**A Basic Waypoint**

With Waypoints included, we now have access to a global Waypoint class. We create waypoints by instantiating this class. When creating a new Waypoint we must pass it an options object. There are many properties you can set on this options object, but two of them are required, element, and handler.

var waypoint = new Waypoint({

element: document.getElementById('basic-waypoint'),

handler: function() {

notify('Basic waypoint triggered')

}

})

The element tells Waypoints which DOM element's position to observe during scroll, and handler is the function that will trigger when the top of that element hits the top of the viewport.

**Directions**

You may notice the basic example above triggers when we scroll through the waypoint both downwards and upwards. What if we want to perform different actions when scrolling up, or limit our handler to one direction? When a waypoint is triggered, the handler function is passed a direction parameter.

var waypoint = new Waypoint({

element: document.getElementById('direction-waypoint'),

handler: function(direction) {

notify('Direction: ' + direction)

}

})

**Offsets**

By default a waypoint triggers when the top of the element hits the top of the window. What if we want it to trigger when the element is 20px from the top instead?

var waypoint = new Waypoint({

element: document.getElementById('px-offset-waypoint'),

handler: function(direction) {

notify('I am 20px from the top of the window')

},

offset: 20

})

The offset option can take a variety of different values to help us control where a waypoint is triggered. For more information, check out the [offset option documentation](http://imakewebthings.com/waypoints/api/offset-option).

**this?**

When we're inside the handler function the this keyword is a reference to the Waypoint instance. We can use this object to access all the properties and methods in the [API](http://imakewebthings.com/waypoints/api/waypoint). One of the most useful properties is element, the waypoint's DOM element.

var waypoint = new Waypoint({

element: document.getElementById('element-waypoint'),

handler: function(direction) {

notify(this.element.id + ' triggers at ' + this.triggerPoint)

},

offset: '75%'

})

**Now What?**

This has been a very brief guide to the basics of Waypoints. The [API documentation](http://imakewebthings.com/waypoints/api/waypoint) goes into more depth about all of the options, methods, and properties available to us. In addition to being a reference for more technically savvy developers, each of the API pages acts as a guide for how to use that feature. The docs are intended to be useful for beginners and experts alike.

If you already have a use for Waypoints in mind, take a moment to check out the [Shortcuts](http://imakewebthings.com/waypoints/shortcuts/sticky-elements) section. These shortcut scripts take some of the most common uses of Waypoints and packages them into ready-made extensions.