# Shadow Text

### API

#### useEl

Should ideally behave like a textElement: .text, .textAnchor, .letterSpacing, .style.opacity, .style.display, .style.fontSize, .style.fontFamily, ...

#### main

.style.fill, .style.opacity, .style.display, ...

#### light and shadow

Same as main, plus .x, .y

### Object Structure

main, light and shadow should be closures that hide a reference to the corresponding TextElement object. They should expose an API through which the element object can be manipulated. See fitbit-subclass-widget? Use separate constructor function or IIFE (see fitbit-subclass-widget)? Use prototype-chained objects for inheritance (see fitbit-polyline)?

widget, main, light and shadow APIs (including .style) might be able to extend/subclass/prototype from common (virtual) base classes.

#### Property Applicability

| **Property** | **widget (use)** | **main** | **light, shadow** |
| --- | --- | --- | --- |
| .style.opacity |  |  |  |
| .style.display |  |  |  |
| .style.fontSize |  |  |  |
| .style.fontFamily |  |  |  |
| .style.fill |  |  |  |
| .text |  |  |  |
| .textAnchor |  |  |  |
| .letterSpacing |  |  |  |
| .x | from <use> |  |  |
| .y | from <use> |  |  |

StyleCommon: opacity, display

StyleWidget extends StyleCommon: fontSize, fontFamily

StyleSubText extends StyleCommon: fill

Use class (with extend) to create objects that comprise the style API for widget and each subText.

## Scratch Area

### getBBox() DOMRect

| **Member** |  |  |
| --- | --- | --- |
| bottom | =y+height |  |
| height | fundamental |  |
| left | =x |  |
| right | =x+width |  |
| top | =y |  |
| width | fundamental |  |
| x | fundamental | reduce by max extent to which light or shadow is to left of main |
| y | fundamental | reduce by max extent to which light or shadow is above main |

Assume that mainEl always has display===’inline’.

Start with getBBox() from mainEl.

#### Adjustments caused by lightEl and shadowEl

If (display===’none’) 0; else...

x -= min(min(light.x,0), min(shadow.x,0))