

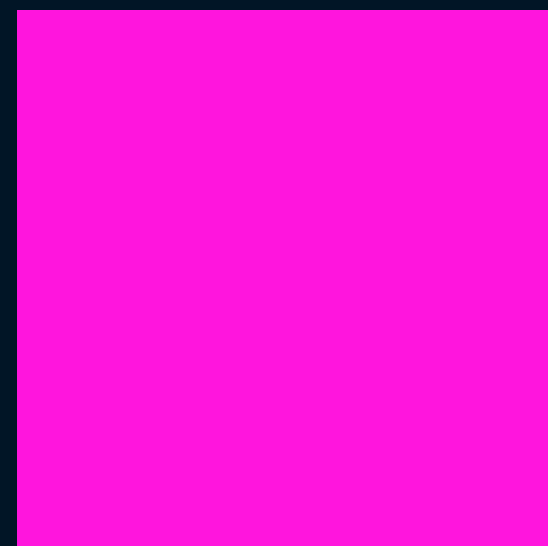
Custom Properties

The Secret Ingredients for CSS Magic

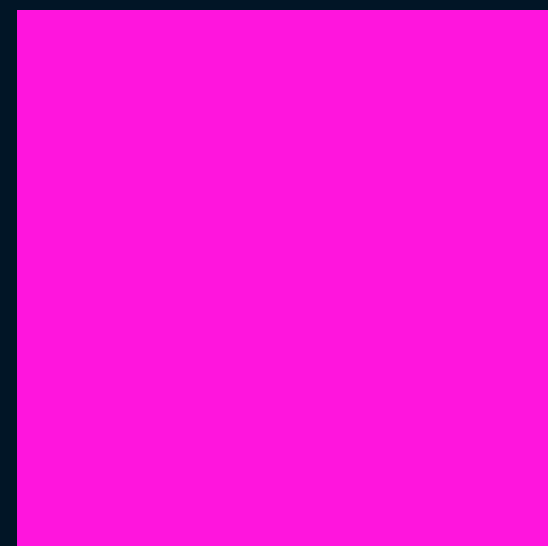


```
:root {  
  --primaryColor: deeppink;  
}
```

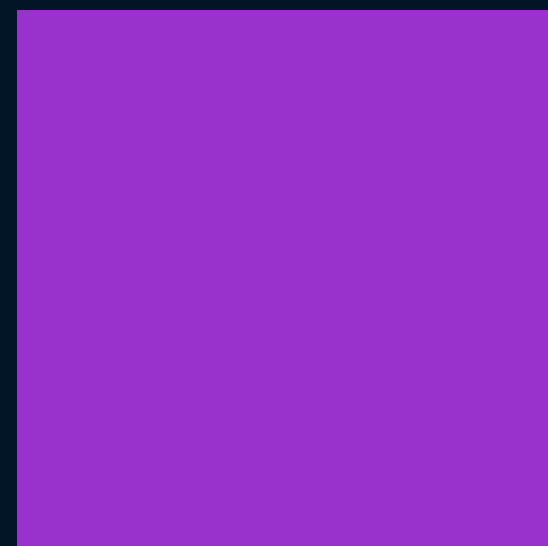
```
:root {  
  --primaryColor: deeppink;  
}  
  
.box {  
  background-color: var(--primaryColor);  
}
```




```
:root {  
  --primaryColor: deeppink;  
}  
  
.box {  
  background-color: var(--primaryColor, darkorchid);  
}
```

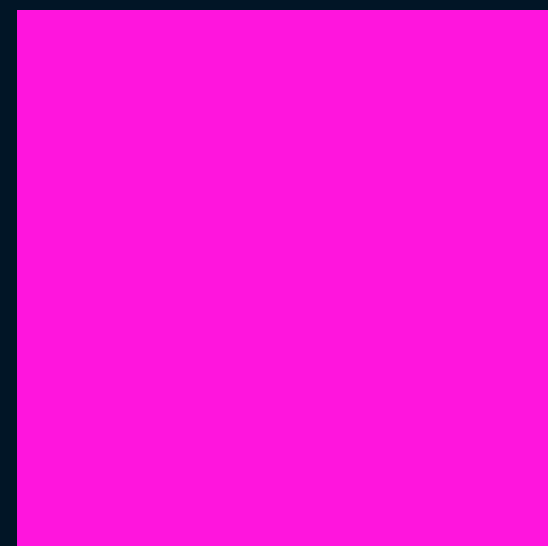


```
/* :root {  
  --primaryColor: deeppink;  
} */  
  
.box {  
  background-color: var(--primaryColor, darkorchid);  
}
```



IE11 🥲

```
.box {  
  background-color: deeppink;  
  background-color: var(--bg, darkorchid);  
}
```



IE11 🥲

```
.box {  
  /* background-color: deeppink; */  
  background-color: var(--bg, darkorchid);  
}
```

The square is here,
you just can't see it



Nice lovely browser

```
.box {  
  background-color: deeppink;  
  background-color: var(--bg, 300);  
}
```

Sorry, it's still
invisible



Sass variable



```
$bgColor: deeppink;
```

!=

```
--primaryColor: deeppink;
```



Custom property

Repeating values

Repeating values

Repeating values

Repeating values

Repeating values

Repeating values

Repeating values

```
.box {  
  --size: min(90vw, 40rem);  
  
  width: var(--size);  
  height: var(--size);  
}
```



```
.some-element {  
  padding: 12rem 1rem 1rem;  
}
```

```
.another-element {  
  padding: 0 1rem 1rem;  
}
```

```
.some-element {  
  padding: 12rem 1rem 1rem;  
}  
  
.another-element {  
  padding: 0 1rem 1rem;  
}  
  
@media (min-width: 50em) {  
  .some-element {  
    padding: 12rem 3rem 3rem;  
  }  
  
  .another-element {  
    padding: 0 3rem 3rem;  
  }  
}
```

```
.some-element {  
  padding: 12rem var(--pad) var(--pad);  
}
```

```
.another-element {  
  padding: 0 var(--pad) var(--pad);  
}
```



```
:root {  
  --pad: 1rem;  
}
```

```
@media (min-width: 50em) {  
  :root {  
    --pad: 3rem;  
  }  
}
```

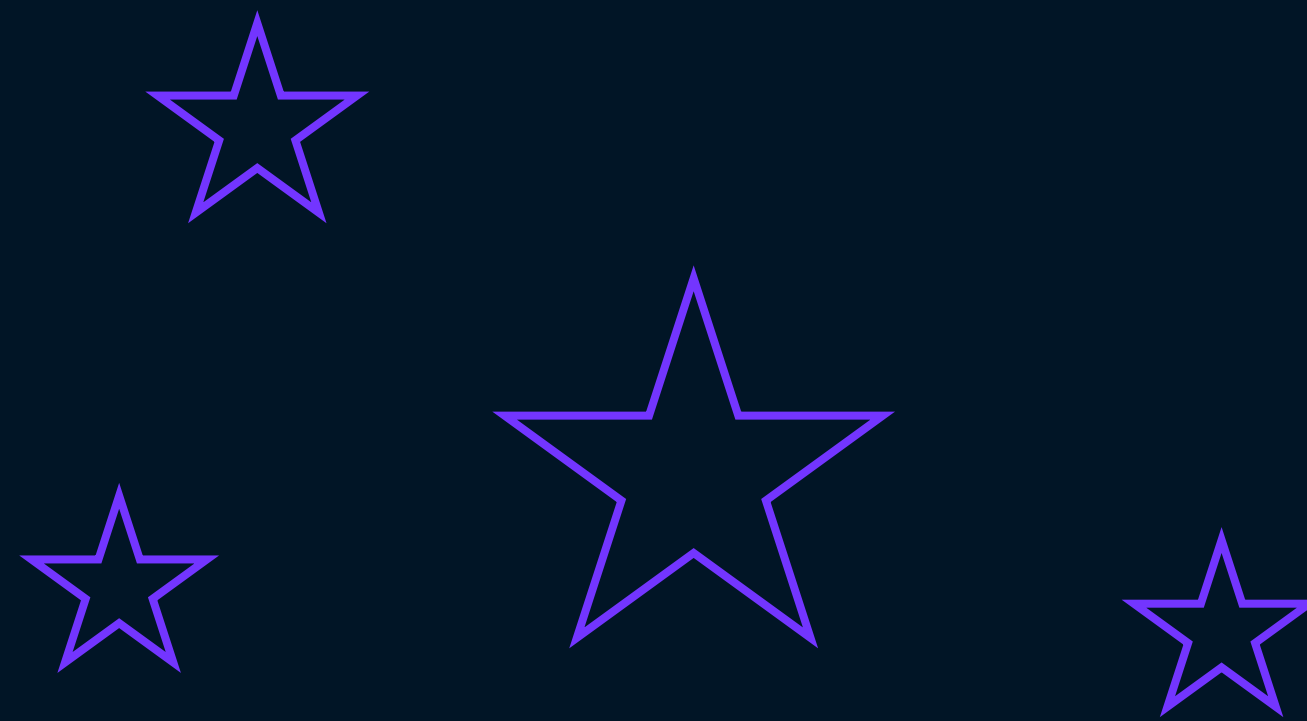
```
.element {  
  --color: 150, 0, 190;  
  
  background:  
    linear-gradient(  
      rgba(var(--color), 0.2),  
      rgba(var(--color), 1));  
}
```



```
:root {  
  --clip: polygon(80% 0, 100% 0, 100% 100%, 20% 100%);  
}  
  
.element {  
  -webkit-clip-path: var(--clip);  
  clip-path: var(--clip);  
}
```

A diagram illustrating the concept of global and local scoping. It features a large magenta circle representing the global scope. Inside this circle, towards the bottom right, is a smaller blue circle representing a local scope. The text "Global vs local scoping" is centered within the magenta circle.

Global vs local scoping



Dark mode

72 Posts

```
:root {  
  --primary: deepPink;  
  --headerBg: #1d1d26;  
  --textColor: #0e0f0f;  
  --textColorInverse: #fcfdff;  
  --bg: #fcfdff;  
  --bgTint: #dfeded;  
  --bgGrey: #e6e8e8;  
  --white: #fcfdff;  
}
```

Lastest article

CSS-only Slide-up Caption How

27 May, 2020

Have you ever needed to build a UI where a caption needed revealing more content? This is something I've...

All the CSS Colours

19 May, 2020

I recently made a demo where I needed to list all the CSS named colours as options in a select input. I didn't want to write them all out by...

Exciting Things For CSS Layout

01 May, 2020

This past week has been a busy one for me from browser vendor...


```
@media (prefers-color-scheme: dark) {  
  :root {  
    --bg: #161618;  
    --bgTint: #27272c;  
    --bgGrey: #27272c;  
    --textColor: #dbd7db;  
    --textColorInverse: #0e0f0f;  
  }  
}
```

72 Posts

Lastest article

CSS-only Slide-up Caption How

27 May, 2020

Have you ever needed to build a UI where a caption needed revealing more content? This is something I've...

All the CSS Colours

19 May, 2020

I recently made a demo where I needed to list all the CSS named colours as options in a select input. I didn't want to write them all out by...

Exciting T For CSS La

01 May, 2020

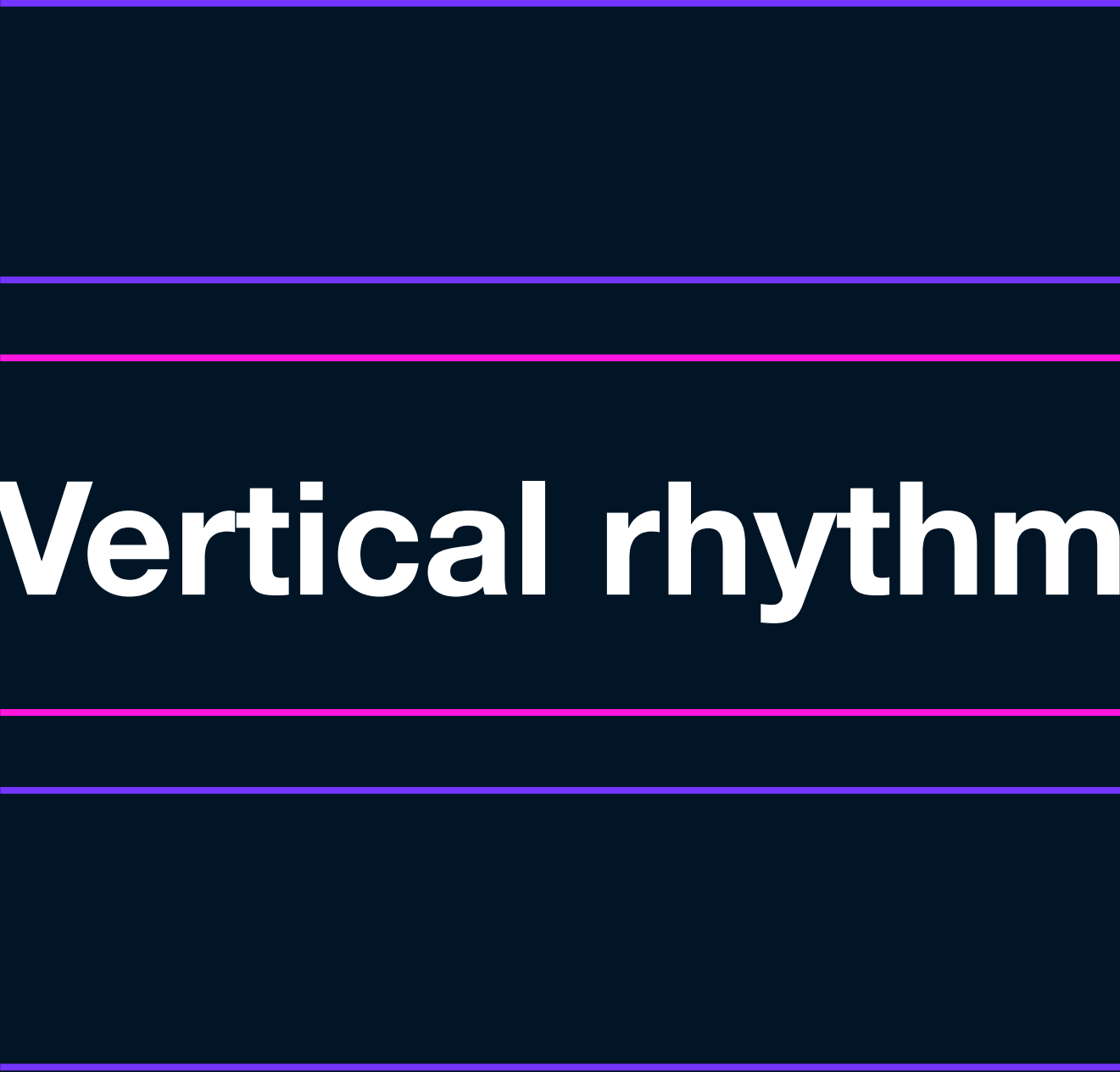
This past week h from browser v

Demo time!

Polyfills:

✗ postcss-custom-properties

✓ ie11CustomProperties

The image features a dark navy blue background. Centered horizontally are five thin, parallel horizontal lines. From top to bottom, the colors of these lines are: light purple, light purple, magenta, magenta, and light purple. The text 'Vertical rhythm' is centered between the second and fourth lines.

Vertical rhythm

--vr: 2rem

Vertical rhythm and custom properties

Using custom properties to set a vertical rhythm for our design system.

First heading

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga delectus harum distinctio iure id. Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

This section has a different vertical rhythm

Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

[Find out more →](#)

var(--vr)

Vertical rhythm and custom properties

Using custom properties to set a vertical rhythm for our design system.

First heading

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga delectus harum distinctio iure id. Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

This section has a different vertical rhythm

Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

[Find out more →](#)


```
calc(2 * var(--vr))
```

Vertical rhythm and custom properties

Using custom properties to set a vertical rhythm for our design system.

First heading

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga delectus harum distinctio iure id. Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

This section has a different vertical rhythm

Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

[Find out more →](#)

```
* + * {  
  margin-top: var(--vr);  
}
```

```
* + * {  
  margin-top: var(--vr);  
}
```

```
* + h2 {  
  margin-top: calc(2 * var(--vr));  
}
```



```
:root {  
  --vr: 1rem;  
}
```

```
@media (min-width: 50em) {  
  :root {  
    --vr: 2rem;  
  }  
}
```

--vr: 1rem;

Vertical rhythm and custom properties

Using custom properties to set a vertical rhythm for our design system.

First heading

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga delectus harum distinctio iure id. Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

This section has a different vertical rhythm

Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

[Find out more →](#)

Second heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris et posuere risus. Nam tortor ex, consequat

--vr: 2rem;

Vertical rhythm and custom properties

Using custom properties to set a vertical rhythm for our design system.

First heading

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Fuga delectus harum distinctio iure id. Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

This section has a different vertical rhythm

Minus nesciunt accusantium tenetur, explicabo, temporibus labore, eligendi commodi sequi molestiae perspiciatis asperiores nihil unde itaque?

[Find out more →](#)

```
:root {  
  --vr: 1rem;  
  --2vr: calc(2 * var(--vr));  
}
```

```
@media (min-width: 50em) {  
  :root {  
    --vr: 2rem;  
  }  
}
```


Color

Hue Saturation Lightness

```
hsl(3deg, 89%, 61%)
```

Hue Saturation Lightness

```
hsl(var(--hue), 89%, 61%)
```



```
.color-pair {  
  --hue: 3deg;  
  --hue2: calc(var(--hue) + 180deg);  
}
```

```
--hue: 3deg;
```



```
--hue2: calc(var(--hue) + 180deg);
```

```
.color-pair--red {  
  --hue: 3deg;  
}
```

`hsl(var(--hue), 89%, 61%);`

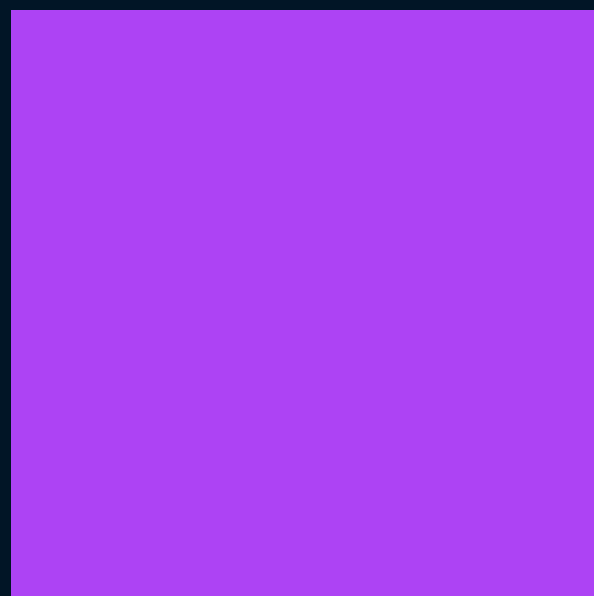


`hsl(var(--hue2), 89%, 61%);`




```
.color-pair--purple {  
  --hue: 276deg;  
}
```

`hsl(var(--hue), 89%, 61%);`

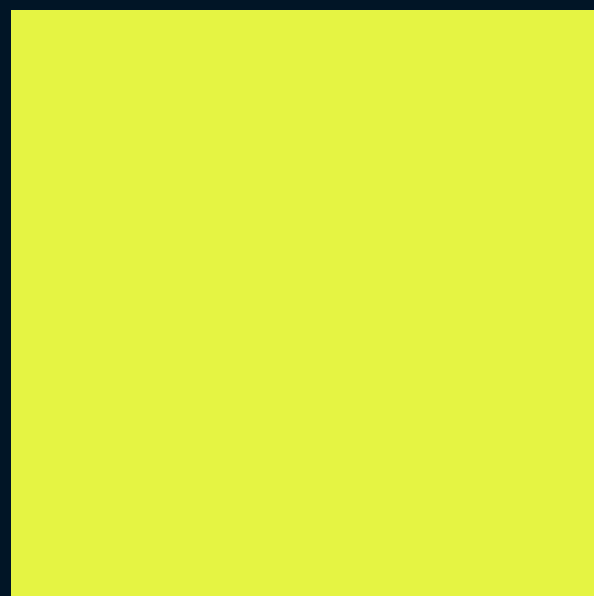


`hsl(var(--hue2), 89%, 61%);`



```
.color-pair--yellow {  
  --hue: 65deg;  
}
```

`hsl(var(--hue), 89%, 61%);`



`hsl(var(--hue2), 89%, 61%);`



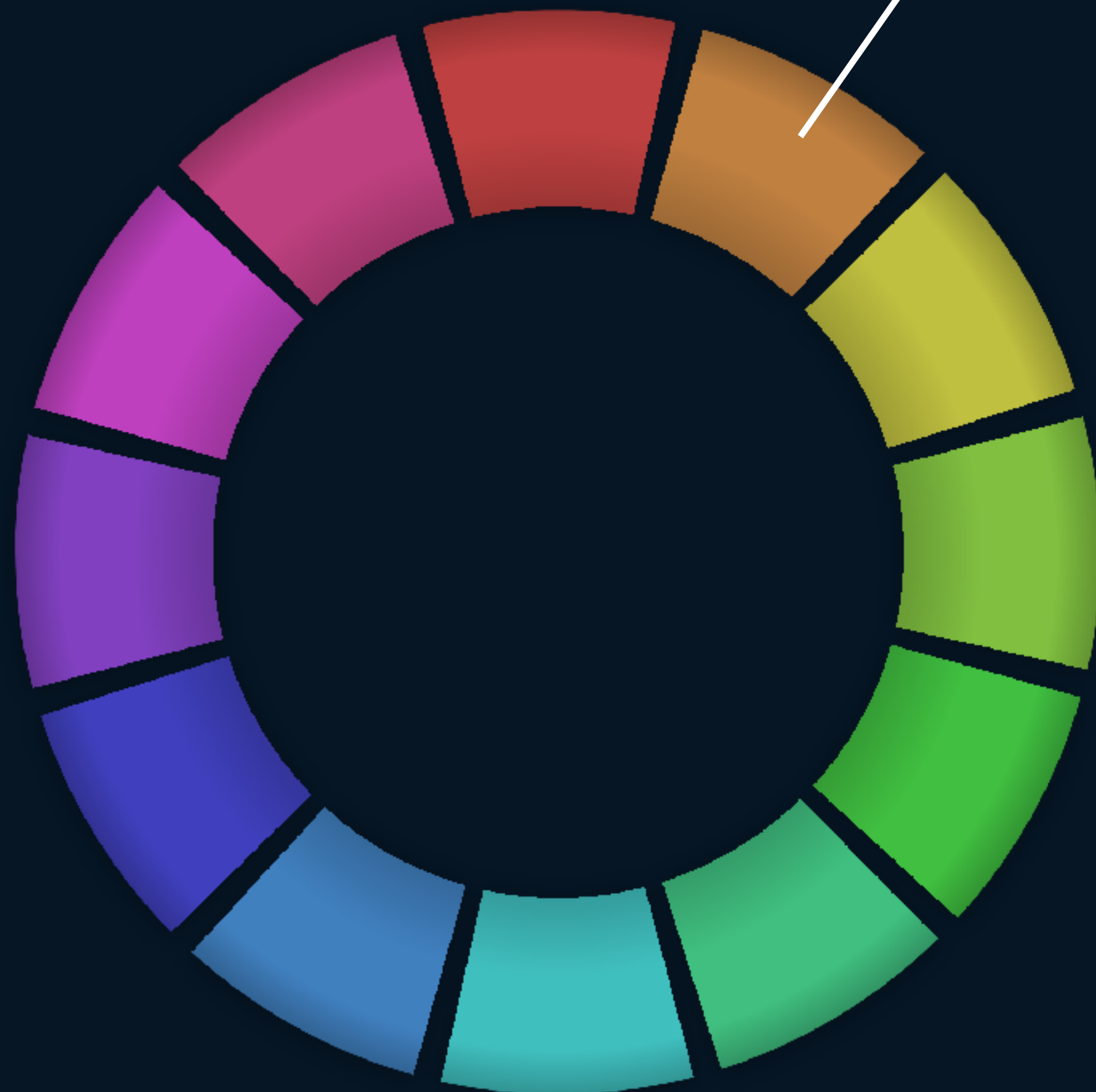


```
.element {  
  --segment: 30deg;  
}
```

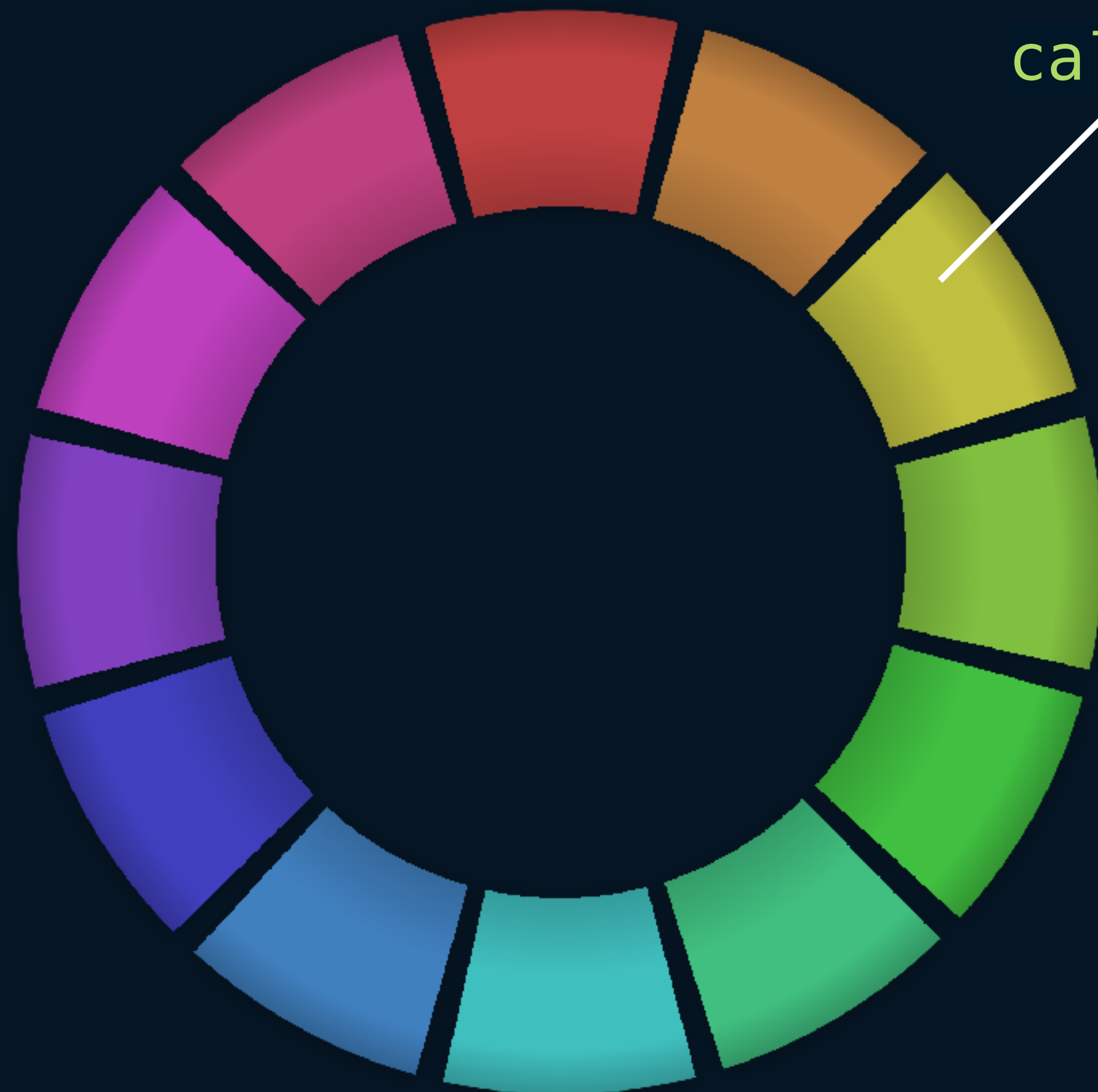


```
.element {  
  --segment: 30deg;  
}
```

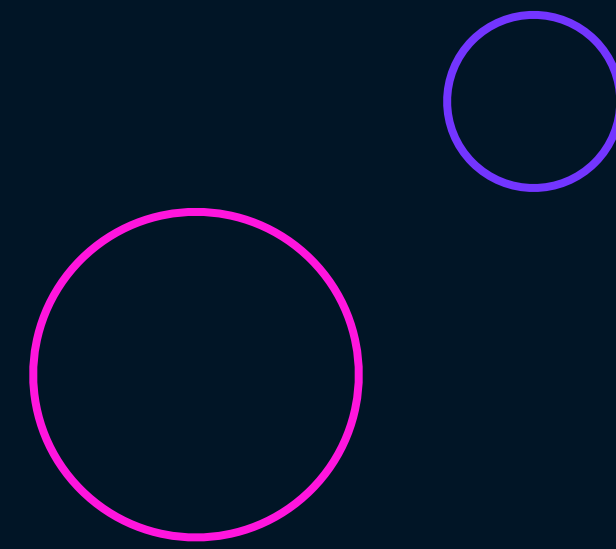
```
calc(var(--segment) * 2);
```




```
.element {  
  --segment: 30deg;  
}
```



`calc(var(--segment) * 3);`



Relative values

HIS DARK MATERIALS

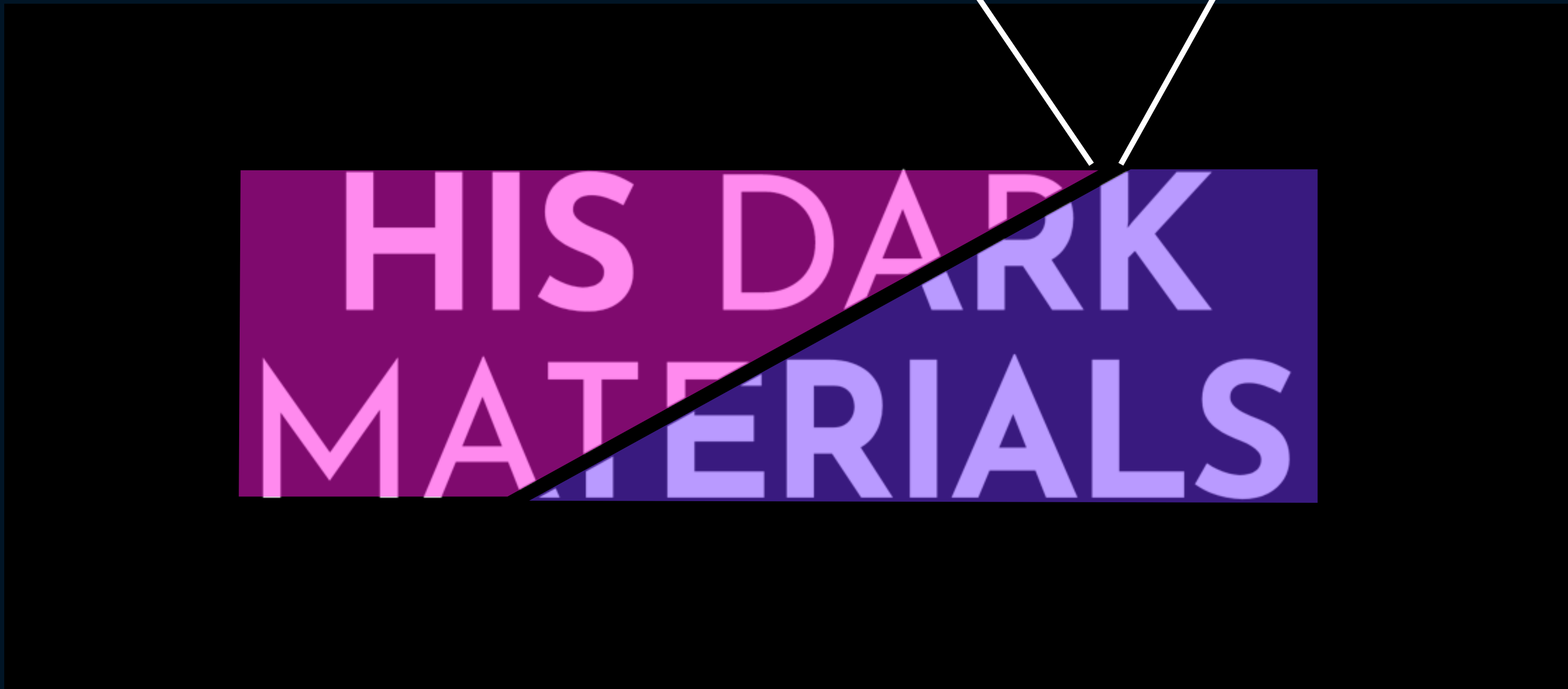




HIS DARK
MATERIALS

`calc(80% - 0.5rem)`

`calc(80% + 0.5rem)`



HIS DARK
MATERIALS

The diagram illustrates a CSS width calculation. A central rectangular box features a horizontal gradient from magenta on the left to purple on the right. Inside the box, the text "HIS DARK MATERIALS" is displayed in a bold, white, sans-serif font, arranged in two lines. Two white arrows originate from the top corners of the box and point upwards towards the CSS formulas. The left arrow points to the formula `calc(80% - 0.5rem)`, and the right arrow points to the formula `calc(80% + 0.5rem)`. These formulas are positioned above the box, with the minus sign in the first formula and the plus sign in the second. The entire scene is set against a dark blue background.

HIS DARK
MATERIALS

`calc(20% - 0.5rem)`

`calc(20% + 0.5rem)`

```
.element {  
  --bottom: 20%;  
  --top: 80%;  
  --offset: 0.5rem;  
}
```

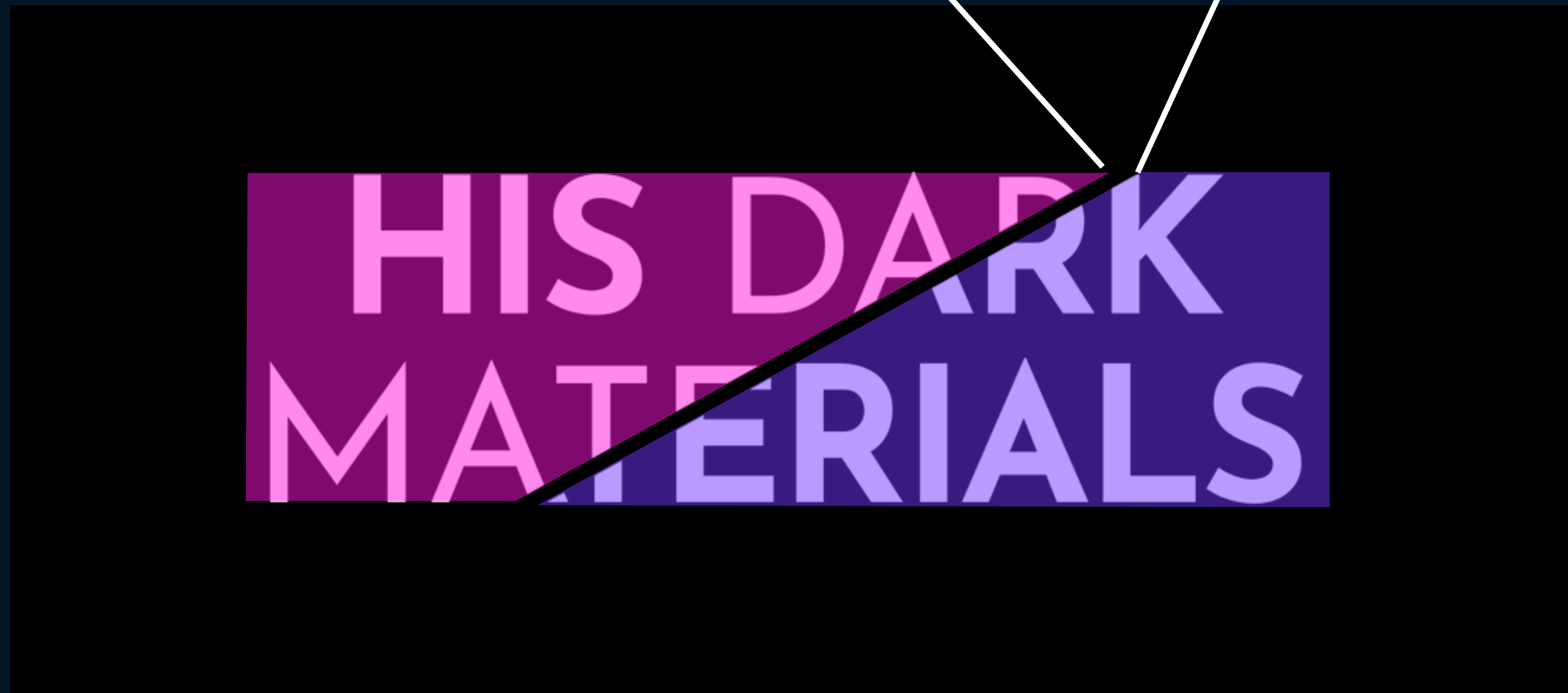


HIS DARK
MATERIALS


```
.element {  
  --bottom: 20%;  
  --top: 80%;  
  --offset: 0.5rem;  
}
```

`calc(var(--top) - var(--offset))`

`calc(var(--top) + var(--offset))`



```
.element {  
  --bottom: 20%;  
  --top: 80%;  
  --offset: 0.5rem;  
}
```



`calc(var(--bottom) - var(--offset))`

`calc(var(--bottom) + var(--offset))`



Local scoping

The diagram consists of a large blue circle centered on the slide. Inside the lower-right quadrant of this blue circle is a smaller red circle. The text "Local scoping" is written in white, bold font in the center of the blue circle.


```
.grid {  
  --cols: 8;  
  
  grid-template-columns: repeat(var(--cols), minmax(0, 1fr));  
  ...  
}  
  
@media (min-width: 60em) {  
  .grid {  
    --cols: 12;  
  }  
}
```

```
@supports (---cols: 8) {  
  .grid {  
    ---cols: 8;  
  
    /* More code here... */  
  }  
}
```


Staggered

Animations

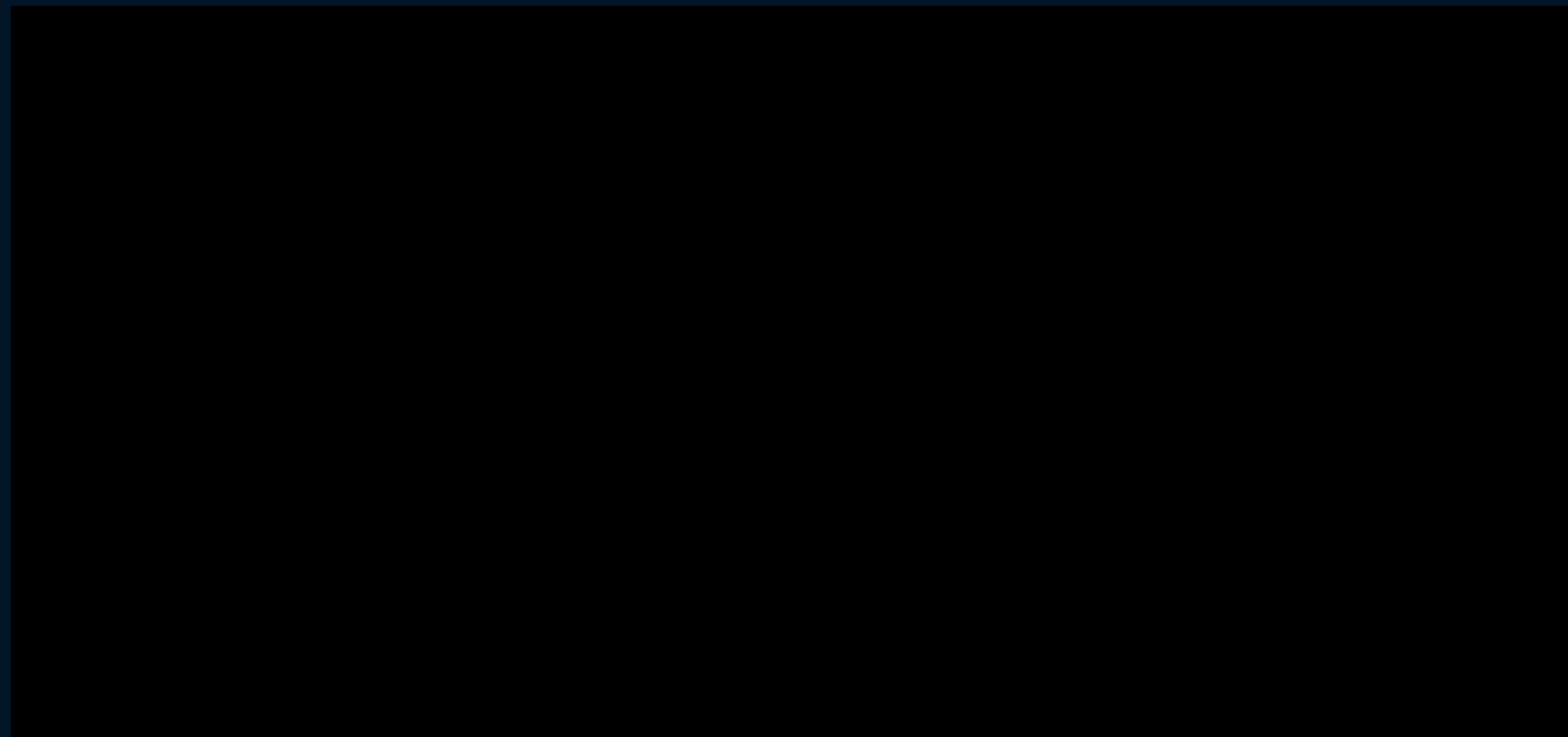

```
.some-element:nth-child(2) {  
  --i: 2;  
}
```

```
.some-element:nth-child(3) {  
  --i: 3;  
}
```



```
.some-element {  
  --delay: calc(var(--i, 1) * 400ms);  
}
```

```
.some-element {  
  --delay: calc(var(--i, 1) * 400ms);  
  animation: appear 1000ms var(--delay) forwards;  
}
```



Progress indicators

1 Discovery and assessment

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Adipiscing diam donec adipiscing tristique risus.

2 Information gathering and analysis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Adipiscing diam donec adipiscing tristique risus.

3 Creating your claim

```
<ol style="--length: 5">
  <li style="--i: 1">...</li>
  <li style="--i: 2">...</li>
  <li style="--i: 3">...</li>
  <li style="--i: 4">...</li>
  <li style="--i: 5">...</li>
</ol>
```



```
li::before {  
  --stop: calc(100% / var(--length) * var(--i));  
  
  background:  
    linear-gradient(to right,  
      var(--c1) var(--stop), var(--c2) var(--stop));  
}
```



Information gathering and analysis

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Adipiscing diam donec adipiscing tristique risus.

Demo time!

splitting.js.org

[Guide](#)[Plugins](#)[API](#)[Demos](#) [View on GitHub](#) 

CSS Vars for split words & chars! (items, grids, images, too!)

Splitting creates elements and adds CSS variables to unlock amazing possibilities for animating text, grids, and more!

[Get Splitting →](#)

Install with `npm i splitting -s` or [Download .zip](#)

DoubleDash your CSS

The previously impractical becomes easy with Splitting's CSS Variables.

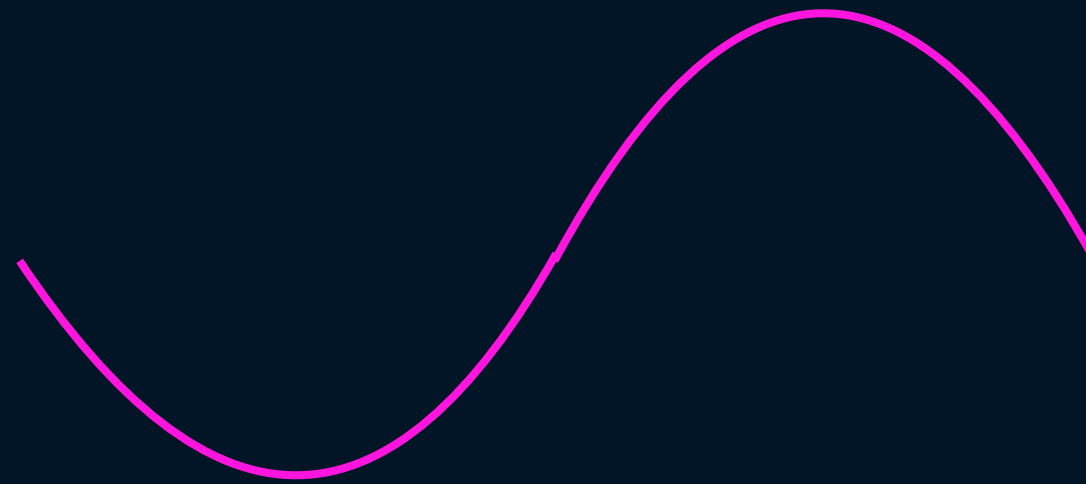
Comically Small

A shocking amount of features in a 1.5kb (minified & gzipped) package.

Progressively Enhance

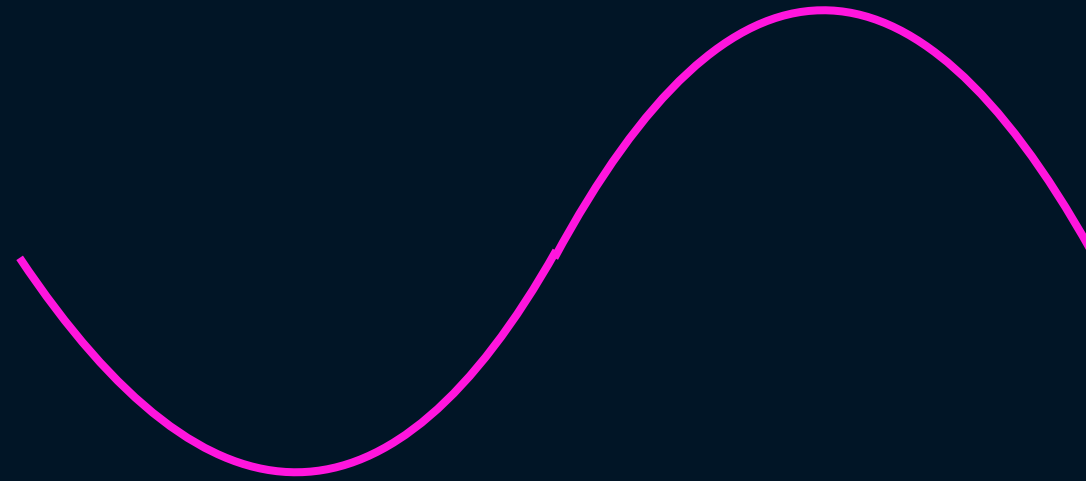
Upgrade websites for modern browsers without breaking IE.

Motion Path





Motion Path



```
.element {  
  offset-path: path('M.4 84.1s127.4 188 267.7 0  
                    247.3 0 247.3 0');  
  offset-distance: 0;  
}
```

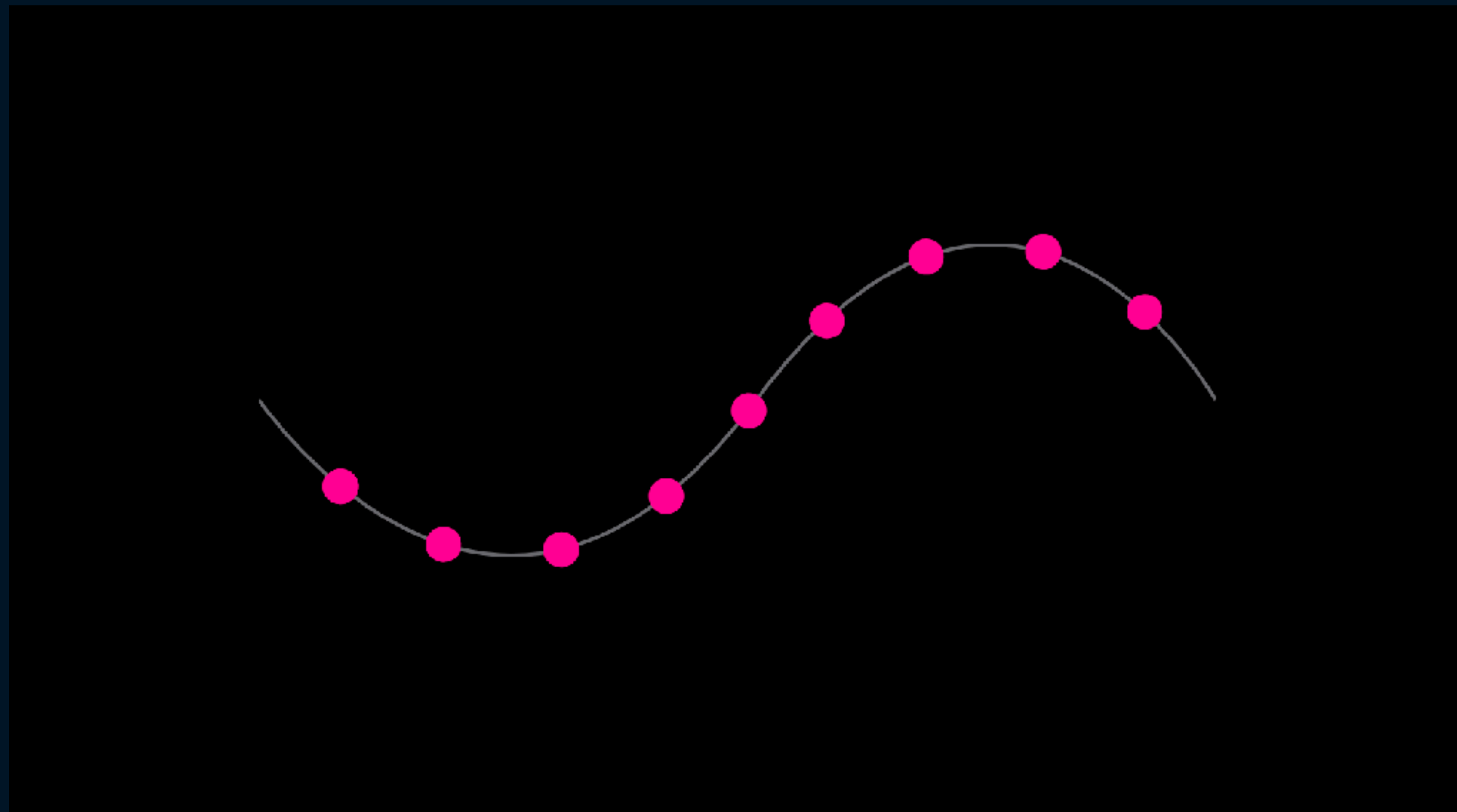


```
.element {  
  offset-path: path('M.4 84.1s127.4 188 267.7 0  
                    247.3 0 247.3 0');  
  offset-distance: 0;  
  animation: move 2000ms infinite alternate;  
}
```

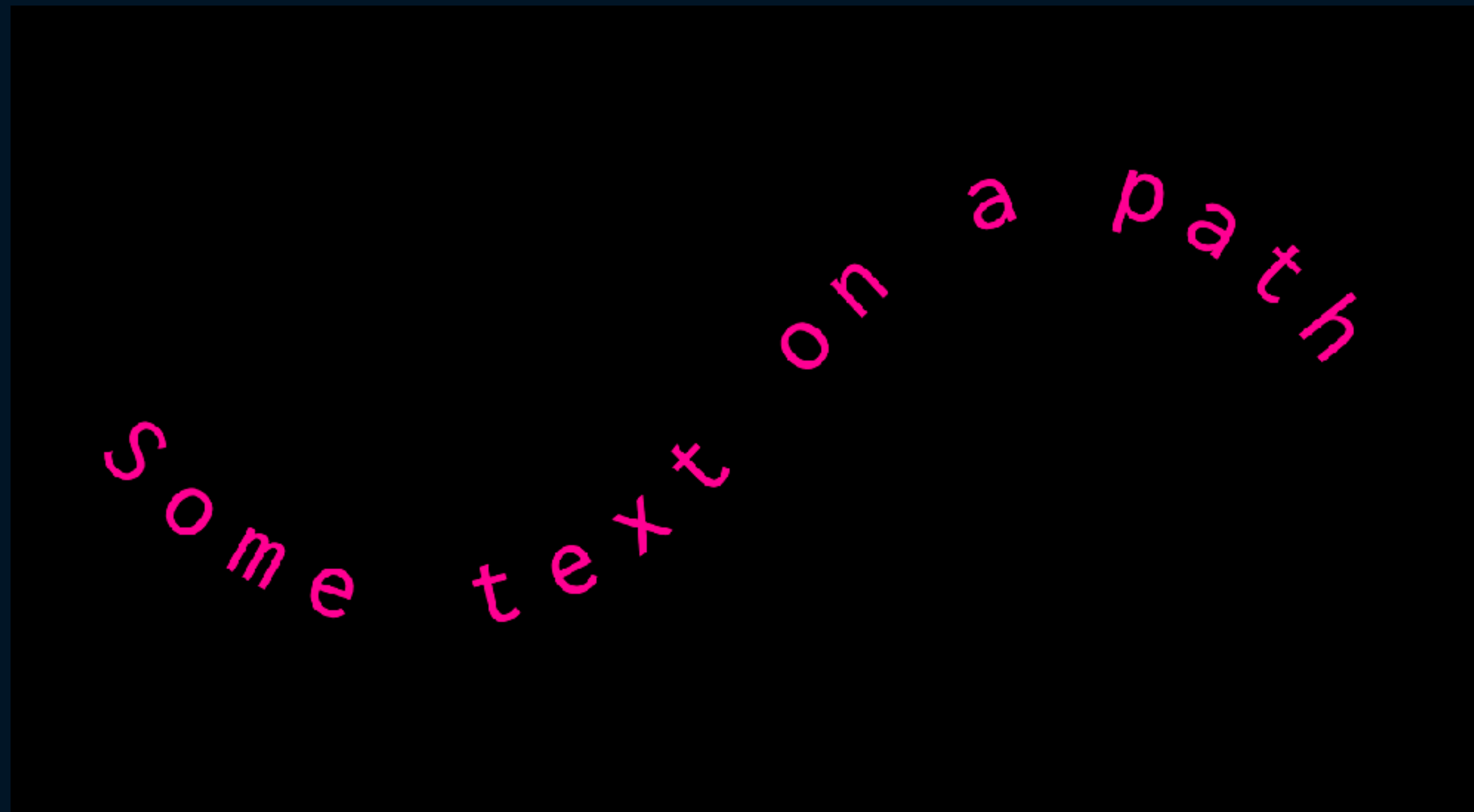
```
@keyframes move {  
  to {  
    offset-distance: 100%;  
  }  
}
```



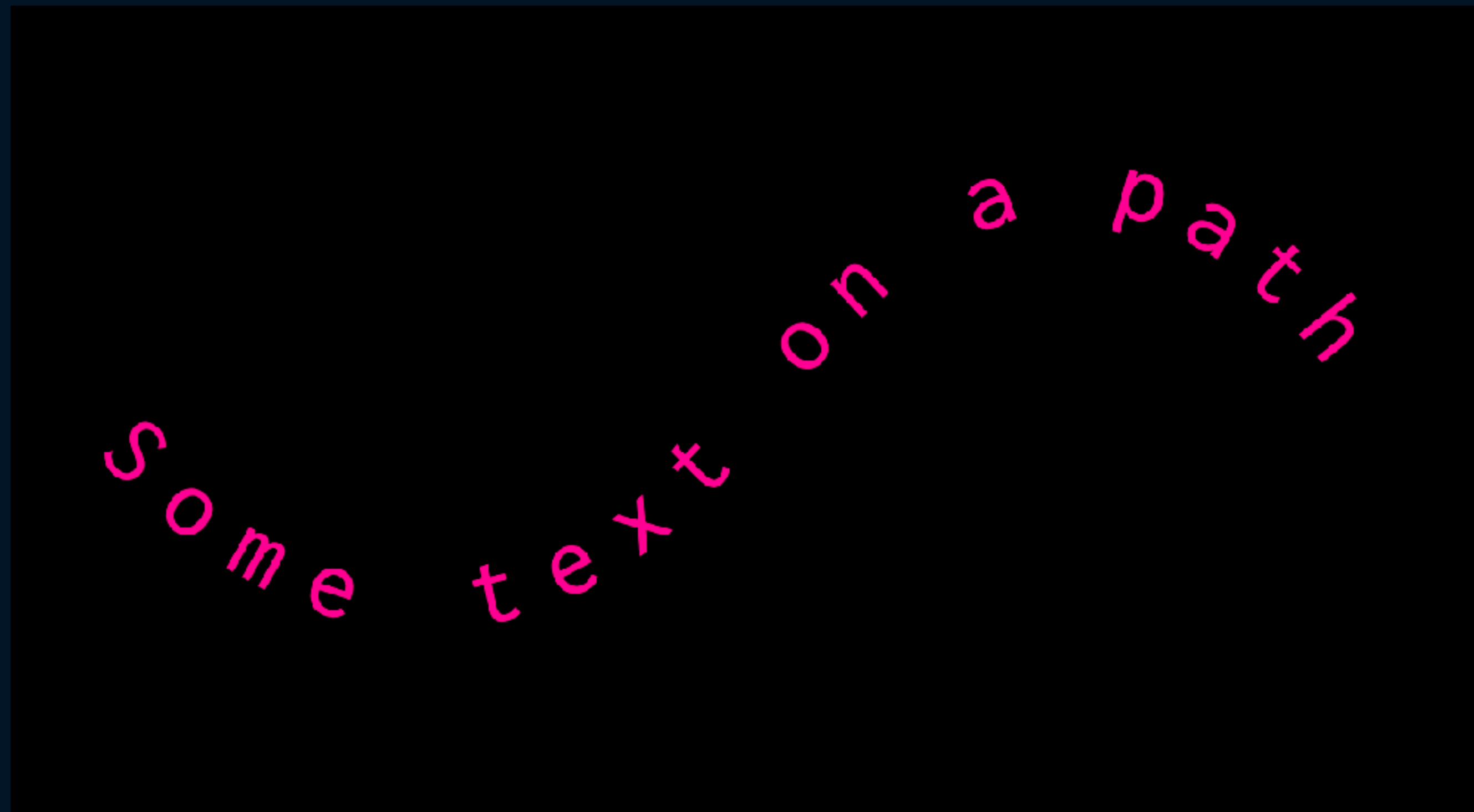
```
.element {  
  offset-distance: calc(var(--i, 1) * 4rem);  
}
```



```
.char {  
  offset-distance: calc(var(--i, 1) * 2rem);  
}
```



```
.char {  
  offset-distance: calc(var(--i, 1) * var(--offset));  
}
```



TEXT IN A CIRCLE
WITH MOTION PATH

```
--offset: calc(100% / (var(--char-total) + 1));
```

TEXT IN A CIRCLE
WITH MOTION PATH

```
offset-distance: calc(var(--offset) * var(--char-index));
```

TEXT IN A CIRCLE
WITH MOTION PATH


```
.char {  
  --position: calc(var(--char-index) * 1em);  
  
  offset-distance: var(--position);  
  animation: move 1500ms infinite alternate forwards;  
}  
  
@keyframes move {  
  to {  
    offset-distance: calc(var(--position) + 12rem);  
  }  
}
```

I AM ANIMATING


```
--delay: calc(var(--char-index) * 30ms);
```

I AM ANIMATING



CAN'T STOP LOOPING

JUST HANGING OUT


```
.item {
  --d: 1000ms;
  --i: calc(var(--item-total) - var(--item-index));
  --x: calc((var(--item-index) - var(--i)) * 4deg);
  --tPerSegment: calc(var(--d) / var(--item-total));
  --itemDuration: calc(var(--tPerSegment) * var(--item-index));
  --itemDuration2: calc(var(--tPerSegment) * var(--i));
  --c: calc(1 / var(--item-total) * 50);
  --h: calc(var(--item-index) * 15);
  --color: hsl(var(--h), 50%, 50%);
  --offset: calc(var(--item-index) * (var(--c) * 1%) + 1.65rem);
  --offset2: calc(var(--i) * (var(--c) * 1%));
  --endPos: calc(var(--item-total) * (var(--c) * 1%));
}
```


Zig-zag gradient lab

Options

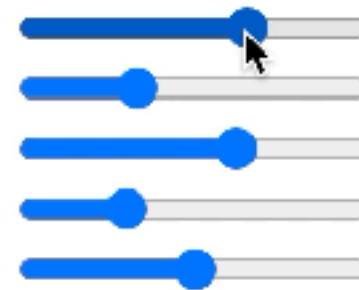
Angle

Stripe thickness

Segment width

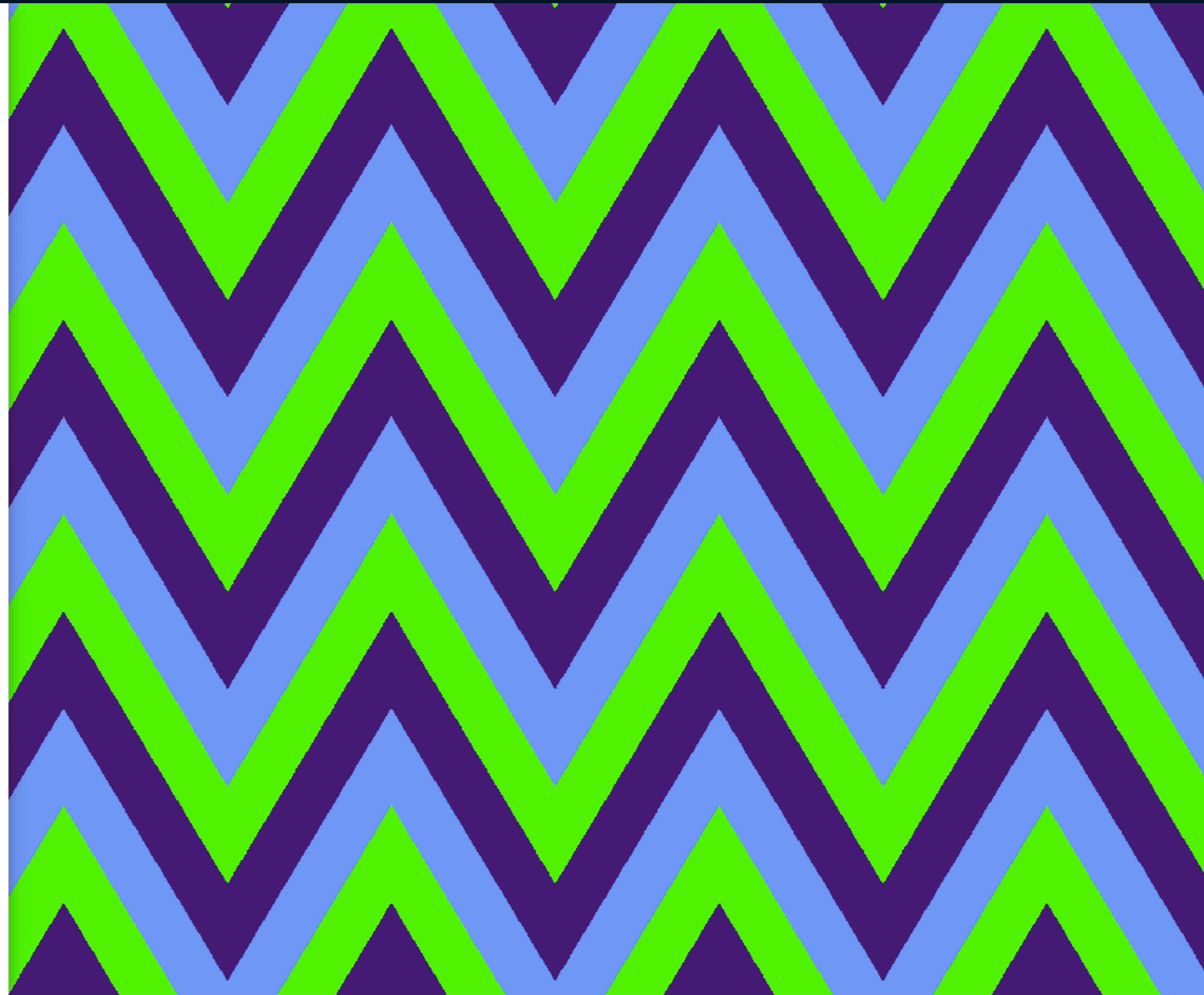
Primary colour

Contrast



CSS output

```
.bg {  
  background-image: repeating-linear-gradient(59deg,  
    background-size: 131px 100%, 131px 100%;  
}  
  
.bg::after {  
  background-image: repeating-linear-gradient(calc(  
    background-size: 131px 100%, 131px 100%;  
    -webkit-mask-image: repeating-linear-gradient(to  
    mask-image: repeating-linear-gradient(to right,  
  })
```



✨ Houdini ✨

```
CSS.registerProperty({  
  name: '--angle',  
  syntax: '<angle>',  
  inherits: true,  
  initialValue: '40deg',  
})
```



```
transition: --angle 200ms;
```

Zig-zag gradient lab

Options

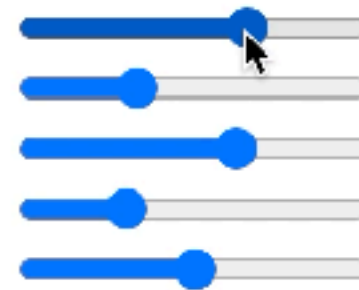
Angle

Stripe thickness

Segment width

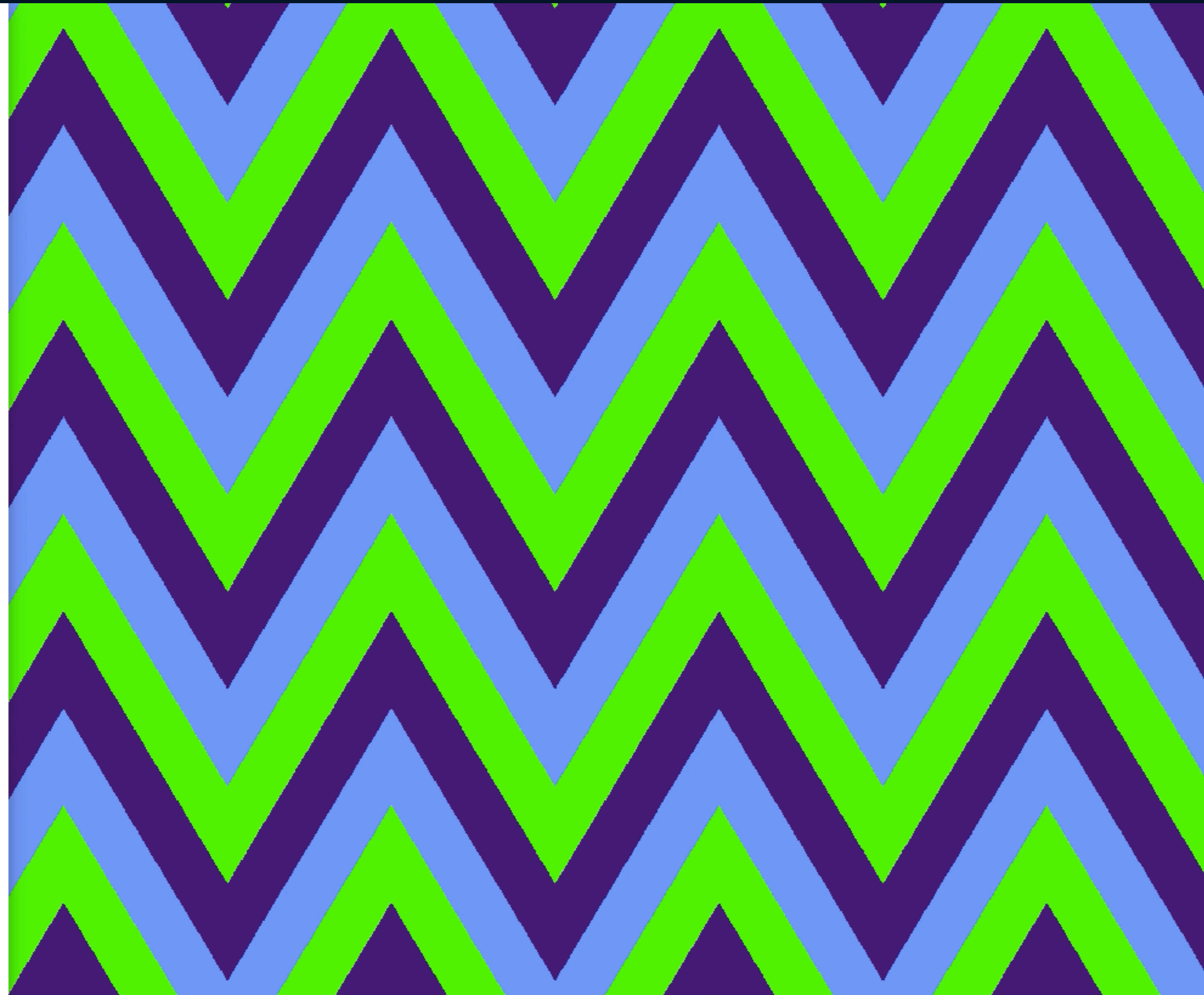
Primary colour

Contrast



CSS output

```
.bg {  
  background-image: repeating-linear-gradient(59deg,  
    background-size: 131px 100%, 131px 100%;  
}  
  
.bg::after {  
  background-image: repeating-linear-gradient(calc(  
    background-size: 131px 100%, 131px 100%;  
    -webkit-mask-image: repeating-linear-gradient(to  
    mask-image: repeating-linear-gradient(to right,  
  })
```



```
@property --angle {  
  syntax: '<angle>';  
  inherits: true;  
  initial-value: 40deg;  
}
```


Demo time!


```
chars.forEach((el, i) => {  
  el.style.setProperty('--s', getRandom[i])  
})
```


A user's guide to CSS variables
by Lea Verou

bit.ly/3fSZCOS

Calculating Color: Dynamic Color Theming with Pure CSS
by Una Kravets

bit.ly/3akbTKP

CSS { In Real Life }

css-irl.info

@MicheBarks

@CSSInRealLife