90;
    lng = Math.random()*360 - 180;
    spider = new L.Marker([lat,lng],{icon:spiderIcon});
    spider.addTo(map);
    spiders.push(spider);
}

// Every 200ms make each spider crawl
setInterval(function(){
    for(var i = 0;i<spiders.length;i++){
        spiders[i].crawl();
    }
},200);
```

Override Method

DEMO

```
// Customization Code
L.Marker.include({

    // Save off the original update function so we can use it later
    _origUpdate: L.Marker.prototype.update,

    // Define our new update function
    update: function(){
        // Convert to a pixel and jiggle it up to 5 pixels in any direction
        var pixel = this._map.latLngToLayerPoint(this._latlng);
        pixel.x += Math.round(Math.random()*10) - 5;
        pixel.y += Math.round(Math.random()*10) - 5;

        this._latlng = this._map.layerPointToLatLng(pixel);
        return this._origUpdate();
    }
});
```

Override Method (cont)

```
// Add some spiders code
var spiders = [];
var spider,lat,lng;
var spiderIcon = new L.Icon({iconUrl:'img/spider.png'});

for(var i = 0;i<10;i++){
    lat = Math.random()*180 - 90;
    lng = Math.random()*360 - 180;
    spider = new L.Marker([lat,lng],{icon:spiderIcon});
    spider.addTo(map);
    spiders.push(spider);
}

setInterval(function(){
    for(var i = 0;i<spiders.length;i++){
        spiders[i].update();
    }
},200);
```

Object Extending

DEMO

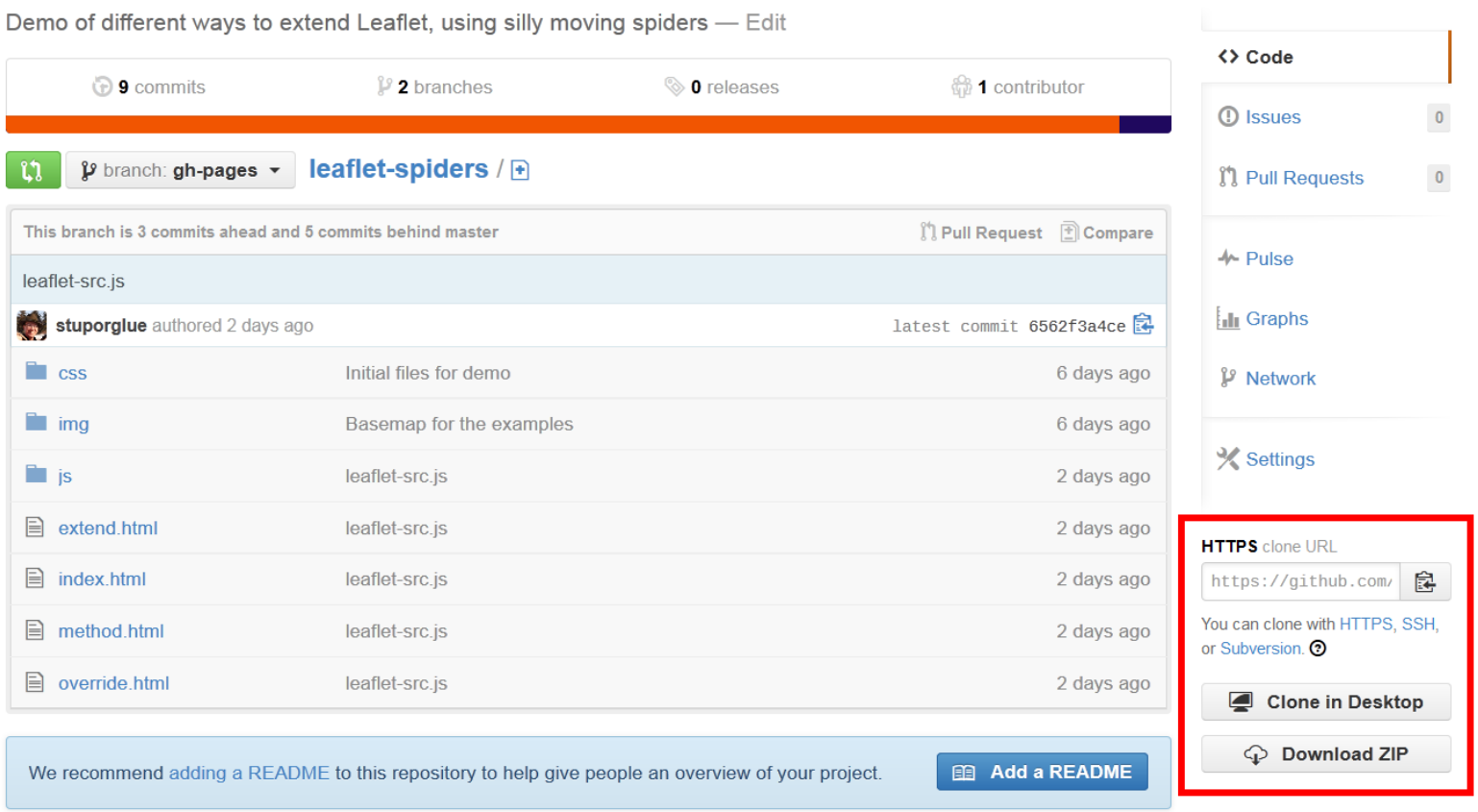
```
L.Spider = L.Marker.extend({
    // Anything we don't define will call up to the L.Marker object
    options: {
        icon: new L.Icon({iconUrl:'img/spider.png'})
    },

    // Called when object created
    initialize: function(){
        var lat = Math.random()*180 - 90;
        var lng = Math.random()*360 - 180;
        this._latlng = L.latLng(lat,lng);

        var self = this;
        setInterval(function(){self.crawl();}, 200);
    },

    // Move the spider
    crawl: function(){
        var pixel = this._map.latLngToLayerPoint(this._latlng);
        pixel.x += Math.round(Math.random()*10) - 5;
        pixel.y += Math.round(Math.random()*10) - 5;

        this._latlng = this._map.layerPointToLatLng(pixel);
        return this.update();
    }
});

L.spider = function(latlng,options){
    return new L.Spider();
};
```

Object Extending (cont)

```
for(var i = 0;i<10;i++){
    new L.Spider().addTo(map);
}
```

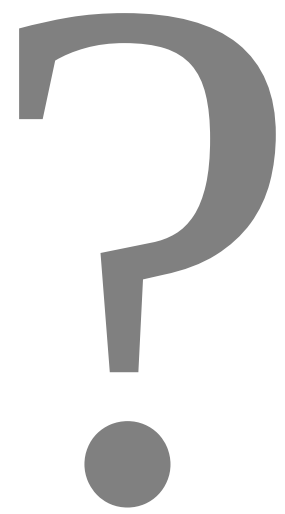
Get the Code on GitHub!

https://github.com/stuporglue/leaflet-spiders



Leaflet

Questions?



Resources

- Leaflet Releases Source Code
- Leaflet API
- Leaflet Plugin Authoring Guide
- Leaflet-Spiders on GitHub
  - Leaflet-Spiders Demo