

SVG Filters

A CRASH COURSE



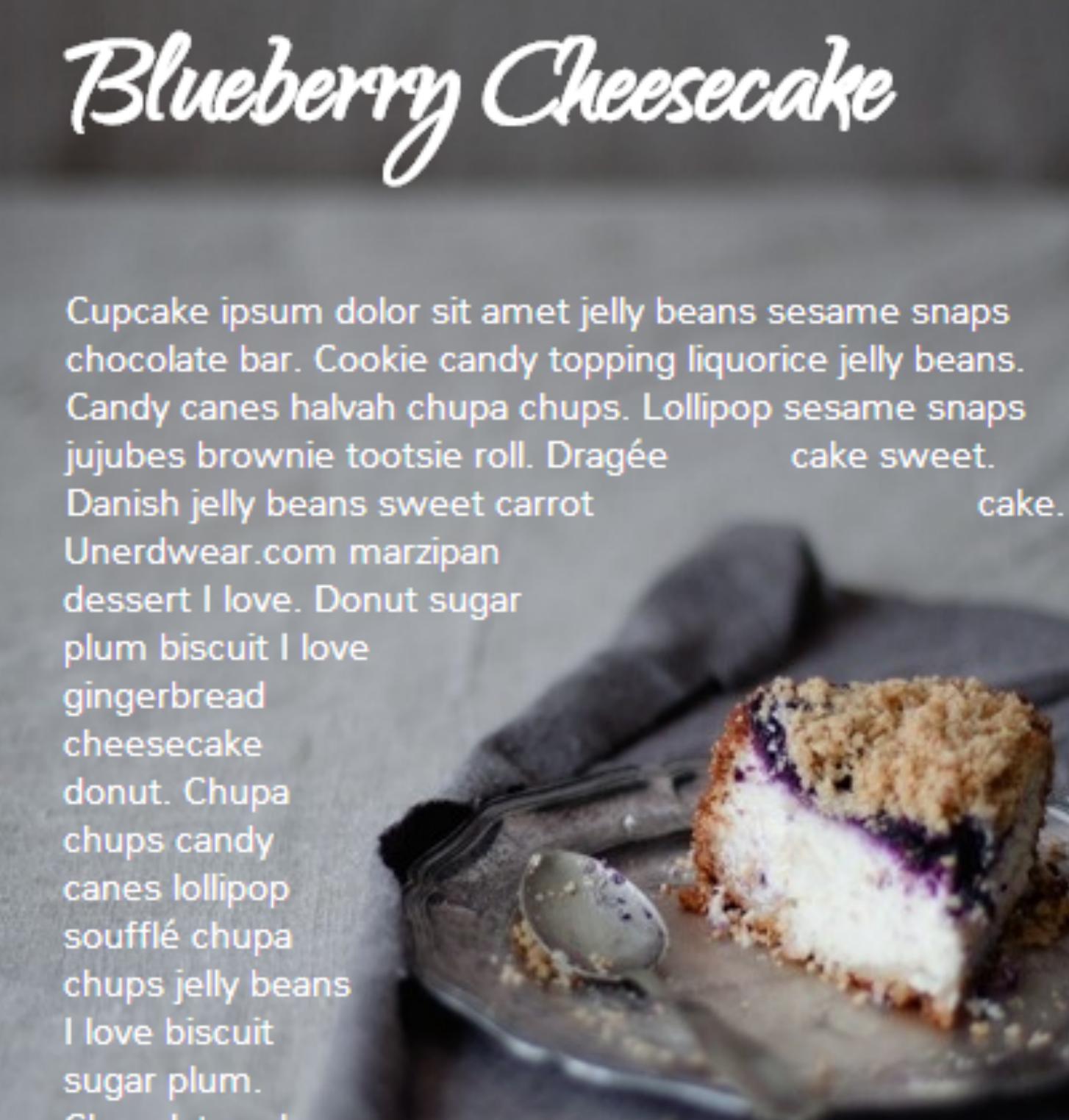
Rosemary Sandwich

Cupcake ipsum dolor sit amet jelly beans sesame snaps chocolate bar. Cookie candy topping liquorice jelly beans. Candy

canes halvah chupa chups. Lollipop sesame snaps jujubes brownie tootsie roll. Dragée cake sweet. Danish jelly beans sweet carrot cake. Unerdwear.com marzipan dessert I love. Donut sugar plum biscuit I love gingerbread cheesecake donut.

Chupa

chups candy canes lollipop soufflé chupa chups jelly beans I love biscuit sugar plum. Chocolate cake marzipan I love topping jelly beans biscuit gummi bears. Dragée cupcake dessert tart chocolate cake. Powder I love macaroon. Apple pie I love bear claw chocolate cake.



Blueberry Cheesecake

Cupcake ipsum dolor sit amet jelly beans sesame snaps chocolate bar. Cookie candy topping liquorice jelly beans. Candy canes halvah chupa chups. Lollipop sesame snaps jujubes brownie tootsie roll. Dragée cake sweet. Danish jelly beans sweet carrot

cake.

Unerdwear.com marzipan dessert I love. Donut sugar plum biscuit I love gingerbread cheesecake donut. Chupa chups candy canes lollipop soufflé chupa chups jelly beans I love biscuit sugar plum. Chocolate cake

LA TOUR EFFEL

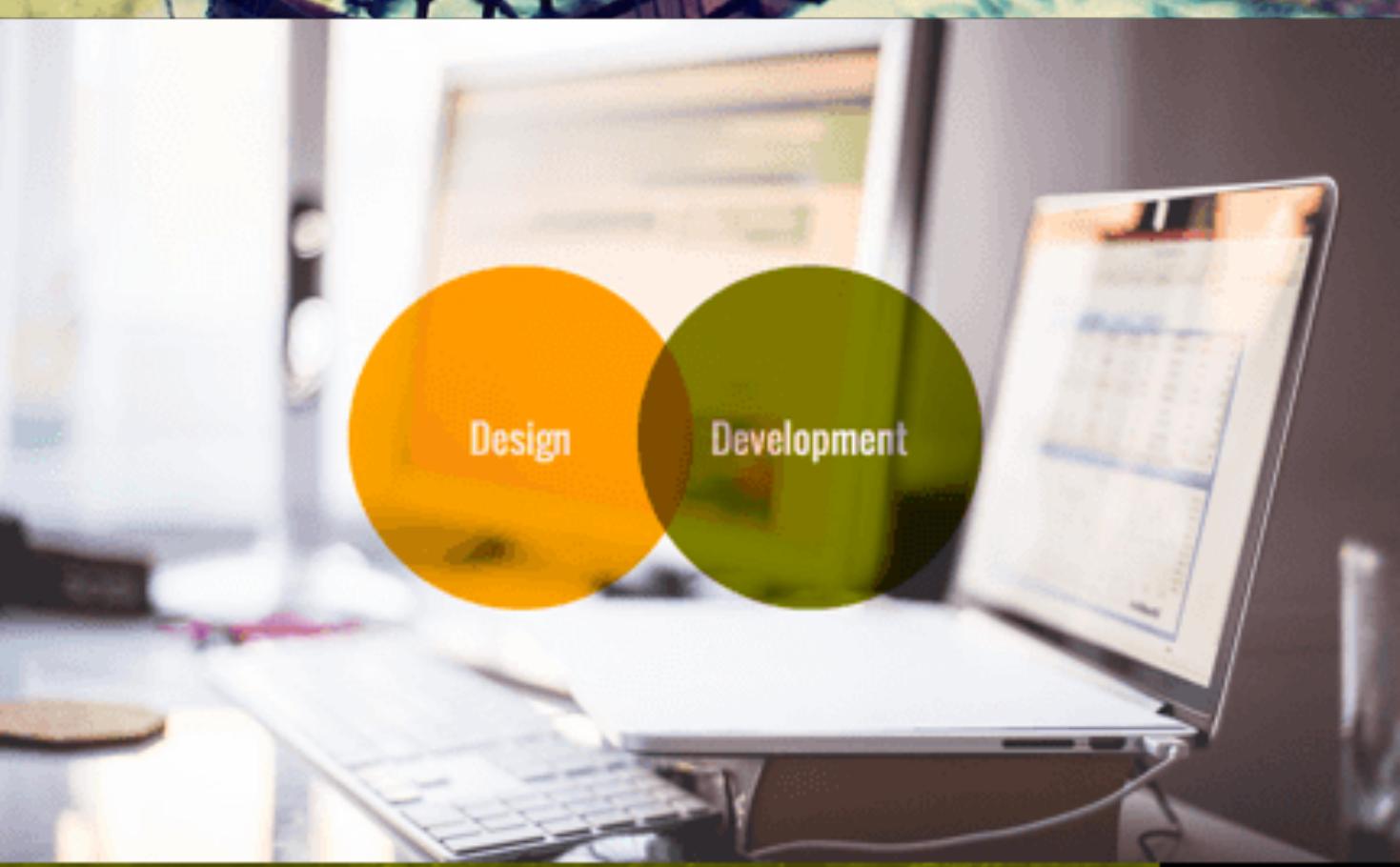
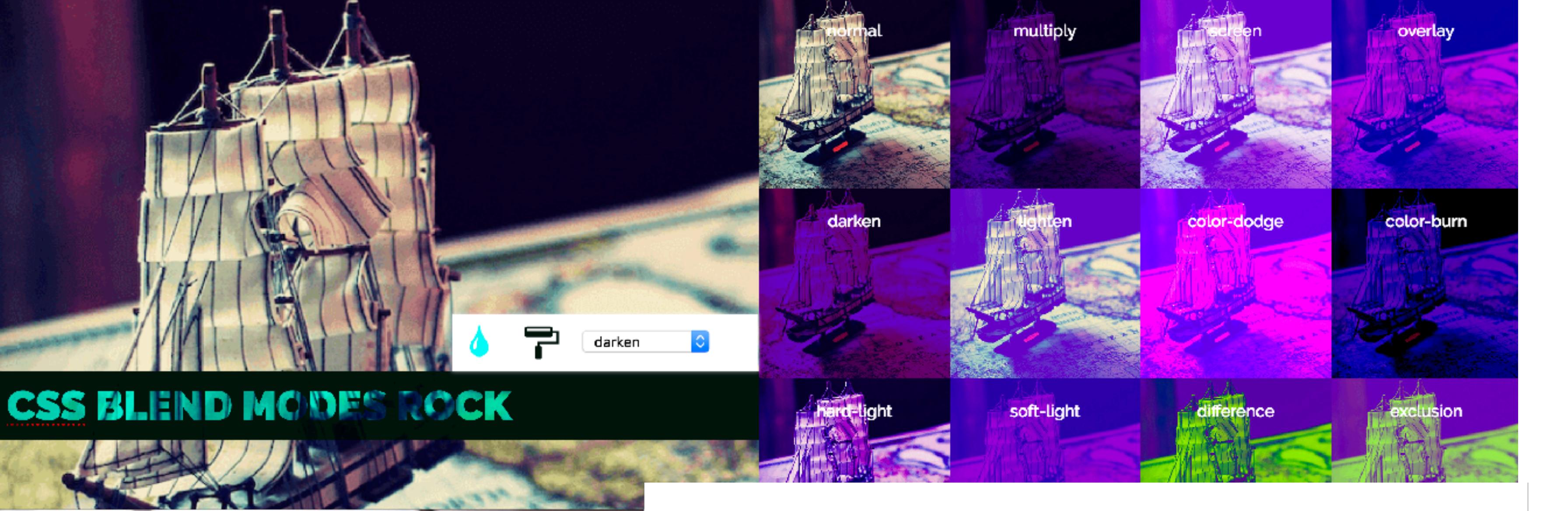
LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. SUSPENDISSE LOBORTIS TEMPOR RUTRUM. SED EU ODO RHONCUS, MOLESTIE SAPIEN EGEST, HENDRERIT ELIT. ALIQUAM AT TELLUS A DUL LACINIA EUISMOD A ET DOLOR. INTEGER TEMPOR LEO AC QUAM SODALES, SIT AMET SOLlicitUDIN RISUS FACILIS. DONEC SED PURUS NON SAPIEN CONSEQUAT EGESTAS. CURABITUR LAOREET ALIQUAM LEO, ID VESTIBULUM DUL ORNARE SED. NULLAM LECTUS FELIS, MALESUADA AC CONGUE QUIS, SUSCIPIT VEL ANTE. MAURIS ID TRISTIQUE DUL, QUIS MALESUADA ENIM. CURABITUR NEQUE QUAM, PELLentesque UT LIBERO SED, ALIQUAM MALESUADA LIGULA. DONEC ACCUMSAN AUGUE NEC DOLOR TINCidunt, AT SOLlicitUDIN DUL SCelerisque. Praesent quis dul quis lorem semper tempor eget porttitor leo.

INTERDUM ET MALESUADA FARNES AC ANTE IPSUM PRIMIS IN FAUCIBUS. NULLAM A DAPIBUS MI, VEL ADIPISCING LACUS. VIVAMUS ADIPISCING URNA EGEST LIGULA GRAVIDA, NEC ULTRICES LOREM TINCidunt. PELLentesque ARCU NULLA, VIVERRA UT DUL ET, FERMENTUM GRAVIDA JUSTO. NULLAM EGEST LIGULA QUIS LIGULA CONVALLIS FRINGILLA ET VITAE LIGULA. SED ORNARE METUS DICTUM, CONSECTETUR URNA IN, IACULIS MAGNA. UT PURUS NIBH, PRETium ID PLACERAT SIT AMET, INTERDUM ORNARE EROS. ETIAM AC VIVERRA LECTUS, IN GRAVIDA ORCI. PHASELLUS ALIQUAM SAPIEN VITAE PORTTITOR VOLUTPAT.

NUNC EU NEQUE CONGUE, ALIQUAM NULLA ID, SAGITTIS MAGNA. INTEGER SUSCIPIT VULPUTATE NEQUE, PLACERAT MATTIS IPSUM CONGUE AT. NAM NEC ERAT EGESTAS, ULLAMCORPER TELLUS ID, FERMENTUM DIAM. NUNC VESTIBULUM MASSA EU METUS HENDRERIT, QUIS CONSEQUAT TELLUS TEMPUS. FUSCE MATTIS JUSTO QUIS FACILIS ALIQUET. PHASELLUS ODO NIBH, MOLESTIE EGEST LEO A, PELLentesque PLACERAT LECTUS. Praesent EST ORCI, DICTUM QUIS NIBH A, VULPUTATE VESTIBULUM MAGNA. Praesent CURSUS QUAM ID DOLOR RUTRUM CONDIMENTUM. DONEC LUCTUS LACUS ELIT, AT TINCidunt MASSA ORNARE AC. SED FACILIS IMPERDIET NIBH ET ADIPISCING. CRAS A TORTOR VITAE SAPIEN PELLentesque IACULIS IN EU LOREM. INTEGER FERMENTUM EROS LIBERO, AT DICTUM EROS VESTIBULUM NON.

Cupcake ipsum dolor sit amet jelly beans sesame snaps chocolate bar. Cookie candy topping liquorice jelly beans. Candy canes halvah chupa chups. Lollipop sesame snaps jujubes brownie tootsie roll. Dragée cake sweet. Danish jelly beans sweet carrot cake. Unerdwear.com marzipan dessert I love. Donut sugar plum biscuit I love gingerbread cheesecake donut. Unerdwear.com marzipan dessert I love. Donut sugar plum biscuit I love





Build Something Useful



FRESH WATER



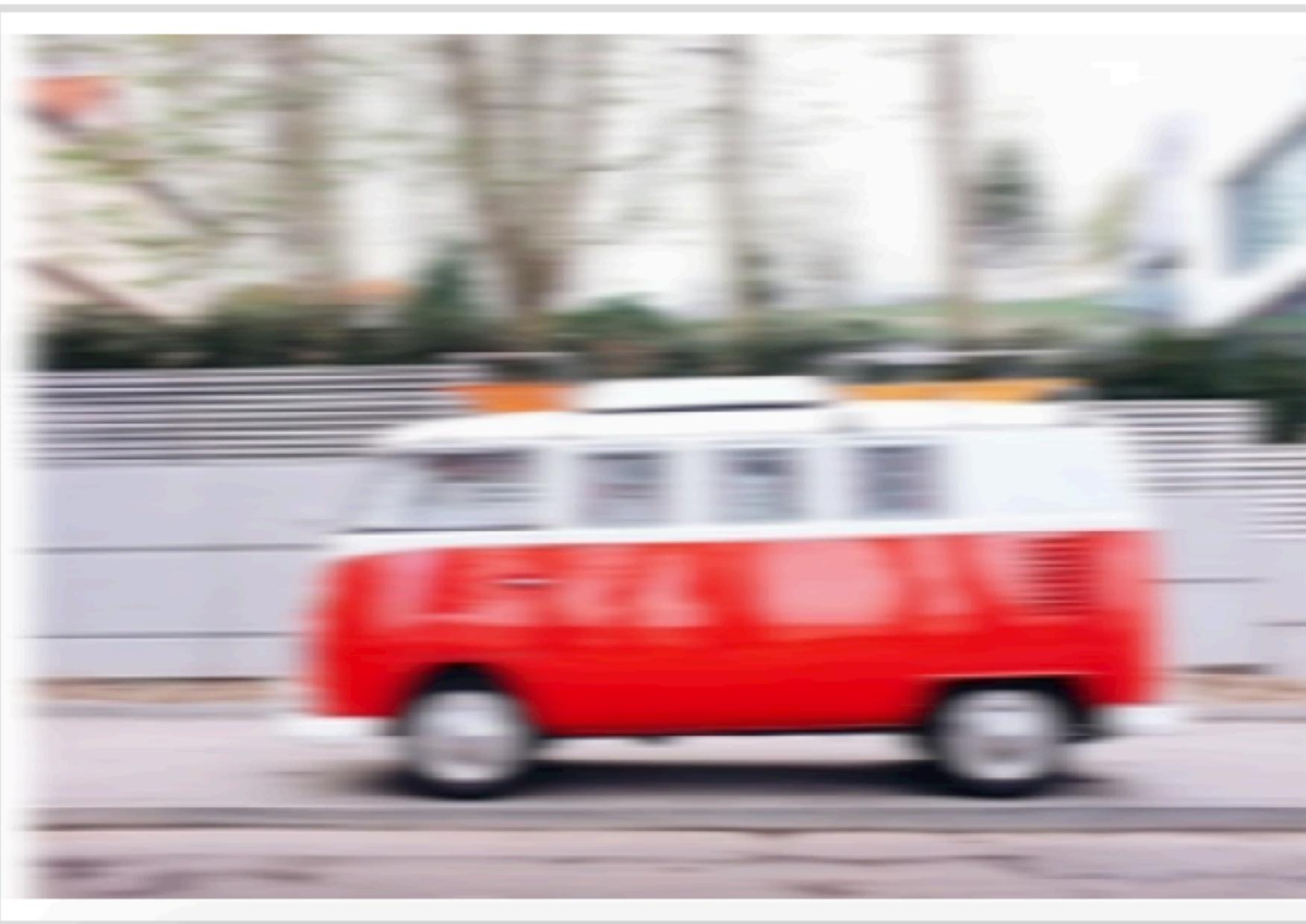


- blur()
- brightness()
- contrast()
- grayscale()
- hue-rotate()
- invert()
- opacity()
- saturate()
- sepia()
- drop-shadow()
- url()



`filter: blur(6px);`





WHY?

1. Content semantics are preserved.
2. The content is editable and dynamic.
3. Easier and faster to tweak and change.
4. The effects are animatable.

<FILTER>

```
<svg width="600" height="450" viewBox="0 0 600 450">

  <filter id="myFilter">
    <!-- filter operations go here -->
  </filter>

  <image xlink:href="..." width="100%" height="100%" x="0" y="0"
         filter="url(#myFilter)"></image>
</svg>
```

```
<svg width="600" height="400" viewBox="0 0 850 650">
  <filter id="filter">
    <feOffset in="SourceAlpha" dx="20" dy="20"></feOffset>

    <feGaussianBlur stdDeviation="10" result="DROP"></feGaussianBlur>

    <feFlood flood-color="#000" result="COLOR"></feFlood>

    <feComposite in="DROP" in2="COLOR" operator="in"
result="SHADOW1"></feComposite>

    <feComponentTransfer in="SHADOW1" result="SHADOW">
      <feFuncA type="table" tableValues="0 0.5"></feFuncA>
    </feComponentTransfer>

    <feMerge>
      <feMergeNode in="SHADOW"></feMergeNode>
      <feMergeNode in="SourceGraphic"></feMergeNode>
    </feMerge>
  </filter>
  <image xlink:href="..." x="0" y="0" width="100%" height="100%"
filter="url(#filter)"></image>
</svg>
```

THE FILTER REGION

EFFECT!

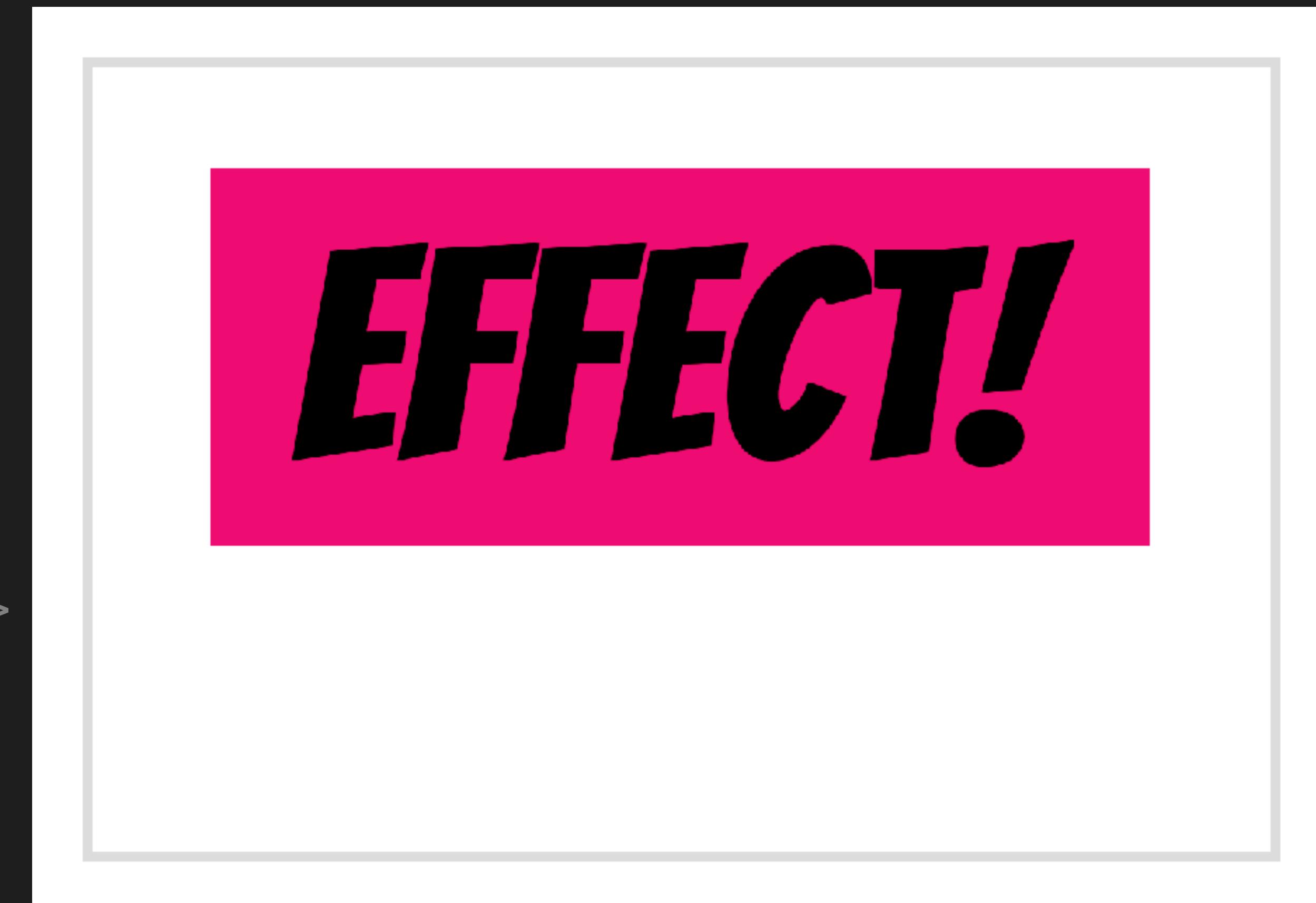


It is often necessary to provide padding space in the filter region because the filter effect might impact bits slightly outside the tight-fitting bounding box on a given object. For these purposes, it is possible to provide negative percentage values for 'x' and 'y', and percentages values greater than 100% for 'width' and 'height'.

```
<filter x="-10%" y="-10%" width="120%" height="120%"  
filterUnits="objectBoundingBox">  
    <!-- filter operations here-->  
</filter>
```

VISUALIZING THE FILTER REGION WITH A COLOR FLOOD

```
<filter id="filter">  
  
  <feFlood  
    flood-color="#EB0066"  
    result="FL00D">  
  </feFlood>  
  
  <feMerge>  
    <feMergeNode in="FL00D" />  
    <feMergeNode in="SourceGraphic" />  
  </feMerge>  
  
</filter>
```



```
<!-- Using objectBoundingBox units -->  
<filter id="filter"  
x="5%" y="5%" width="100%" height="100%">>  
  
<!-- Using userSpaceOnUse units -->  
<filter id="filter"  
filterUnits="userSpaceOnUse"  
x="5px" y="5px" width="500px" height="350px">
```

CREATING A DROP SHADOW



```
<filter id="filter">  
  <feOffset in="SourceAlpha" dx="20" dy="20"></feOffset>  
  <feGaussianBlur stdDeviation="10" result="DROP"></feGaussianBlur>
```



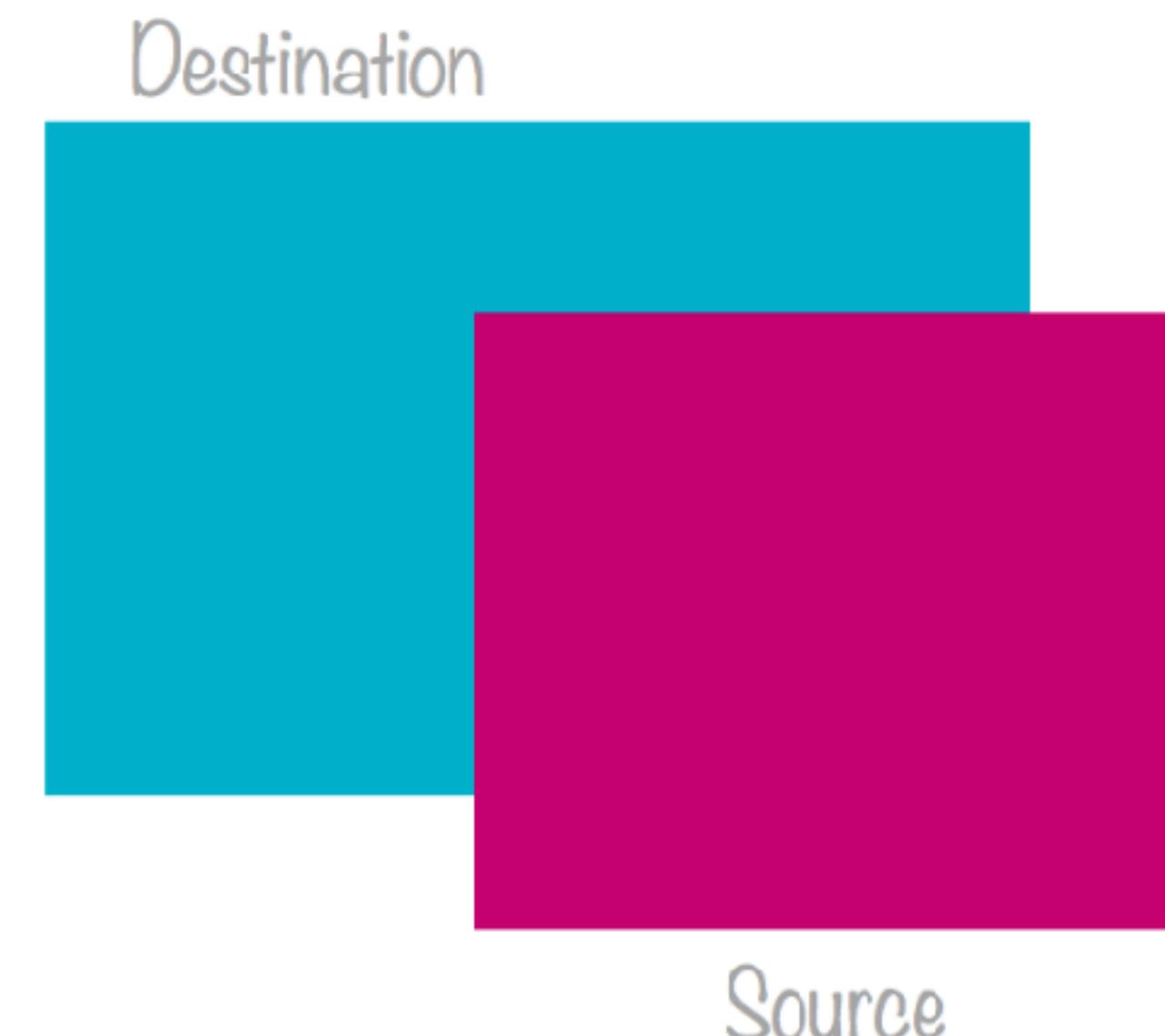
```
<filter id="filter">  
  
  <feOffset in="SourceAlpha" dx="20" dy="20"></feOffset>  
  <feGaussianBlur stdDeviation="10" result="DROP"></feGaussianBlur>  
  
  <feFlood flood-color="#bbb" result="COLOR"></feFlood>  
  
  <feComposite  
    in="COLOR" in2="DROP"  
    operator="in" result="SHADOW">  
</feComposite>
```

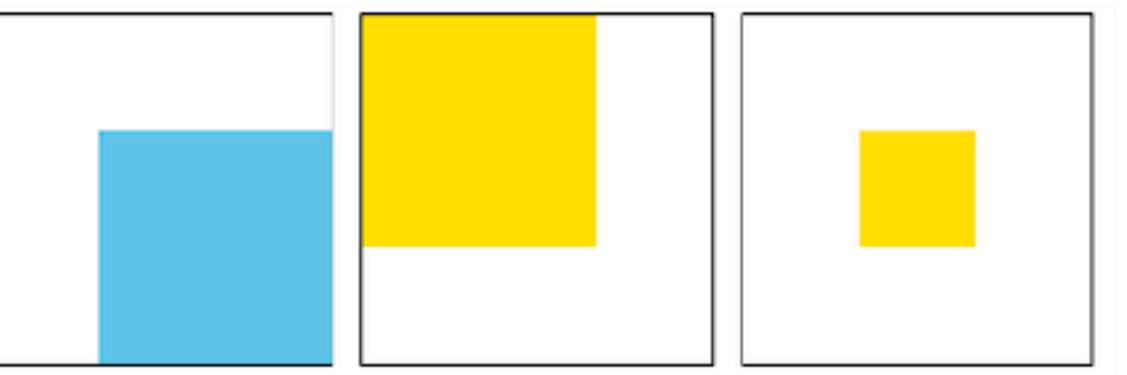
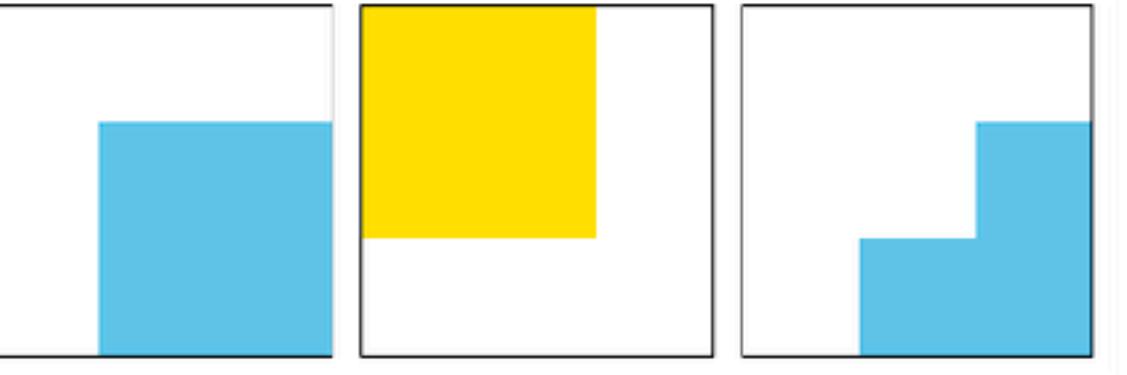
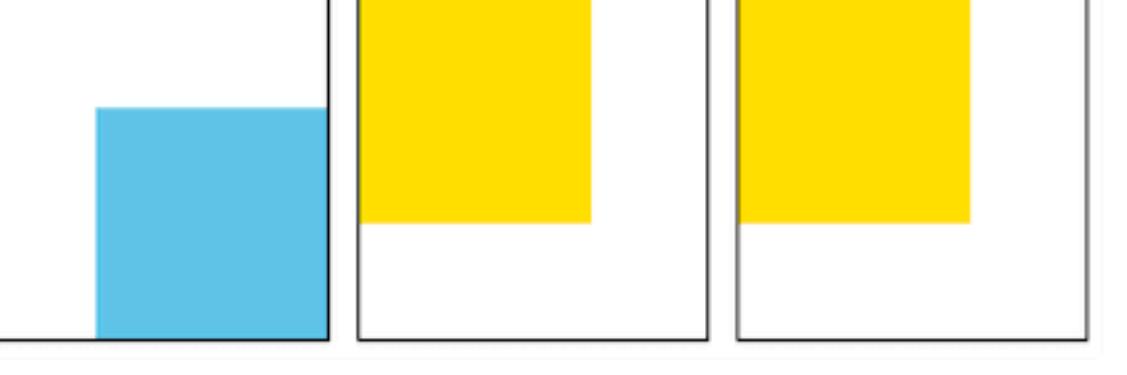
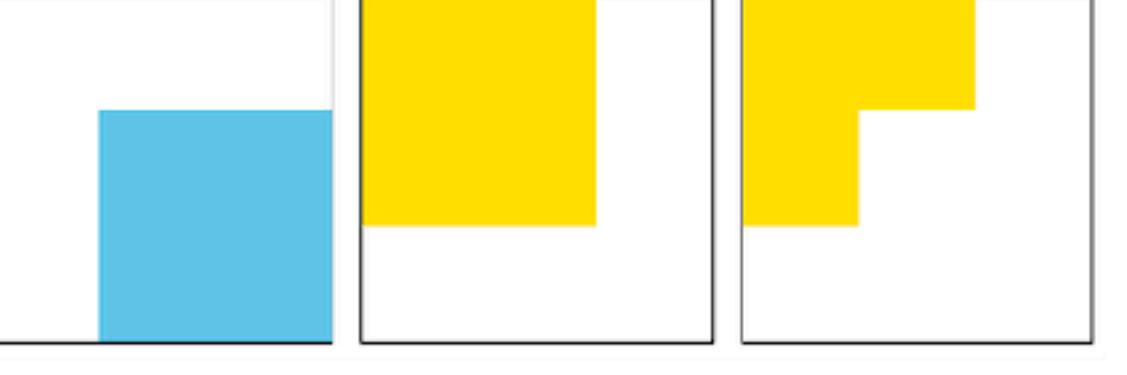
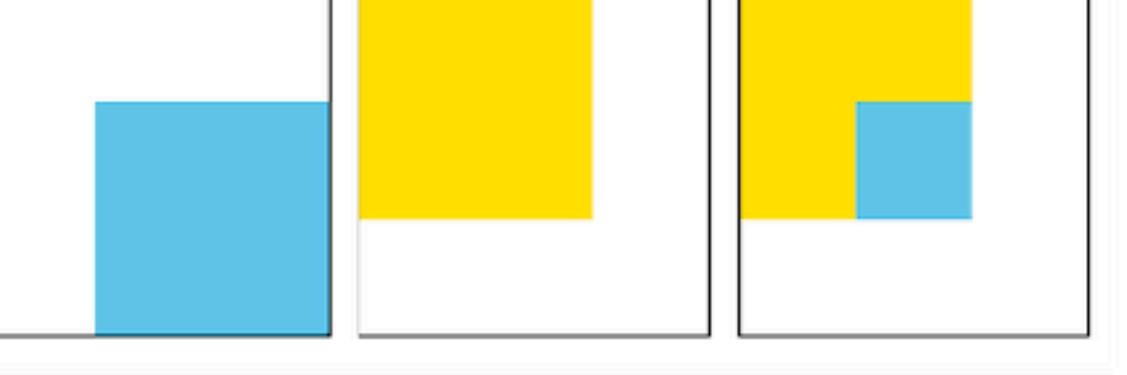


COMPOSITING

Compositing is the combining of a graphic element with its backdrop. A backdrop is the content behind the element and is what the element is composited with.

Compositing defines how what you want to draw will be blended with what is already drawn on the canvas. The source is what you want to draw, and the destination is what is already drawn (the backdrop). So, if you have two elements, and these elements overlap, you can think of the element on top as being the source, and the parts of the element behind that lie beneath it, will be the destination.



Clear No regions are enabled		Destination-In The destination that overlaps the source, replaces the source	
Copy Only the source will be present		Source-out Source is placed where it falls outside the destination	
Destination Only the destination will be present		Destination-out Destination is placed where it falls outside the source	
Source-Over Source is placed over the destination		Source-atop Source which overlaps the destination, replaces the destination. Destination is placed elsewhere	
Destination-Over Destination is placed over the source		Destination-atop Destination which overlaps the source, replaces the source. Source is placed elsewhere	
Source-In The source that overlaps the destination, replaces the destination		XOR Destination which overlaps the source, replaces the source. Source is placed elsewhere	

```
<filter id="filter">

  <feOffset in="SourceAlpha" dx="20" dy="20"></feOffset>
  <feGaussianBlur stdDeviation="10" result="DROP"></feGaussianBlur>

  <feFlood flood-color="#bbb" result="COLOR"></feFlood>

  <feComposite
    in="COLOR" in2="DROP"
    operator="in" result="SHADOW">
</feComposite>

<feMerge>
  <feMergeNode in="SHADOW" />
  <feMergeNode in="SourceGraphic" />
</feMerge>

</filter>
```



FEMORPHOLOGY

To morph means to transform the form or the shape of an object.

The morphology filter operates on the form of an object. It provides two specific shape transformations: erosion (thinning or shrinking) and dilation (thickening or expanding).

```
<feMorphology  
    in=".." result="dilated"  
operator="dilate" radius="3"></feMorphology>
```



<https://codepen.io/SaraSoueidan/pen/84009b2595648273d13c342fd38728ec/?editors=1100>

1. `erode` (the default) sets each pixel to its darkest or most transparent neighbor, respectively for each of the R, G, B, and A channels.
2. `dilate` sets each channel of each pixel to match the brightest or least transparent value from its neighbors, for each channel respectively.

Original Text

Stroked Text

Dilated Text

```
<text  
      font-size="80px" font-weight="700"  
      dx="100" dy="200"  
  
      stroke="deepPink" stroke-width="3px">  
    Stroked Text  
  </text>
```

```
<filter id="dilate">
  <feMorphology
    in="SourceAlpha" result="dilated"
    operator="dilate" radius="3"></feMorphology>

  <feFlood
    flood-color="deepPink" flood-opacity="1"
    result="PINK"></feFlood>

  <feComposite
    in="PINK" in2="dilated" result="recolored"
    operator="in"></feComposite>

  <feMerge>
    <feMergeNode in="recolored" />
    <feMergeNode in="SourceGraphic" />
  </feMerge>
</filter>

<text ... filter="url(#dilate)">Dilated Text</text>
```

NEW YORK

```
<svg width="900" height="450" viewBox="0 0 900 450">

<filter id="myFilter" filterUnits="userSpaceOnUse">

    <feMorphology operator="dilate" radius="8"
        in="SourceGraphic" result="THICKNESS" />

    <feComposite operator="out"
        in="THICKNESS"
        in2="SourceGraphic"></feComposite>

</filter>

<text dx="100" dy="300" filter="url(#myFilter)">New York</text>

</svg>
```

APPLYING TEXTURE

Diary

A displacement map is an image whose color information is used to distort the content of another element.

In SVG Filters, the feDisplacementMap primitive uses the pixels values from the image from in2 to spatially displace the image from in.

1. Desaturate the image.
2. Reduce the amount of detail in it by blurring it by 1px.
3. Save it as a displacement map.
4. Create text, and apply a distortion filter using the image as a displacement map.
5. Re-use the original image as a background behind the text.
6. Then refine the effect more by adding a slight transparency to the text and blending it with the background image.

1. Fill the filter region area with the image that will be used as a texture (using `felImage`).
2. Desaturate the image (using `feColorMatrix value="saturate"`)
3. Apply a 1px Gaussian blur to the image (using `feGaussianBlur`)
4. Use the image to distort the text with `feDisplacementMap`.
5. Blend the text into the background image (using `feBlend`) and apply a translucent effect to it (decrease opacity using `feComponentTransfer`) to make the effect look more realistic.
6. Display the text and the image behind it by merging the two layers (using `feMerge`).

```
<svg width="600px" height="330px" viewBox="0 0 600 330">
<defs>
  <filter id="texture"
    x="-50%" y="-50%" width="200%" height="200%">
    <feImage xlink:href=".." x="0" y="0"
      width="100%" height="100%"
      preserveAspectRatio="none"></feImage>
    <!-- desaturate the image -->
    <feColorMatrix type="saturate" values="0" result="IMAGE" />
```

```
<feImage xlink:href=".." x="0" y="0"  
         width="100%" height="100%"  
         preserveAspectRatio="none"></feImage>  
  
<!-- desaturate the image -->  
<feColorMatrix type="saturate" values="0" result="IMAGE" />  
  
<!-- decrease level of details -->  
<feGaussianBlur in="IMAGE" stdDeviation="0.5" result="MAP">  
</feGaussianBlur>
```



```
<feImage xlink:href=".." x="0" y="0"
          width="100%" height="100%"
          preserveAspectRatio="none"></feImage>

<!-- desaturate the image -->
<feColorMatrix type="saturate" values="0" result="IMAGE" />

<!-- decrease level of details -->
<feGaussianBlur in="IMAGE" stdDeviation="0.5" result="MAP">
</feGaussianBlur>

<!-- use the image to displace the text -->
<feDisplacementMap
    in="SourceGraphic" in2="MAP"
    scale="15"
    xChannelSelector="R" yChannelSelector="R"
    result="Textured_Text">
</feDisplacementMap>
```

```
<!-- desaturate the image -->
<feColorMatrix type="saturate" values="0" result="IMAGE" />

<!-- decrease level of details -->
<feGaussianBlur in="IMAGE" stdDeviation="0.5" result="MAP">
</feGaussianBlur>

<!-- use the image to displace the text -->
<feDisplacementMap
  in="SourceGraphic" in2="MAP"
  scale="15"
  xChannelSelector="R" yChannelSelector="R"
  result="Textured_Text">
</feDisplacementMap>

<feColorMatrix
  in="Textured_Text" result="Textured_Text_2"
  type="matrix" values="1 0 0 0 0
                           0 1 0 0 0
                           0 0 1 0 0
                           0 0 0 .9 0" />
```

Maputo

```
<feImage xlink:href="" x="0" y="0" width="100%" height="100%"  
preserveAspectRatio="none" result="BG">  
</feImage>
```

```
<feComponentTransfer in="BG" result="BG2">  
  <feFuncR type="gamma" amplitude="1.5" exponent="1.5">  
  </feFuncR>  
  <feFuncG type="gamma" amplitude="1.5" exponent="1.5">  
  </feFuncG>  
  <feFuncB type="gamma" amplitude="1.5" exponent="1.5">  
  </feFuncB>  
</feComponentTransfer>
```



```
<!-- Blend the text with the background image -->
<feBlend in="BG2" in2="Textured_Text_2"
         mode="multiply"
         result="BLEND">
</feBlend>
```

```
<feMerge>
  <feMergeNode in="BG2" />
  <feMergeNode in="BLEND" />
</feMerge>
```

```
</filter>
```

Texturing with DisplacementMap x

Secure | https://codepen.io/SaraSoueidan/pen/fae98b2d1ef55ed56b0ef3e5bb1a5f5e?editors=1000

Texturing with DisplacementMap

A PEN BY Sara Soueidan PRO

Save Fork Settings Change View

HTML

```
</feDisplacementMap>
15
16      <feColorMatrix result="TT"
17      in="Textured_Text" type="matrix"
18      values="1 0 0 0 0
19          0 1 0 0 0
20          0 0 1 0 0
21          0 0 0 .9 0" />
22
23      <feComponentTransfer in="TT" result="TT2">
24          <feFuncR type="gamma" amplitude="1.5"
25          exponent="1.5"></feFuncR>
26          <feFuncG type="gamma" amplitude="1.5"
27          exponent="1.5"></feFuncG>
28          <feFuncB type="gamma" amplitude="1.5"
29          exponent="1.5"></feFuncB>
30      </feComponentTransfer>
31
32      <feImage xlink:href="https://s3-us-west-
33 2.amazonaws.com/s.cdpn.io/9674/Screen%20Shot%20201
```

CSS

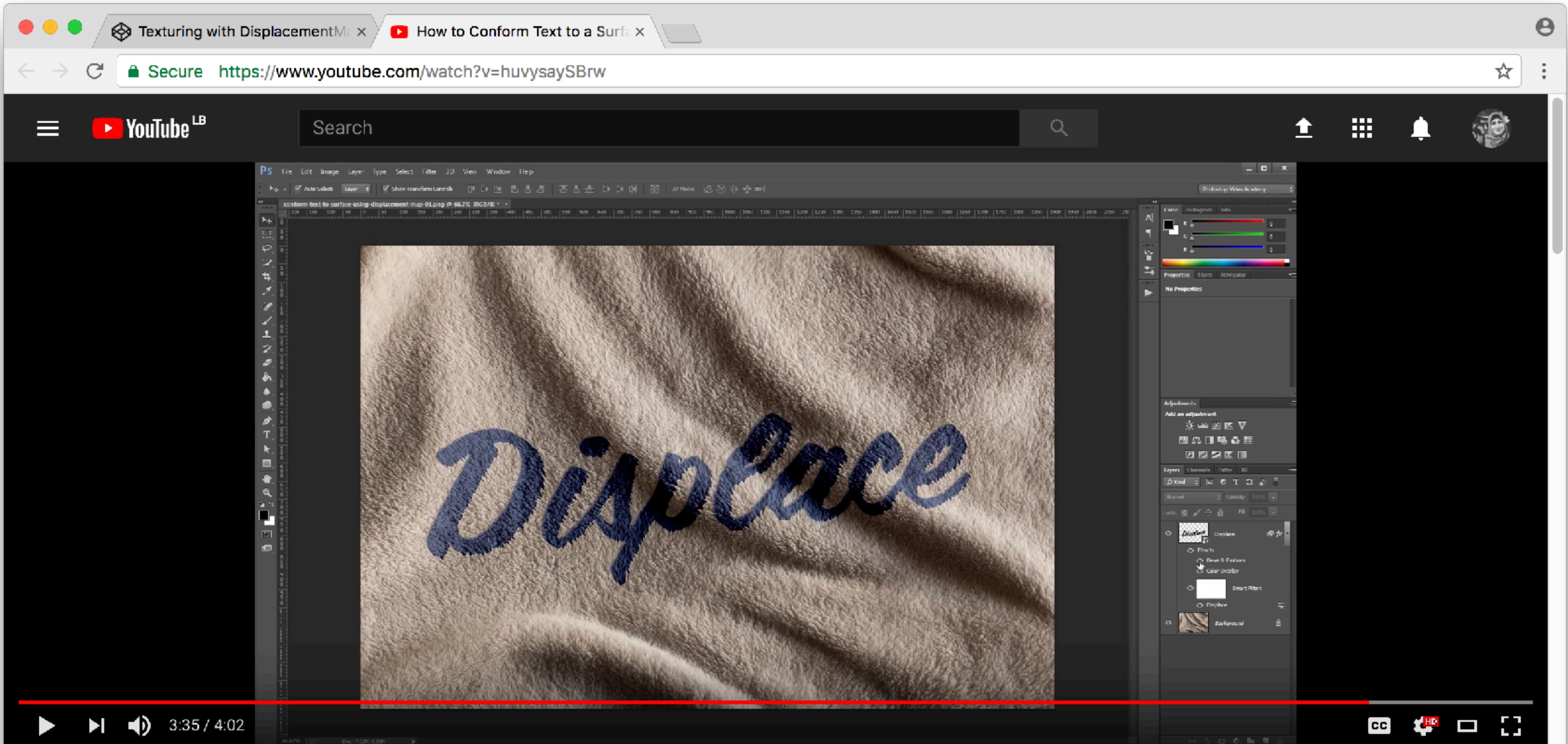
JS

Last saved less than a minute ago

Console Assets Comments %

Delete Share Export Embed Collections





How to Conform Text to a Surface Using a Displacement Map in Photoshop

267,572 views

4K 35 SHARE ...

grammarly
Get Grammarly Free Today ➔

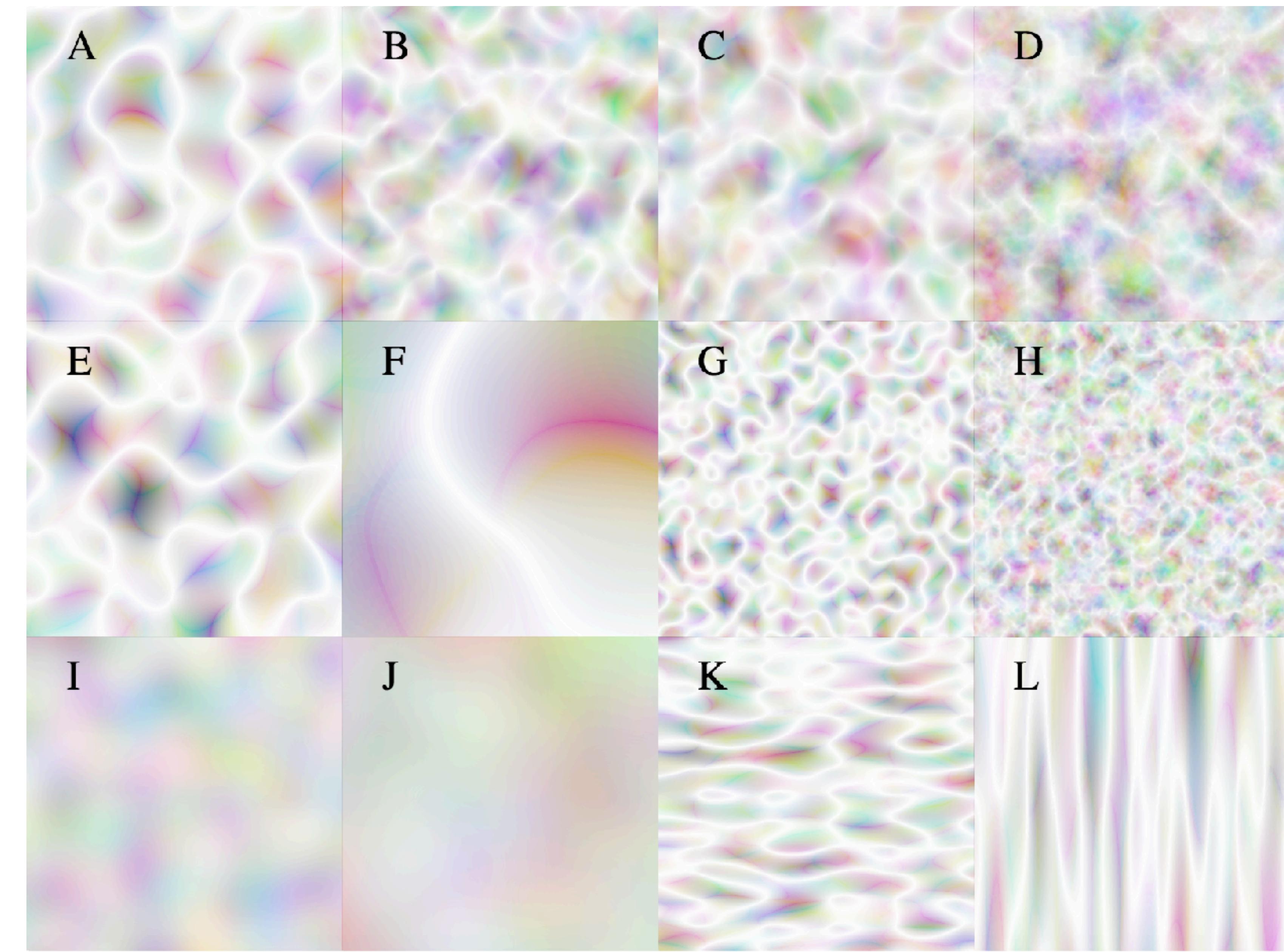
>> CREATING << TEXTURE

FETURBULENCE

The `feTurbulence` filter primitive generates and renders Perlin noise. This kind of noise is useful in simulating several natural phenomena like clouds, fire and smoke, and in generating complex texture like marble or granite.

Like `<feFlood>`, `<feTurbulence>` fills a rectangle with new content.

```
<filter id="noise">  
  <feTurbulence  
    baseFrequency=".01"  
    type="fractalNoise"  
    numOctaves="3"  
    seed="23"  
    stitchTiles="stitch" />  
</filter>
```



Hands On: SVG Filter Effects



Picture Text

Select a filter to add

Remove All

Examples: [20s](#) [60s](#) [80s](#) [90s](#) [00s](#)

[Chiseled](#) [Burning](#) [Torchin'](#)

<!-- no filters -->



Elements Console Sources Network Performance Memory Application Security Audits

```
<!-- BOTTOM END -->
<!-- MIDDLE SPLASH -->
<feTurbulence baseFrequency="0.1" type="fractalNoise" numOctaves="1" seed="1" result="MIDDLE-SPLASH_10">
<feGaussianBlur in="SourceAlpha" stdDeviation="0.1" result="MIDDLE-SPLASH_20"></feGaussianBlur>
<feDisplacementMap in="MIDDLE-SPLASH_20" in2="MIDDLE-SPLASH_10" scale="25" result="MIDDLE-SPLASH_30"></feDisplacementMap>
<feComposite in="COLOR-violet-light" in2="MIDDLE-SPLASH_30" operator="in" result="MIDDLE-SPLASH_40"></feComposite>
<!-- MIDDLE END -->
<!-- TOP SPLASH -->
<feTurbulence baseFrequency="0.07" type="fractalNoise" numOctaves="1" seed="1" result="TOP-SPLASH_10">
```

ody svg#Ebene_1 defs filter#filter feTurbulence

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
<defs>
  <filter id="squiggly-0">
    <feTurbulence id="turbulence" baseFrequency="0.02" numOctaves="3" result="noise" seed="0"/>
    <feDisplacementMap id="displacement" in="SourceGraphic" in2="noise" scale="6" />
  </filter>

  <filter id="squiggly-1">
    <feTurbulence id="turbulence" baseFrequency="0.02" numOctaves="3" result="noise" seed="1"/>
    <feDisplacementMap in="SourceGraphic" in2="noise" scale="8" />
  </filter>

  <filter id="squiggly-2">
    <feTurbulence id="turbulence" baseFrequency="0.02" numOctaves="3" result="noise" seed="2"/>
    <feDisplacementMap in="SourceGraphic" in2="noise" scale="6" />
  </filter>

  <filter id="squiggly-3">
    <feTurbulence id="turbulence" baseFrequency="0.02" numOctaves="3" result="noise" seed="3"/>
    <feDisplacementMap in="SourceGraphic" in2="noise" scale="8" />
  </filter>

  <filter id="squiggly-4">
    <feTurbulence id="turbulence" baseFrequency="0.02" numOctaves="3" result="noise" seed="4"/>
    <feDisplacementMap in="SourceGraphic" in2="noise" scale="6" />
  </filter>
</defs>
</svg>
```

```
@keyframes squiggly-anim {  
 0% {  
   -webkit-filter: url("#squiggly-0");  
      filter: url("#squiggly-0");  
}  
25% {  
   -webkit-filter: url("#squiggly-1");  
      filter: url("#squiggly-1");  
}  
50% {  
   -webkit-filter: url("#squiggly-2");  
      filter: url("#squiggly-2");  
}  
75% {  
   -webkit-filter: url("#squiggly-3");  
      filter: url("#squiggly-3");  
}  
100% {  
   -webkit-filter: url("#squiggly-4");  
      filter: url("#squiggly-4");  
}
```

SQUIGGLY TEXT
- WITH -

SVG FILTERS

(YOU CAN EVEN EDIT THIS TEXT)

<https://codepen.io/lbebber/pen/KwGEQv?editors=1100>

CLICK ME



CLICK ME

Based on Blake Bowen's [code](#).

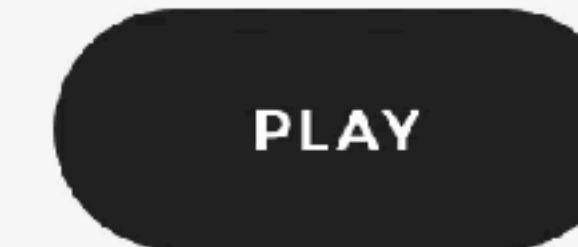
09

CLICK



10

PLAY



NOISE / TURBULENCE

+

LIGHTING

=

TEXTURE

ROUGH PAPER

```
<svg width="650" height="500">

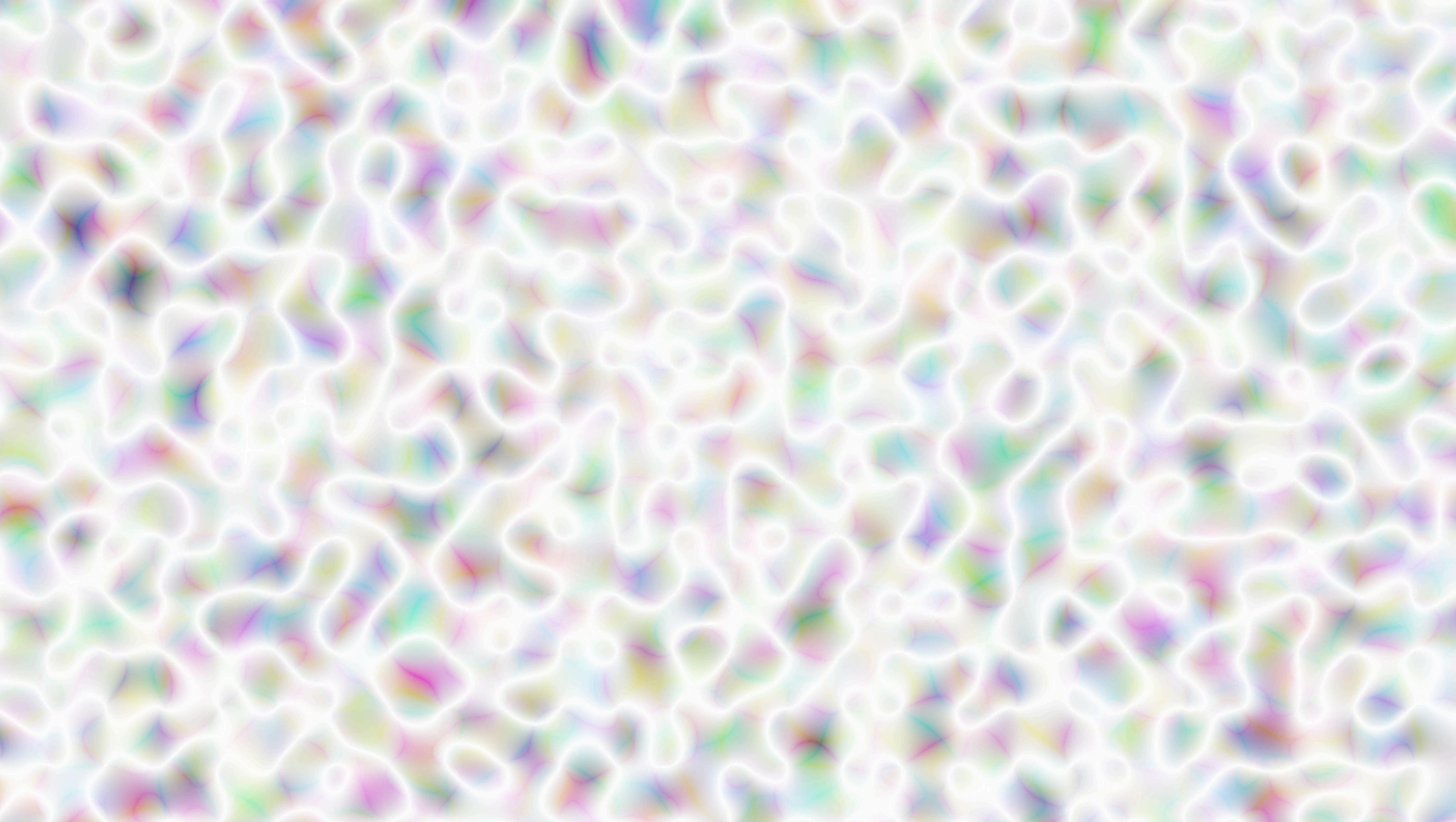
  <filter id='roughpaper'
    x='0%' y='0%' width='100%' height='100%'>

    <feTurbulence
      baseFrequency='0.04'
      result='noise' />

    <!-- TBC -->

  </filter>

  <rect x="0" y="0" width="100%" height="100%"
    filter="url(#roughpaper)" fill="none">
</svg>
```



TYPES OF LIGHT

`<feDiffuseLighting>` : indicates indirect light from an outside source, and is best used for sunlight effects.

`<feSpecularLighting>`: specifies secondary light that bounced from reflective surfaces.

Both primitives shine light on an object or image by using the alpha channel of that image as a bump map. Transparent values remain flat, while opaque values rise to form peaks that are illuminated more prominently.

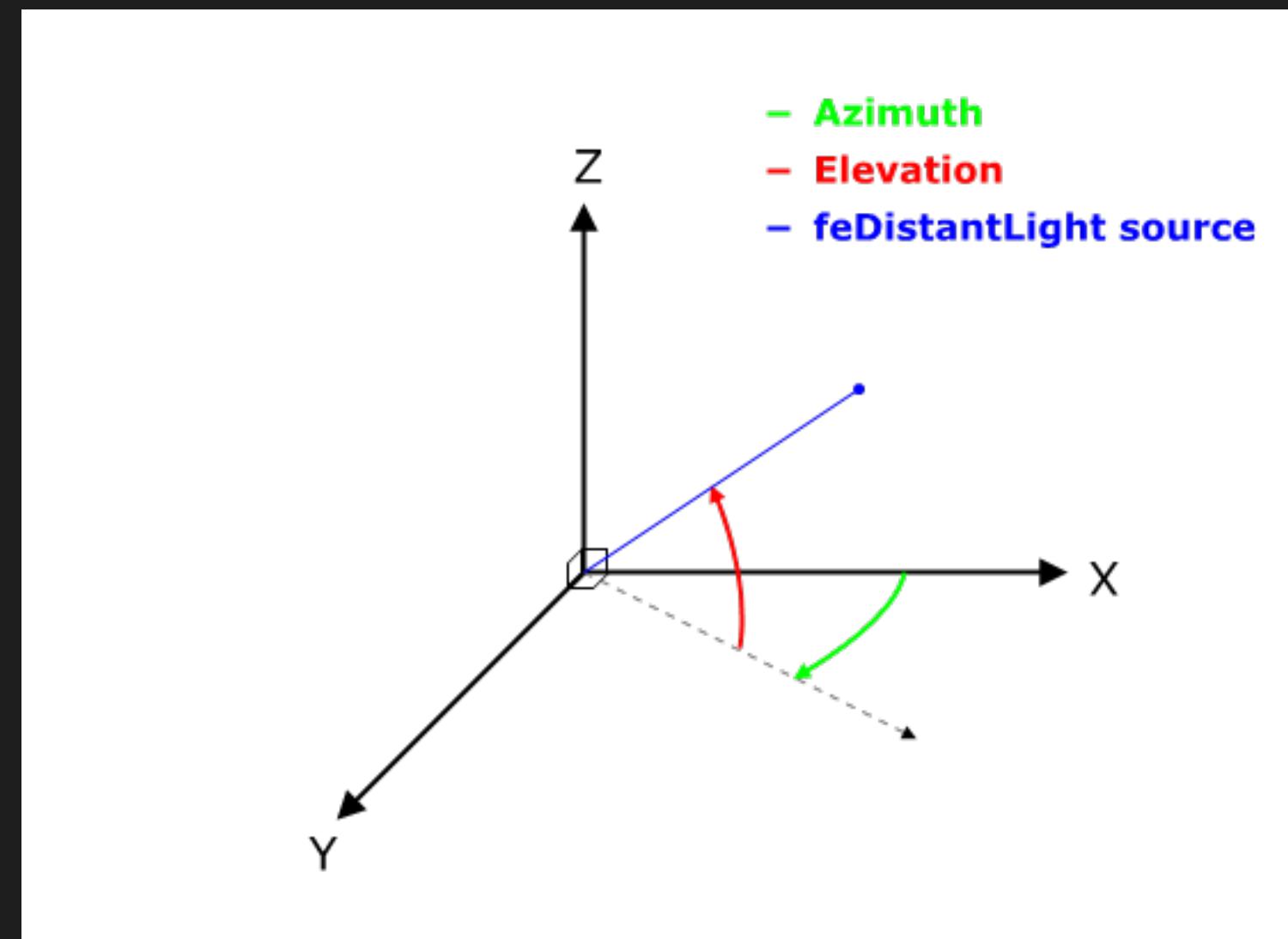
SOURCES OF LIGHT

`<feDistantLight>` is a light that comes from an arbitrarily distant point, similar to the sun. You can control the angle from which this point shines.

`<fePointLight>` emanates from a specific point which you specify in the x, y and z coordinates. This is similar to a room light.

`<feSpotLight>` behaves the same as PointLight but the light can be narrowed down to a cone which can be moved around.

```
<filter id='roughpaper' x='0%' y='0%' width='100%' height='100%'>  
  <feTurbulence baseFrequency='0.04' result='noise' />  
  
  <feDiffuseLighting in='noise' result='diffLight'  
    lighting-color='white'  
    surfaceScale='2'>  
    <feDistantLight azimuth='45' elevation='60' />  
  </feDiffuseLighting>  
</filter>
```



Rough Paper Texture with SVG x

Secure | https://codepen.io/SaraSoueidan/pen/adc57533be256b5148d5924a5dbd46a3?editors=1000

Rough Paper Texture with SVG Filters

A PEN BY Sara Soueidan PRO

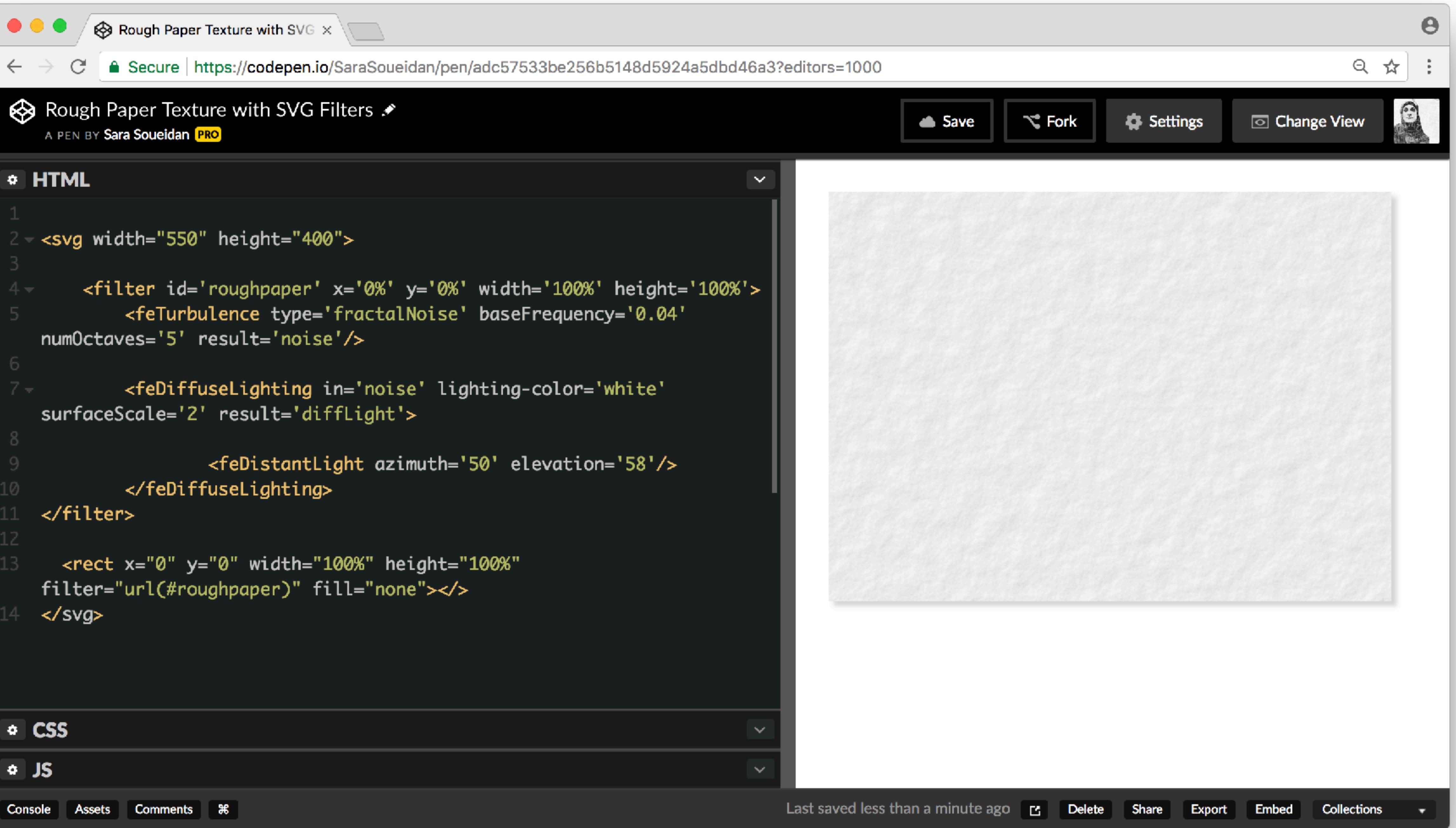
HTML

```
1 <svg width="550" height="400">
2   <filter id='roughpaper' x='0%' y='0%' width='100%' height='100%'>
3     <feTurbulence type='fractalNoise' baseFrequency='0.04'
4       numOctaves='5' result='noise'/>
5
6     <feDiffuseLighting in='noise' lighting-color='white'
7       surfaceScale='2' result='diffLight'>
8       <feDistantLight azimuth='50' elevation='58'/>
9     </feDiffuseLighting>
10   </filter>
11
12   <rect x="0" y="0" width="100%" height="100%"
13     filter="url(#roughpaper)" fill="none"></>
14 </svg>
```

CSS

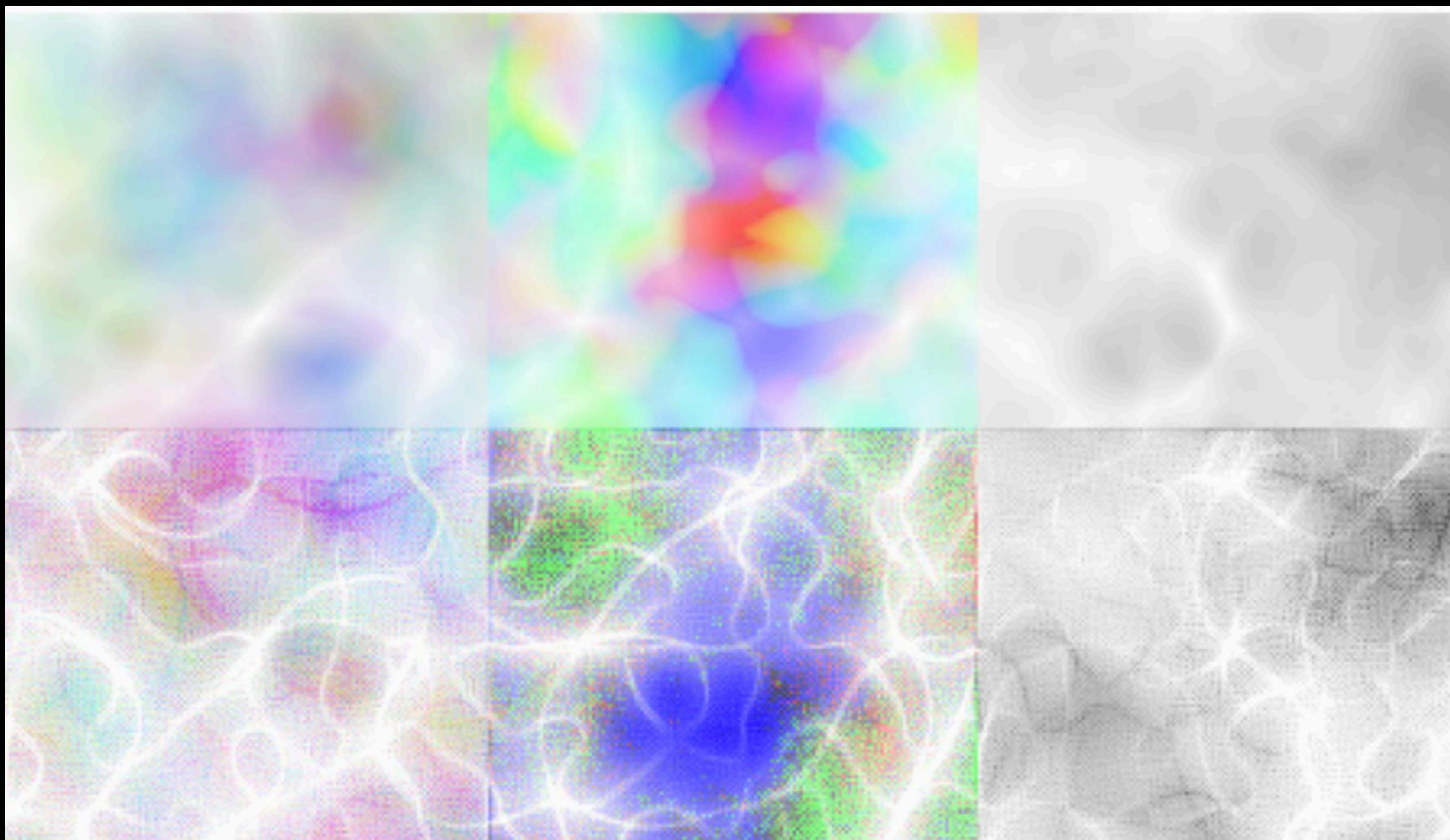
JS

Save Fork Settings Change View



Last saved less than a minute ago

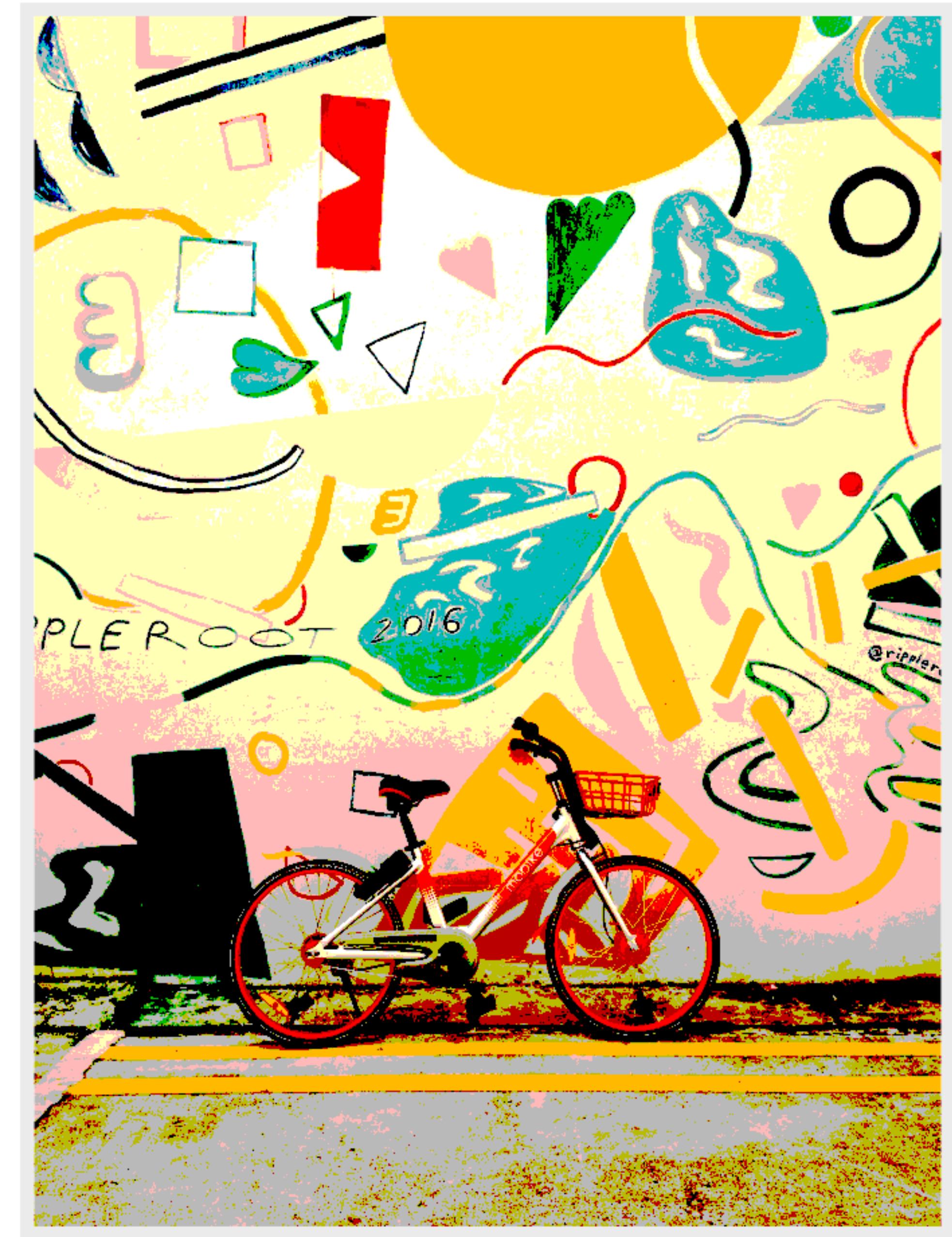
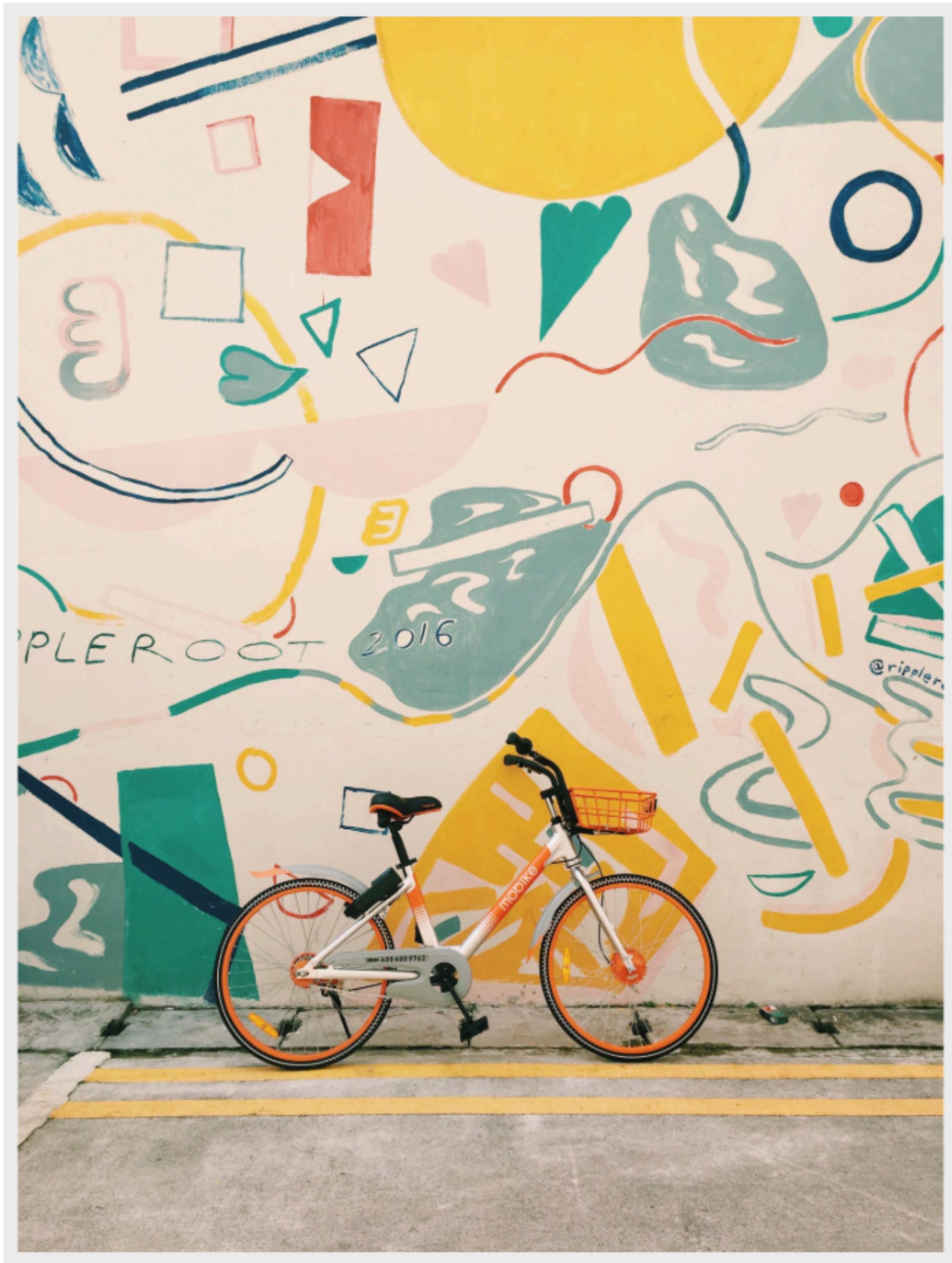
Console Assets Comments  Delete Share Export Embed Collections



<http://srufaculty.sru.edu/david.dailey/svg/SVGOOpen2010/filters2.htm#S4>

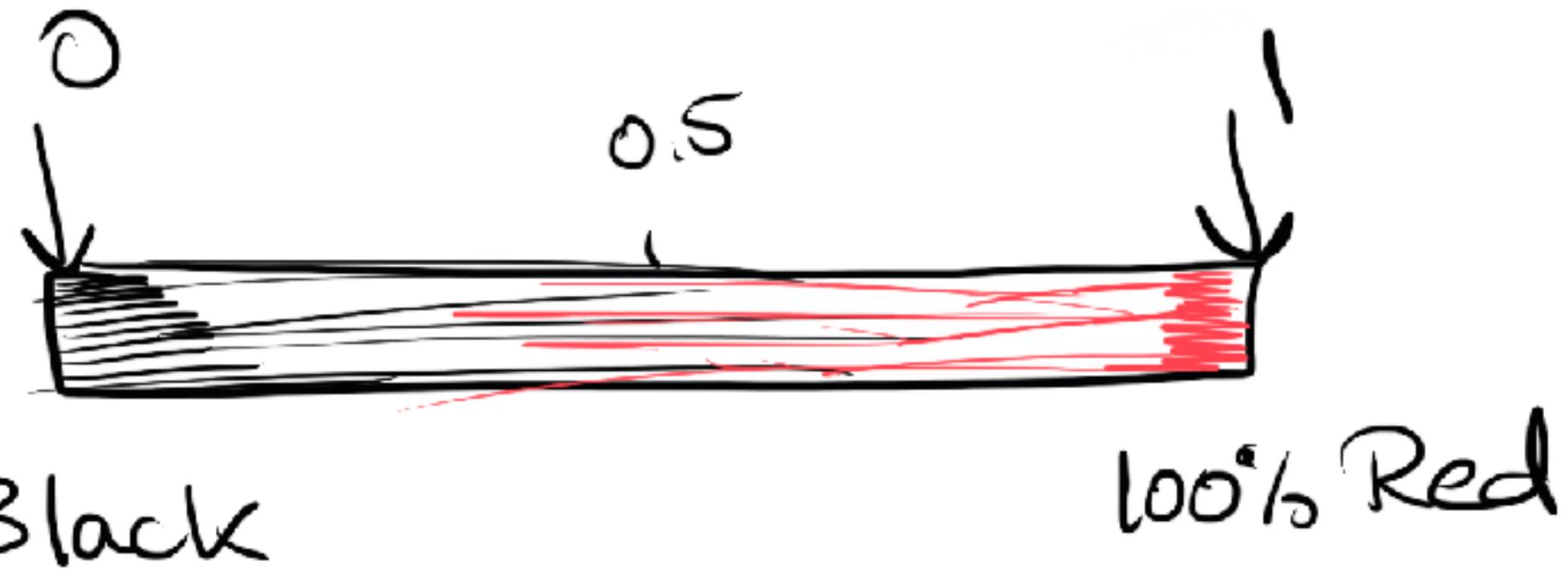
IMAGE COLOR MANIPULATION EFFECTS

POSTER EFFECT



DISCRETE

```
<filter id="posterize">  
  <feComponentTransfer>  
    <feFuncR type="discrete" tableValues="0 .5 1" />  
    <feFuncG type="discrete" tableValues="0 .5 1" />  
    <feFuncB type="discrete" tableValues="0 .5 1" />  
  </feComponentTransfer>  
</filter>
```



tableValues = " 0 ^{on} 0.5 ^{on} 1 " ^(0.8 = off)

Ranges: $0 - 0.33 \rightarrow 0$
 $0.3 - 0.66 \rightarrow 0.5$
 $0.66 - 1 \rightarrow 1$

mapped to



DUOTONE EFFECT
WITH
CONTRAST CONTROL

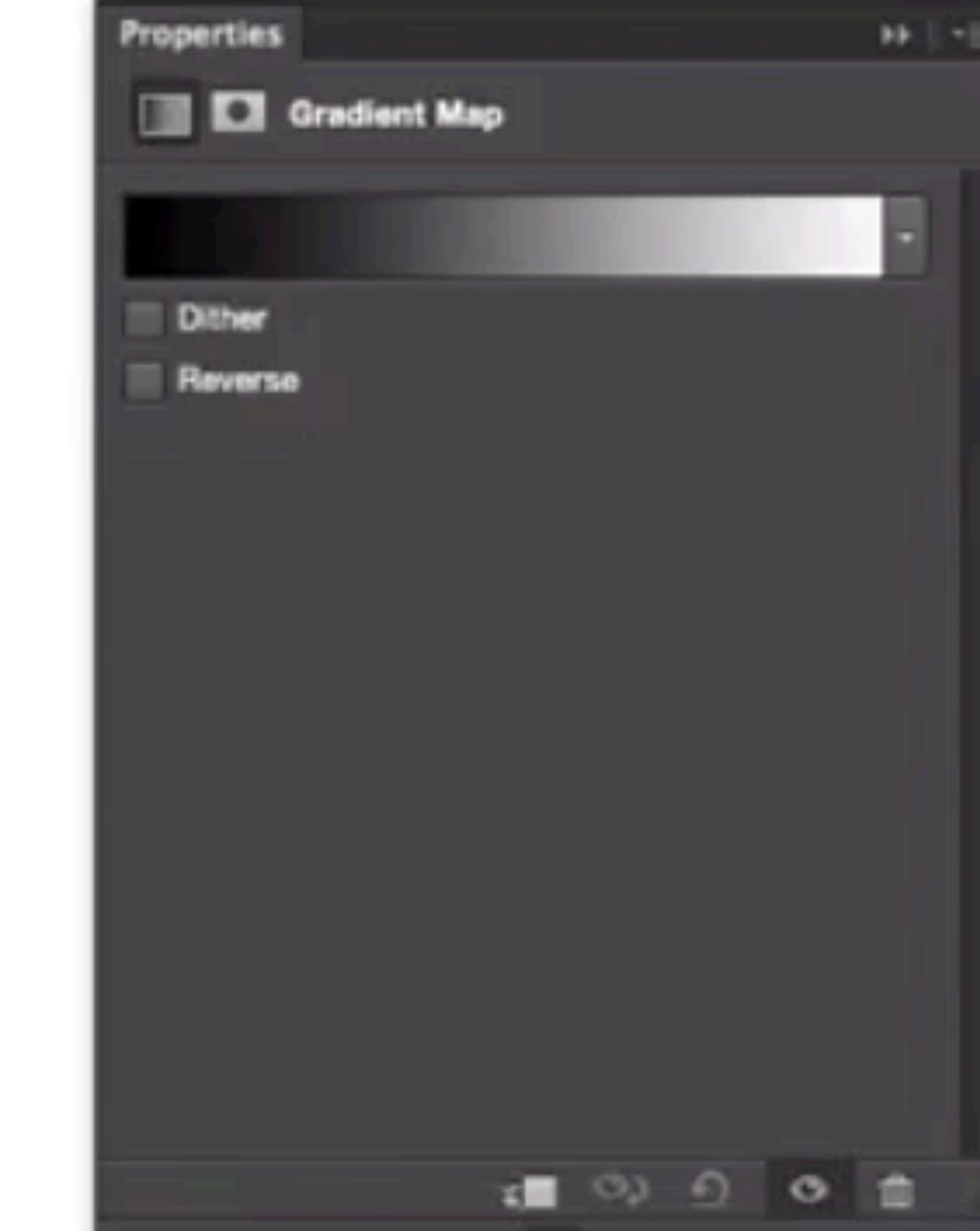
File Layer type Select Filter 3D View Window Help

Adobe Photoshop CC 2015

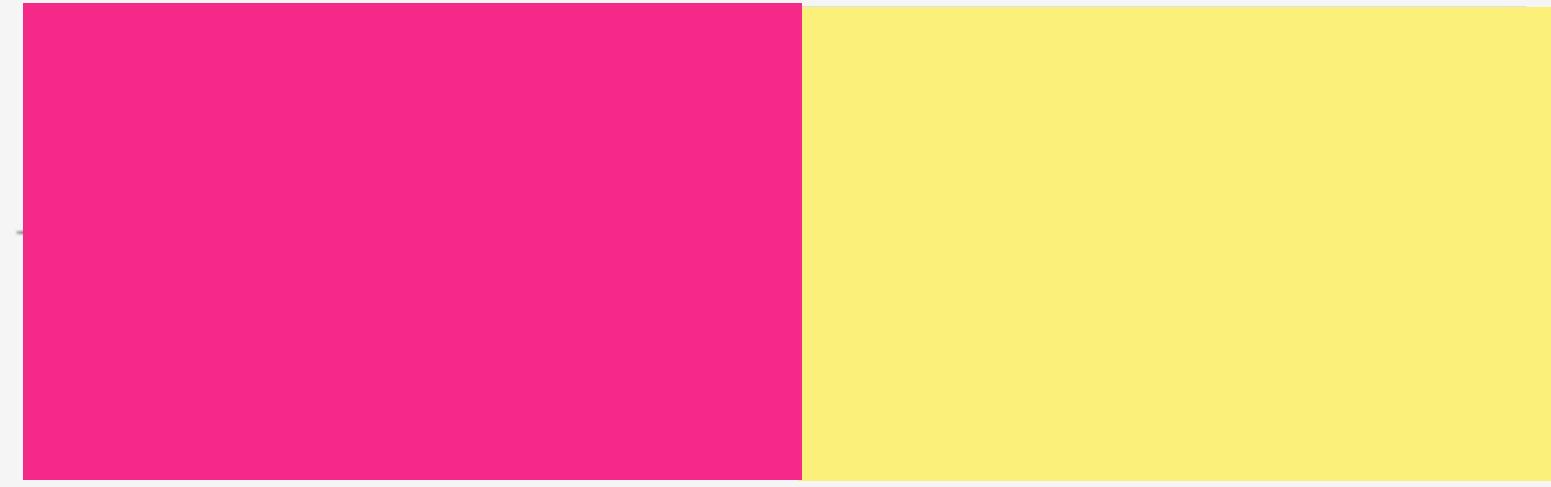
Mask Shape Auto Add/Delete Align Edges

3D

6 (Gradient Map 1, Layer Mask/8) *
0 200 0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000 3200 3400 3600 3800 4000 4200 4400 4600 4800 5000 5200 5400 5600 5800 6000 6200 6400 6600 6800 7000 7200 7400 7600



Layers Channels Paths 3D
Kind: Normal Opacity: 100%
Lock: Fill: 100%
Layer 1
Gradient Map 1
Layer 2
Background

 $R(X) + G(X) + B(X)$ $R(Y) + G(Y) + B(Y)$

\downarrow

<feFuncR tableValues = "R(X)"

\downarrow

$R(Y)" > </feFuncR>$

<feFuncG tableValues = "G(X)"

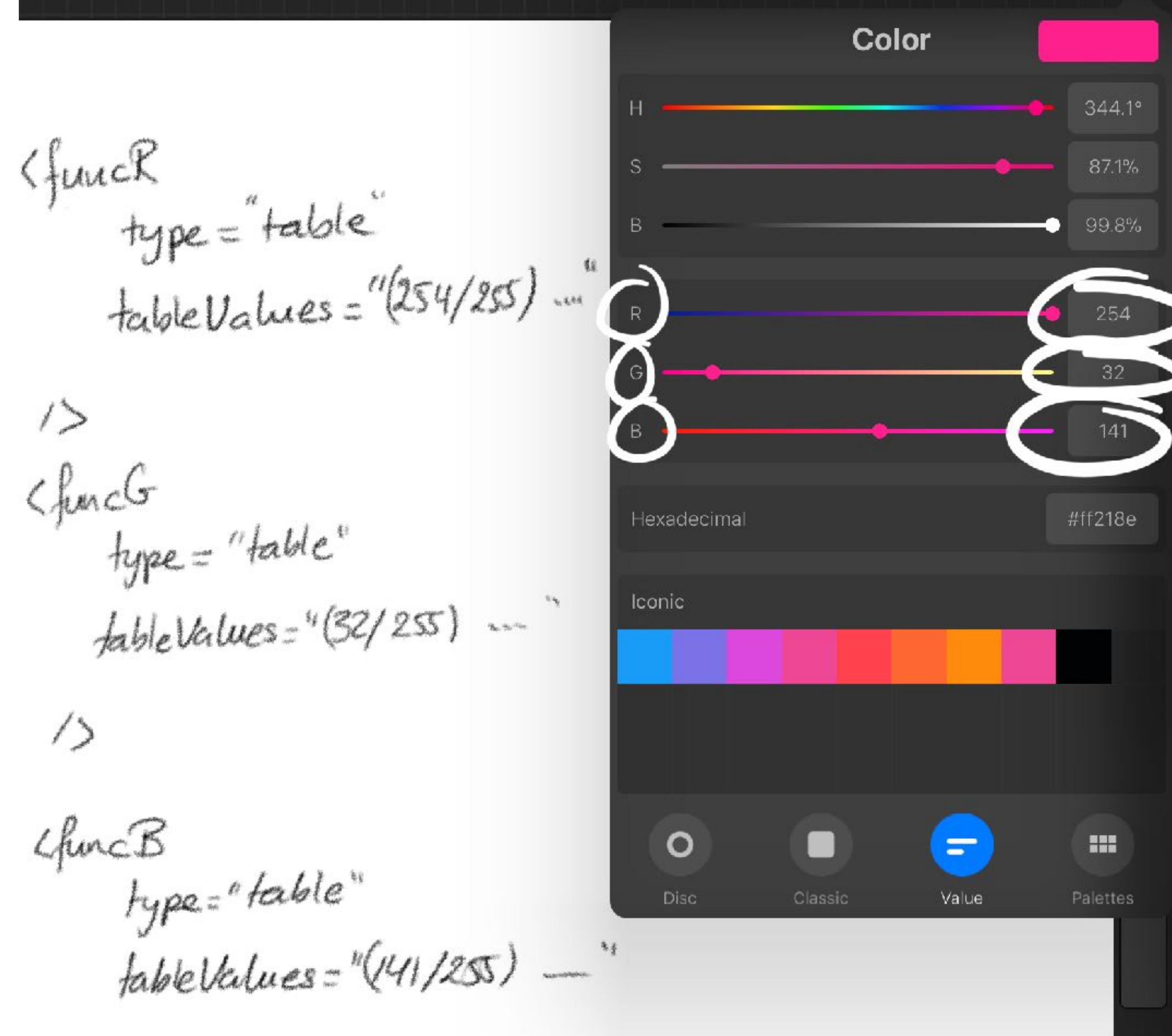
$G(Y)" > </feFuncG>$

<feFuncB tableValues = "B(X)"

$B(Y)" > </feFuncB>$

feFuncR: .996078431
feFuncG: .125490196
feFuncB: .552941176

```
<funcR  
    type = "table"  
    tableValues = "(254/255) ..."  
/>  
  
<funcG  
    type = "table"  
    tableValues = "(32/255) ..."  
/>  
  
<funcB  
    type = "table"  
    tableValues = "(141/255) ..."
```



```
<filter id="duotone">

    <feColorMatrix type="matrix"
        values=".33 .33 .33 0 0
                  .33 .33 .33 0 0
                  .33 .33 .33 0 0
                  0     0     0     1 0">
    </feColorMatrix>

    <feComponentTransfer color-interpolation-filters="sRGB">
        <feFuncR type="table" tableValues=".996078431   .984313725"></feFuncR>
        <feFuncG type="table" tableValues=".125490196   .941176471"></feFuncG>
        <feFuncB type="table" tableValues=".552941176   .478431373"></feFuncB>
    </feComponentTransfer>

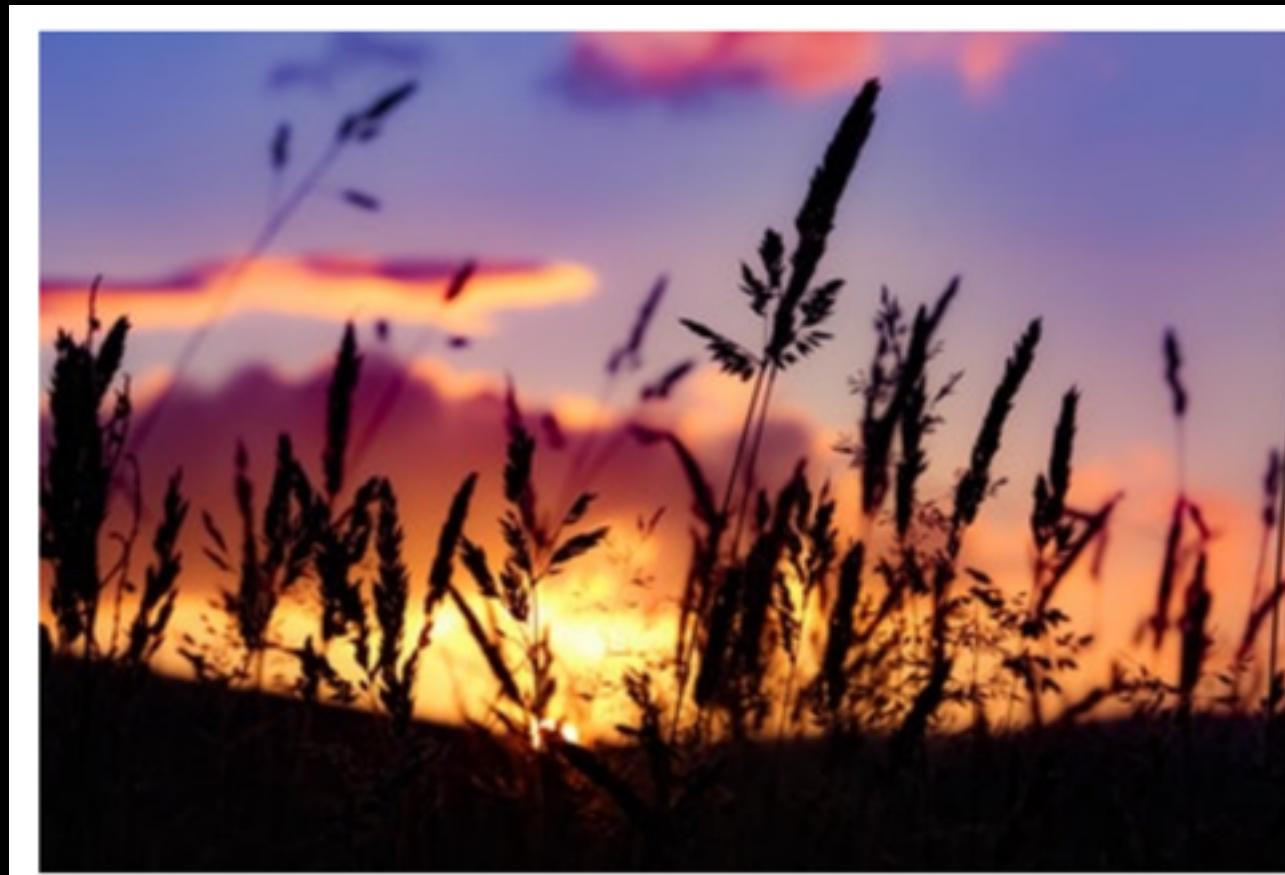
</filter>
```



```
<feComponentTransfer color-interpolation-filters="sRGB">  
  <feFuncR  
    type="gamma" exponent="1.25" amplitude="1.29" offset="0">  
  </feFuncR>  
  <feFuncG  
    type="gamma" exponent="1.25" amplitude="1.29" offset="0">  
  </feFuncG>  
  <feFuncB  
    type="gamma" exponent="1.25" amplitude="1.29" offset="0">  
  </feFuncB>  
</feComponentTransfer>
```



color-interpolation-filters



SVG Filters: A Crash Course - X

Secure | https://codepen.io/collection/0b42ddcbfd4072bbe500bab9e139563/

JAN 9, 2018

Delete Collection

Edit Collection

Sara Soueidan

TV

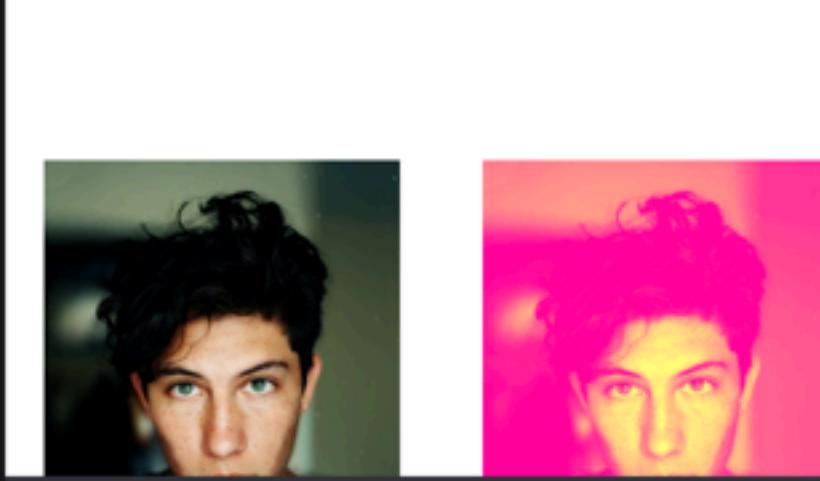
SVG Filters: A Crash Course

Pens related to and part of SVG filters explorations for the talk "SVG Filters: A Crash Course" and for general SVG Filters examples.



Image Posterization 2
Sara Soueidan PRO

0 0 0



Duotone Image effect
Sara Soueidan PRO

0 0 0

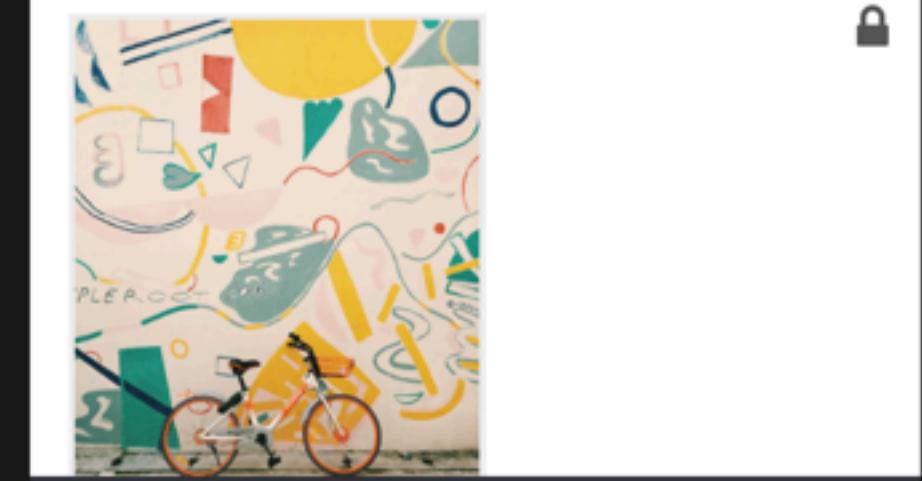


Image Posterization
Sara Soueidan PRO

0 0 0



Displace



Displace



NEW

THANK YOU

@SaraSoueidan
sarasoueidan.com