Scrollytelling

I’ve always loved the idea of scrolling text and having something happen as a reaction; somehting like haing images load when the top of the image is approaching the viewport or to change a portion of the page in relation to how other portion of the same page scrolls in or out of focus.

> https://russellgoldenberg.github.io/scrollama/fixed-css/

```
<!DOCTYPE html>
<html>

<head>
<html>a charset="utf-8">
  <meta http-equiv="X-UA-Compatible" co<meta http-equiv="X-UA-Compatible"
  <meta name="description" content="Scrollama Demo: Fixed CSS">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <style><meta name="description" content="Scrollama Demo: Fixed CSS">
    /* default / demo page */

    * {
      box-sizing: border-box;
    }

    html,
    body {
      margin: 0;
      padding: 0;
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,
    }'Segoe UI'body {
      min-height: 1280px;
      font-weight: 300;
      color: #2a2a2a;
    }

    p,
    h1,
    h2,
```

```css
h3,
h4,
a {
  margin: 0;
  font-weight: 300;
}

a,
a:visited,
a:hover {
  color: #f30;
  text-decoration: none;
  border-bottom: 1px solid currentColor;
}

#intro {
  max-width: 40rem;
  margin: 1rem auto;
  text-align: center;
}

.intro__overline {
  font-size: 1.4rem;
}

.intro__hed {
  font-size: 1.4rem;
  margin: 1.5rem auto;
  text-transform: uppercase;
  font-weight: 900;
  letter-spacing: 0.05em;
}

.intro__dek {
  font-size: 1.4rem;
}

/* demo */
```

```
#intro {
  margin-bottom: 320px;
}

#outro {
  height: 640px;
}

/* scrollama */

#scroll {
  position: relative;
  border-top: 1px dashed #000;
  border-bottom: 1px dashed #000;
}

.scroll__graphic {
  position: absolute;
  top: 0;
  left: 0;
  bottom: auto;
  background-color: #fff;
  -webkit-transform: translate3d(0, 0, 0);
  -moz-transform: translate3d(0, 0, 0);
  transform: translate3d(0, 0, 0);
}

.scroll__graphic.is-fixed {
  position: fixed;
}

.scroll__graphic.is-bottom {
  bottom: 0;
  top: auto;
}

.chart {
  position: absolute;
  right: 1rem;
  top: 50%;
  -moz-transform: translateY(-50%);
  -webkit-transform: translateY(-50%);
  transform: translateY(-50%);
  background-color: #ddd;
  border: 1px solid #000;
}

.chart p {
  text-align: center;
  padding: 1rem;
  position: absolute;
  top: 50%;
  left: 50%;
  -moz-transform: translate(-50%, -50%);
  -webkit-transform: translate(-50%, -50%);
  transform: translate(-50%, -50%);
  font-size: 8rem;
  font-weight: 900;
  color: #666;
}

.scroll__text {
  position: relative;
  padding: 0 1rem;
  max-width: 30rem;
  width: 33%;
}

.step {
  margin: 2rem auto;
  border: 1px solid #333;
}

.step.is-active {
  background-color: lightgoldenrodyellow;
  }

  .step p {
    text-align: center;
    padding: 1rem;
    font-size: 1.5rem;
  }
='intro__overline'>
    <a href='https://github.com/russellgoldenberg/scrollama'>scrollama.j
  </p>
  <h1 class='intro__hed'>Demo: Sticky Graphic</h1>
  <p class='intro__dek'>
    Start scrolling to see how it works.
  </p>
</section>
<section id='scroll'>
  <div class='scroll__graphic'>
    <div class='chart'>
      <p>0</p>
    </div>
  </div>
  <div class='scroll__text'>
    <div class='step' data-step='1'>
      <p>STEP 1</p>
    </div>
    <div class='step' data-step='2'>
      <p>STEP 2</p>
    </div>
    <div class='step' data-step='3'>
      <p>STEP 3</p>
    </div>
    <div class='step' data-step='4'>
      <p>STEP 4</p>
    </div>
  </div>
</section>
<section id='outro'></section>
<div class='debug'></div>
```

```
<script src='../d3.v4.min.js'></script>
<script src='../scrollama.min.js'></script>
<script>
  // using d3 for convenience
  var container = d3.select('#scroll');
  var graphic = container.select('.scroll__graphic');
  var chart = graphic.select('.chart');
  var text = container.select('.scroll__text');
  var step = text.selectAll('.step');

  // initialize the scrollama
  var scroller = scrollama();

  // generic window resize listener event
  function handleResize() {
    // 1. update height of step elements
    var stepHeight = Math.floor(window.innerHeight * 0.75);
    step.style('height', stepHeight + 'px');

    // 2. update width/height of graphic element
    var bodyWidth = d3.select('body').node().offsetWidth;

    graphic
      .style('width', bodyWidth + 'px')
      .style('height', window.innerHeight + 'px');

    var chartMargin = 32;
    var textWidth = text.node().offsetWidth;
    var chartWidth = graphic.node().offsetWidth - textWidth - chartMarg

    chart
      .style('width', chartWidth + 'px')
      .style('height', Math.floor(window.innerHeight / 2) + 'px');


    // 3. tell scrollama to update new element dimensions
    scroller.resize();
  }
```

```
// scrollama event handlers
function handleStepEnter(response) {
  // response = { element, direction, index }

  // add color to current step only
  step.classed('is-active', function (d, i) {
    return i === response.index;
  })

  // update graphic based on step
  chart.select('p').text(response.index + 1)
}

function handleContainerEnter(response) {
  // response = { direction }

  // sticky the graphic (old school)
  graphic.classed('is-fixed', true);
  graphic.classed('is-bottom', false);
}

function handleContainerExit(response) {
  // response = { direction }

  // un-sticky the graphic, and pin to top/bottom of container
  graphic.classed('is-fixed', false);
  graphic.classed('is-bottom', response.direction === 'down');
}

function init() {
  // 1. force a resize on load to ensure proper dimens<a href='https:/
  // this will also initialize trigger observations
  // 3. bind scrollama event handlers (this can be chained like below)
  scroller.setup({
    container: '#scroll',
    graphic: '.scroll__graphic',
    text: '.scroll__text',
        step: '.scroll__text .step',
        debug: true,
      })
        .onStepEnter(handleStepEnter)
        .onContainerEnter(handleContainerEnter)
        .onContainerExit(handleContainerExit);

      // setup resize event
      window.addEventListener('resize', handleResize);
    }

    // kick things off
    init();
  </script>
</body>

</html>
</script>
```

https://pudding.cool/process/how-to-implement-scrollytelling/

https://pudding.cool/process/responsive-scrollytelling/

Alternatives

- Scrollama