"use strict"; var \_typeof = "function" == typeof Symbol && "symbol" == typeof Symbol.iterator ? function (t) {

return typeof t

}

: function (t) {

return t && "function" == typeof Symbol && t.constructor === Symbol && t !== Symbol.prototype ? "symbol" : typeof t

};

!function (t) {

"function" == typeof define && define.amd ? define(["jquery"], t) : "object" === ("undefined" == typeof module ? "undefined" : \_typeof(module)) && module.exports ? module.exports = function (i, s) {

return void 0 === s && (s = "undefined" != typeof window ? require("jquery") : require("jquery")(i)), t(s), s

} :

t(jQuery)

}

(

function (t) {

return t.fn.tilt = function (i) {

var s = function () {

this.ticking || (requestAnimationFrame(g.bind(this)), this.ticking = !0)

},

e = function () {

var i = this; t(this).on("mousemove", o), t(this).on("mouseenter", a), this.settings.reset && t(this).on("mouseleave", l), this.settings.glare && t(window).on("resize", d.bind(i))

},

n = function () {

var i = this; void 0 !== this.timeout && clearTimeout(this.timeout), t(this).css({ transition: this.settings.speed + "ms " + this.settings.easing }),

this.settings.glare && this.glareElement.css(

{

transition: "opacity " + this.settings.speed + "ms " + this.settings.easing

}),

this.timeout = setTimeout(function () {

t(i).css({ transition: "" }), i.settings.glare && i.glareElement.css({

transition: ""

})

},

this.settings.speed)

},

a = function (i) {

this.ticking = !1, t(this).css({

"will-change": "transform"

}),

n.call(this), t(this).trigger("tilt.mouseEnter")

},

r = function (i) {

return "undefined" == typeof i && (i = { pageX: t(this).offset().left + t(this).outerWidth() / 2, pageY: t(this).offset().top + t(this).outerHeight() / 2 }), { x: i.pageX, y: i.pageY }

}, o = function (t) { this.mousePositions = r(t), s.call(this) }, l = function () { n.call(this), this.reset = !0, s.call(this), t(this).trigger("tilt.mouseLeave") }, h = function () { var i = t(this).outerWidth(), s = t(this).outerHeight(), e = t(this).offset().left, n = t(this).offset().top, a = (this.mousePositions.x - e) / i, r = (this.mousePositions.y - n) / s, o = (this.settings.maxTilt / 2 - a \* this.settings.maxTilt).toFixed(2), l = (r \* this.settings.maxTilt - this.settings.maxTilt / 2).toFixed(2), h = Math.atan2(this.mousePositions.x - (e + i / 2), -(this.mousePositions.y - (n + s / 2))) \* (180 / Math.PI); return { tiltX: o, tiltY: l, percentageX: 100 \* a, percentageY: 100 \* r, angle: h } }, g = function () { return this.transforms = h.call(this), this.reset ? (this.reset = !1, t(this).css("transform", "perspective(" + this.settings.perspective + "px) rotateX(0deg) rotateY(0deg)"), void (this.settings.glare && (this.glareElement.css("transform", "rotate(180deg) translate(-50%, -50%)"), this.glareElement.css("opacity", "0")))) : (t(this).css("transform", "perspective(" + this.settings.perspective + "px) rotateX(" + ("x" === this.settings.disableAxis ? 0 : this.transforms.tiltY) + "deg) rotateY(" + ("y" === this.settings.disableAxis ? 0 : this.transforms.tiltX) + "deg) scale3d(" + this.settings.scale + "," + this.settings.scale + "," + this.settings.scale + ")"), this.settings.glare && (this.glareElement.css("transform", "rotate(" + this.transforms.angle + "deg) translate(-50%, -50%)"), this.glareElement.css("opacity", "" + this.transforms.percentageY \* this.settings.maxGlare / 100)), t(this).trigger("change", [this.transforms]), void (this.ticking = !1)) }, c = function () { var i = this.settings.glarePrerender; if (i || t(this).append('<div class="js-tilt-glare"><div class="js-tilt-glare-inner"></div></div>'), this.glareElementWrapper = t(this).find(".js-tilt-glare"), this.glareElement = t(this).find(".js-tilt-glare-inner"), !i) { var s = { position: "absolute", top: "0", left: "0", width: "100%", height: "100%" }; this.glareElementWrapper.css(s).css({ overflow: "hidden", "pointer-events": "none" }), this.glareElement.css({ position: "absolute", top: "50%", left: "50%", "background-image": "linear-gradient(0deg, rgba(255,255,255,0) 0%, rgba(255,255,255,1) 100%)", width: "" + 2 \* t(this).outerWidth(), height: "" + 2 \* t(this).outerWidth(), transform: "rotate(180deg) translate(-50%, -50%)", "transform-origin": "0% 0%", opacity: "0" }) } }, d = function () { this.glareElement.css({ width: "" + 2 \* t(this).outerWidth(), height: "" + 2 \* t(this).outerWidth() }) }; return t.fn.tilt.destroy = function () { t(this).each(function () { t(this).find(".js-tilt-glare").remove(), t(this).css({ "will-change": "", transform: "" }), t(this).off("mousemove mouseenter mouseleave") }) }, t.fn.tilt.getValues = function () { var i = []; return t(this).each(function () { this.mousePositions = r.call(this), i.push(h.call(this)) }), i }, t.fn.tilt.reset = function () { t(this).each(function () { var i = this; this.mousePositions = r.call(this), this.settings = t(this).data("settings"), l.call(this), setTimeout(function () { i.reset = !1 }, this.settings.transition) }) }, this.each(function () {

var s = this; this.settings = t.extend({ maxTilt: t(this).is("[data-tilt-max]") ? t(this).data("tilt-max") : 20, perspective: t(this).is("[data-tilt-perspective]") ? t(this).data("tilt-perspective") : 300, easing: t(this).is("[data-tilt-easing]") ? t(this).data("tilt-easing") : "cubic-bezier(.03,.98,.52,.99)", scale: t(this).is("[data-tilt-scale]") ? t(this).data("tilt-scale") : "1", speed: t(this).is("[data-tilt-speed]") ? t(this).data("tilt-speed") : "400", transition: !t(this).is("[data-tilt-transition]") || t(this).data("tilt-transition"), disableAxis: t(this).is("[data-tilt-disable-axis]") ? t(this).data("tilt-disable-axis") : null, axis: t(this).is("[data-tilt-axis]") ? t(this).data("tilt-axis") : null, reset: !t(this).is("[data-tilt-reset]") || t(this).data("tilt-reset"), glare: !!t(this).is("[data-tilt-glare]") && t(this).data("tilt-glare"), maxGlare: t(this).is("[data-tilt-maxglare]") ? t(this).data("tilt-maxglare") : 1 }, i), null !== this.settings.axis && (console.warn("Tilt.js: the axis setting has been renamed to disableAxis. See https://github.com/gijsroge/tilt.js/pull/26 for more information"), this.settings.disableAxis = this.settings.axis), this.init = function () { t(s).data("settings", s.settings), s.settings.glare && c.call(s), e.call(s) },

this.init()

})

}, t("[data-tilt]").tilt(), !0

});