



IF.03.01 _BASIC WEB TECHNIQUES

Creating Web Sites with HTML, CSS and Javascript

UNIT 06

Css 3

GOALS FOR THIS UNIT

- You know how to make gradients, corners, and shadows
- You know how to make transitions, animations, and transformations in CSS 3
- You know how to do absolute positioning

GRADIENTS, CORNERS, AND SHADOWS

BACKGROUNDS – LINEAR GRADIENT

BACKGROUNDS – LINEAR GRADIENT



BACKGROUNDS – LINEAR GRADIENT



```
#gradientbox {  
  background-image:
```


BACKGROUNDS – LINEAR GRADIENT



```
#gradientbox {  
  background-image:  
    linear-gradient(to right, red, orange,  
      yellow, green, blue, indigo, violet);  
}
```


BACKGROUNDS – LINEAR GRADIENT



```
#gradientbox {  
  background-image:  
    linear-gradient(to right, red, orange,  
      yellow, green, blue, indigo, violet);  
}
```

Direction

BACKGROUNDS – LINEAR GRADIENT

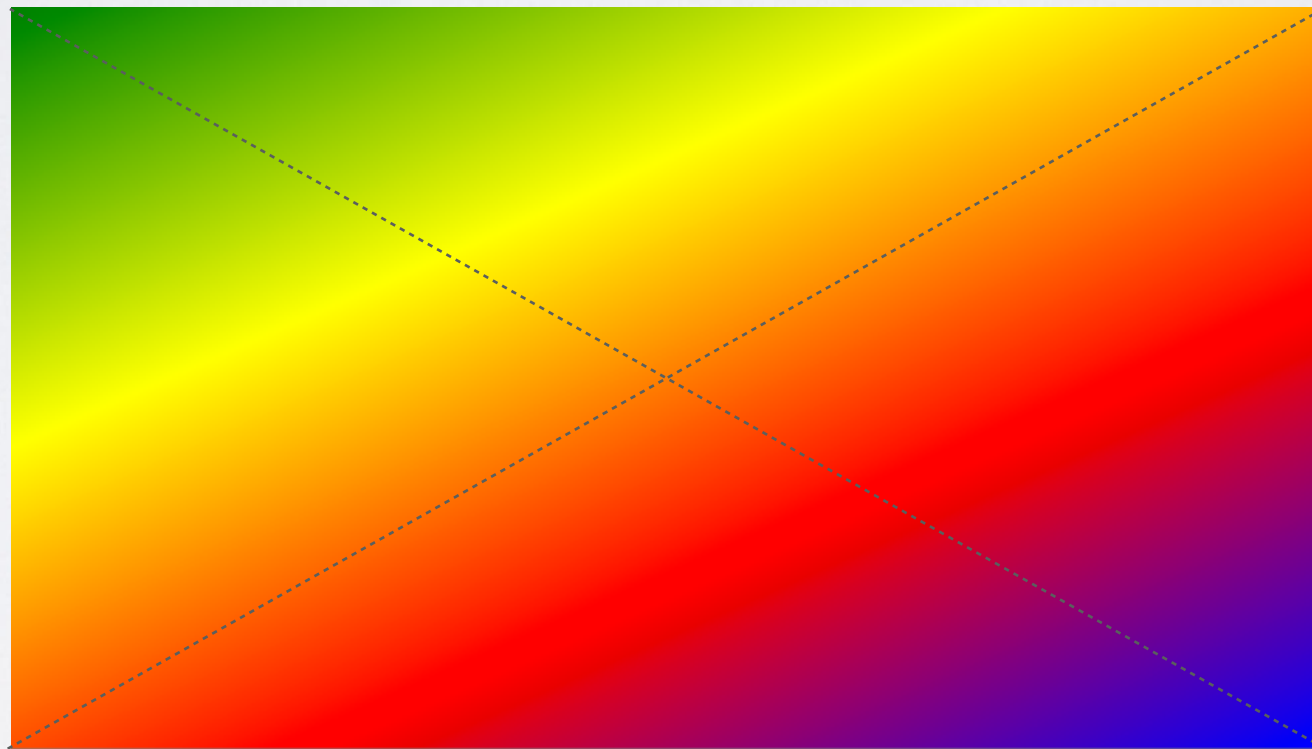


```
#gradientbox {  
  background-image:  
    linear-gradient(to right, red, orange,  
      yellow, green, blue, indigo, violet);  
}
```

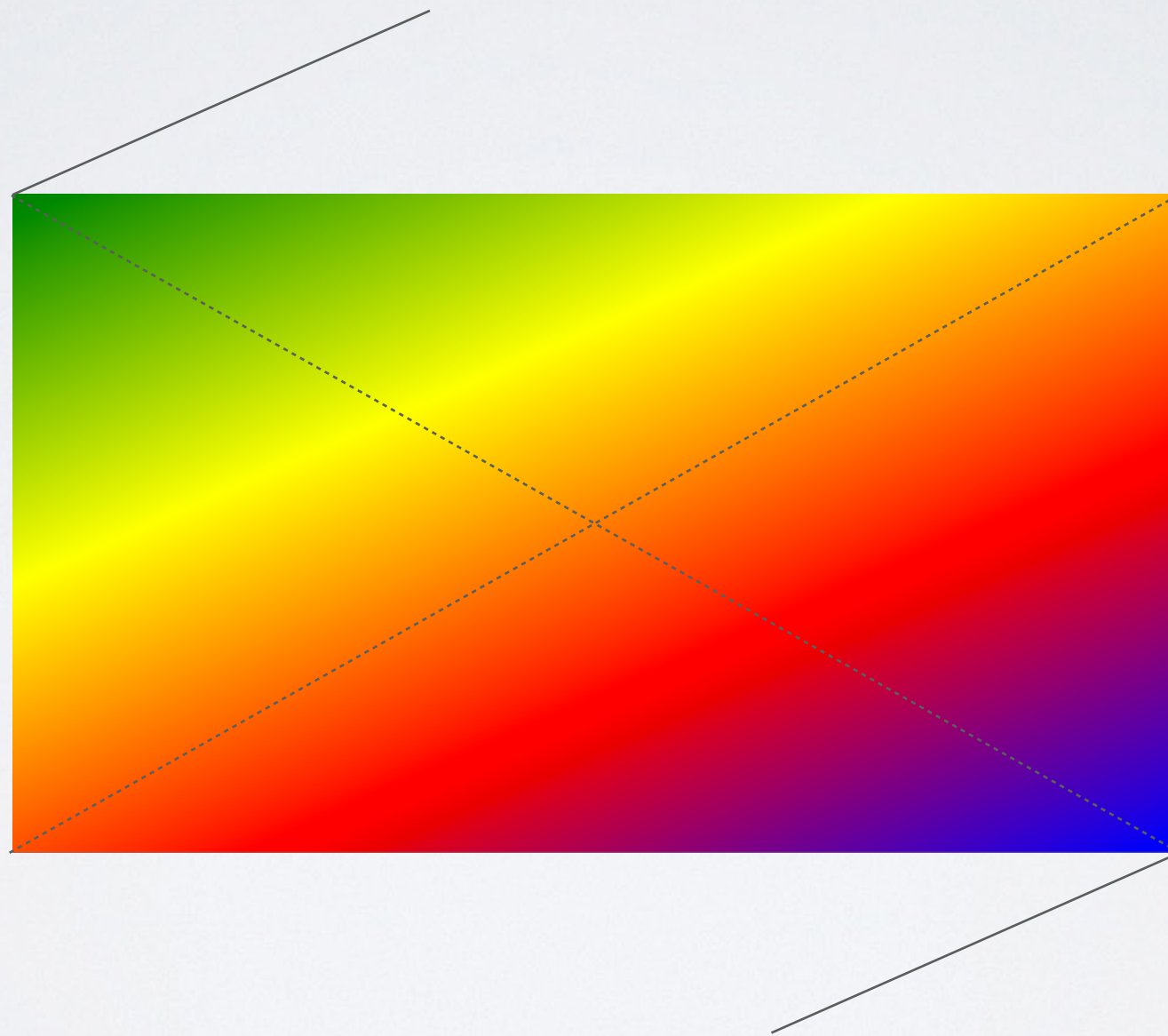
Direction

Color
Stops

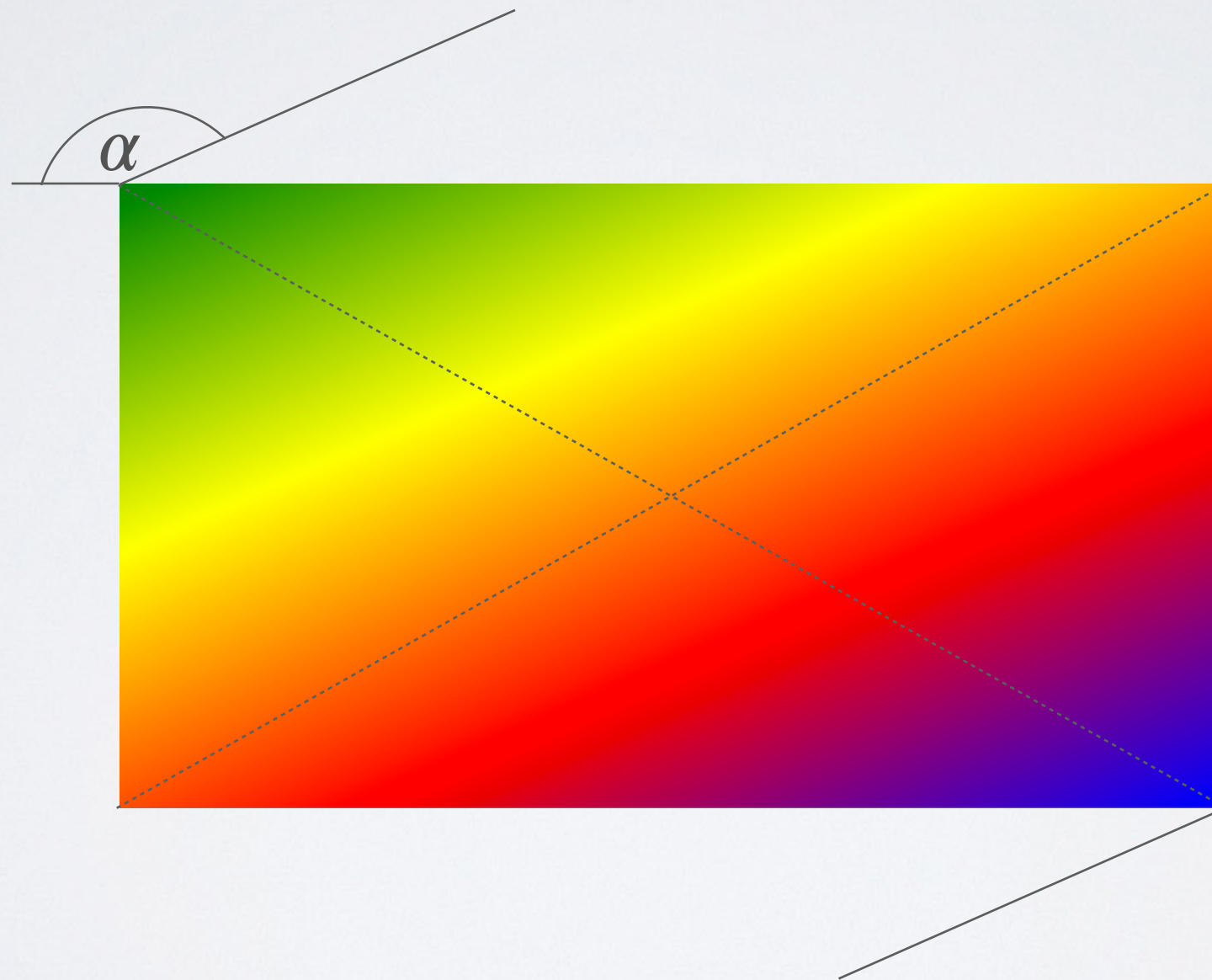
UNDERSTANDING LINEAR GRADIENTS



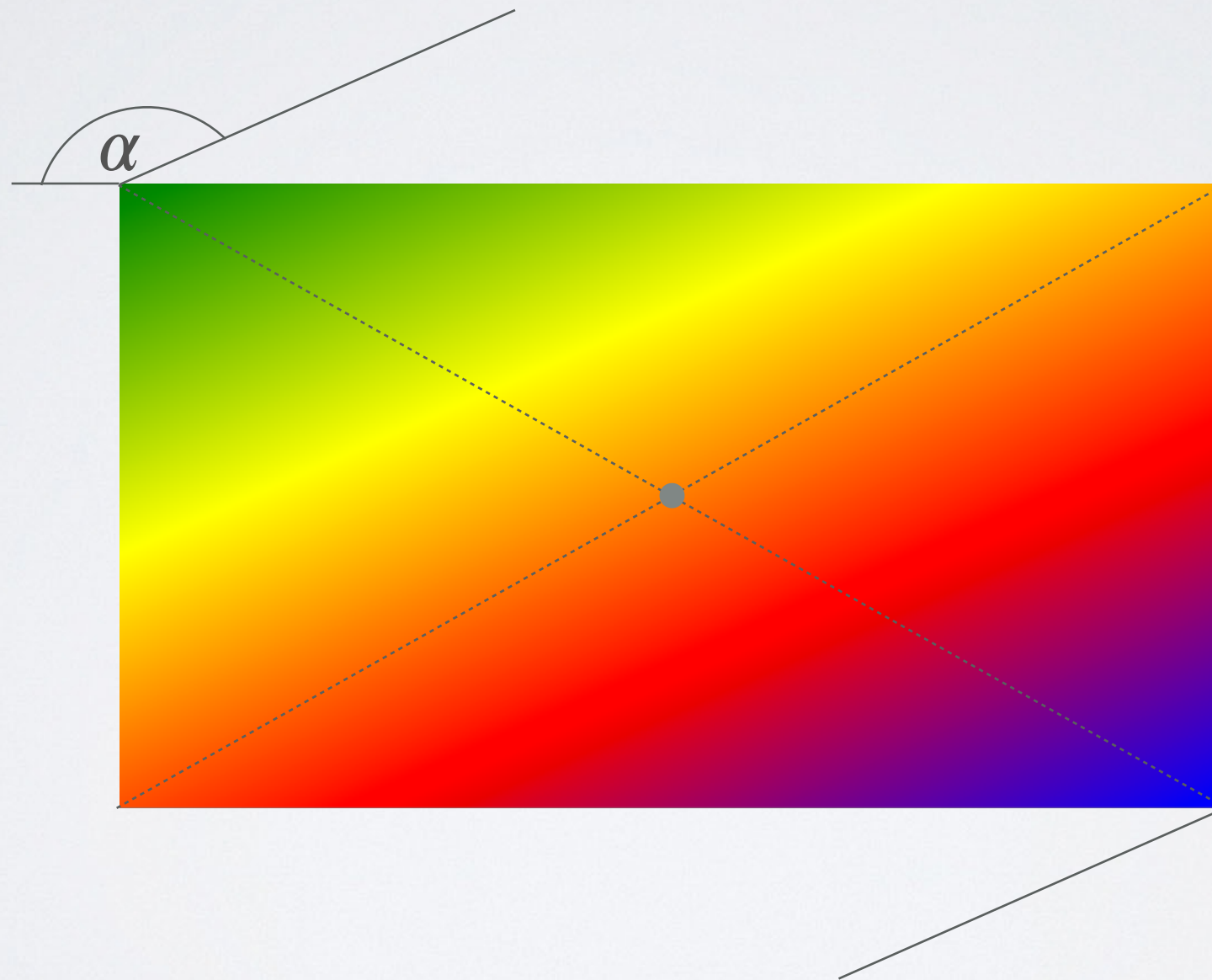
UNDERSTANDING LINEAR GRADIENTS



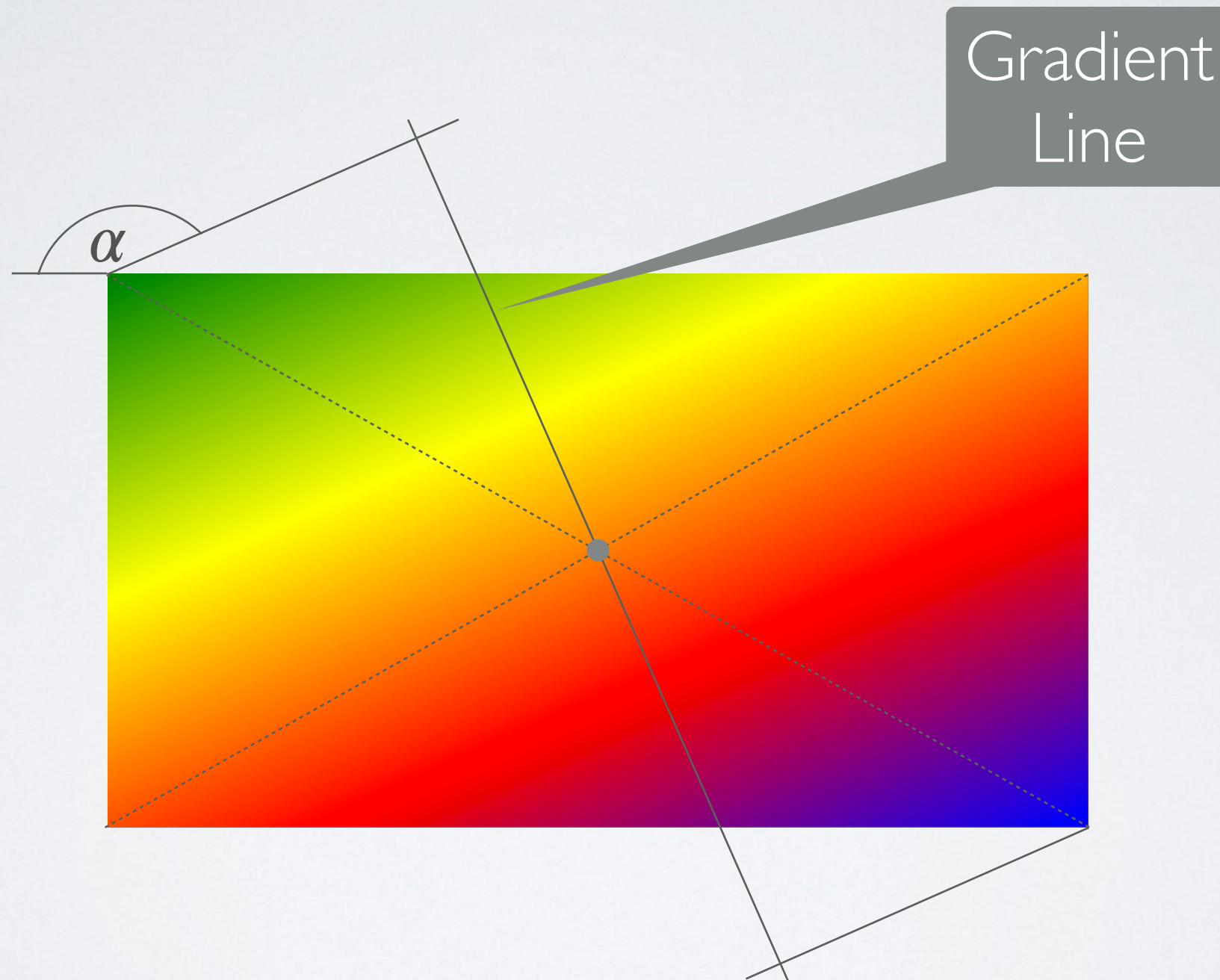
UNDERSTANDING LINEAR GRADIENTS



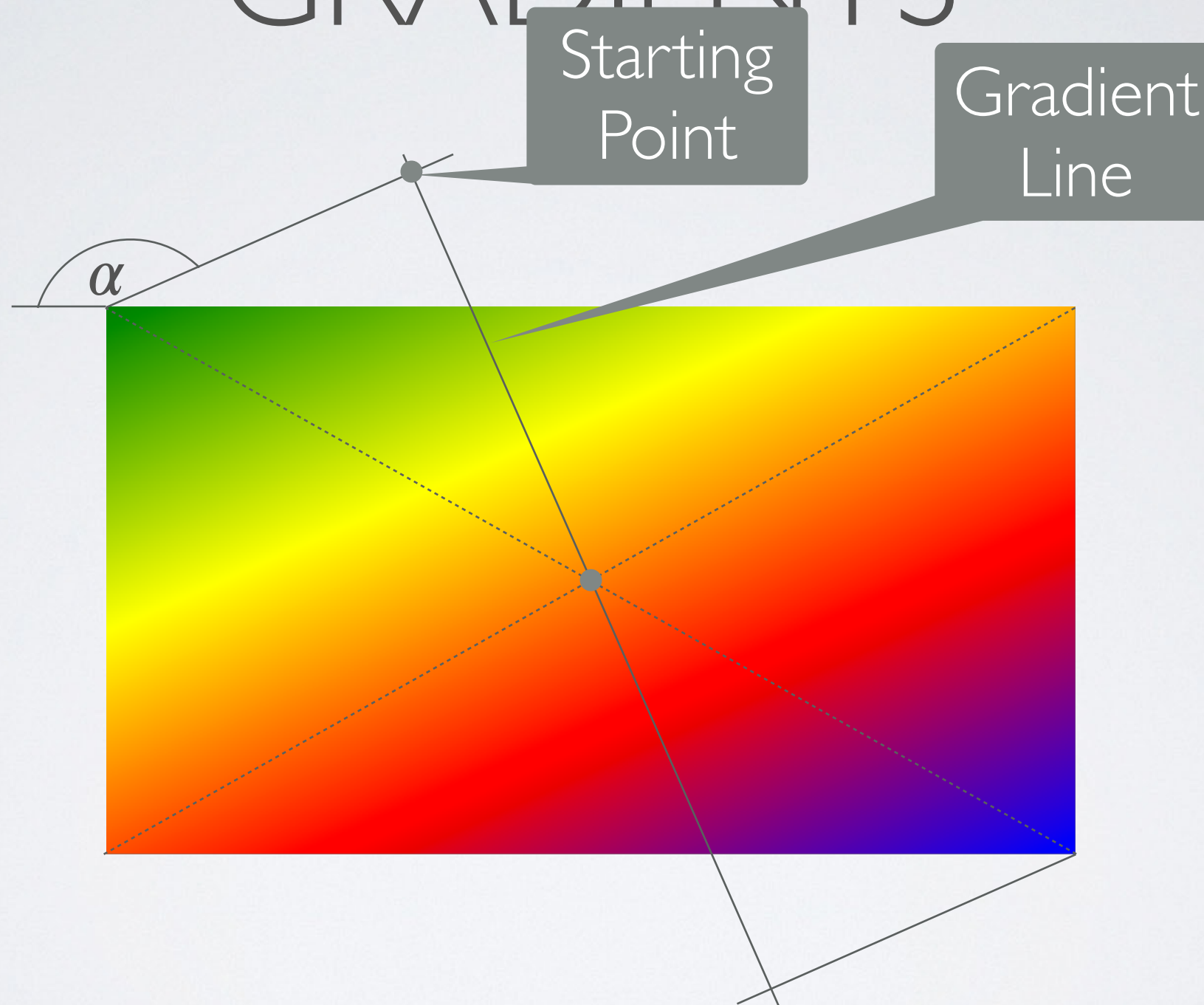
UNDERSTANDING LINEAR GRADIENTS



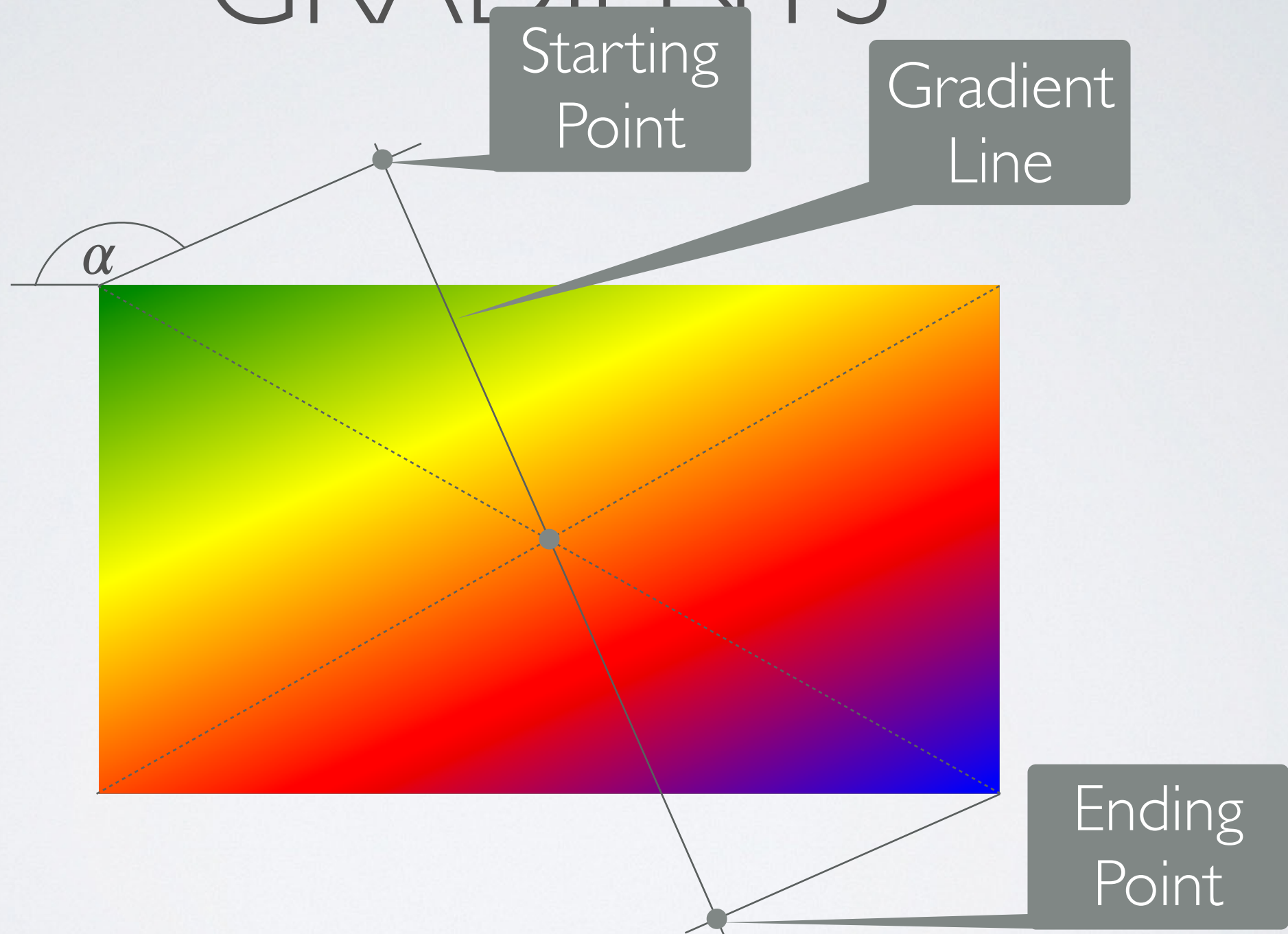
UNDERSTANDING LINEAR GRADIENTS



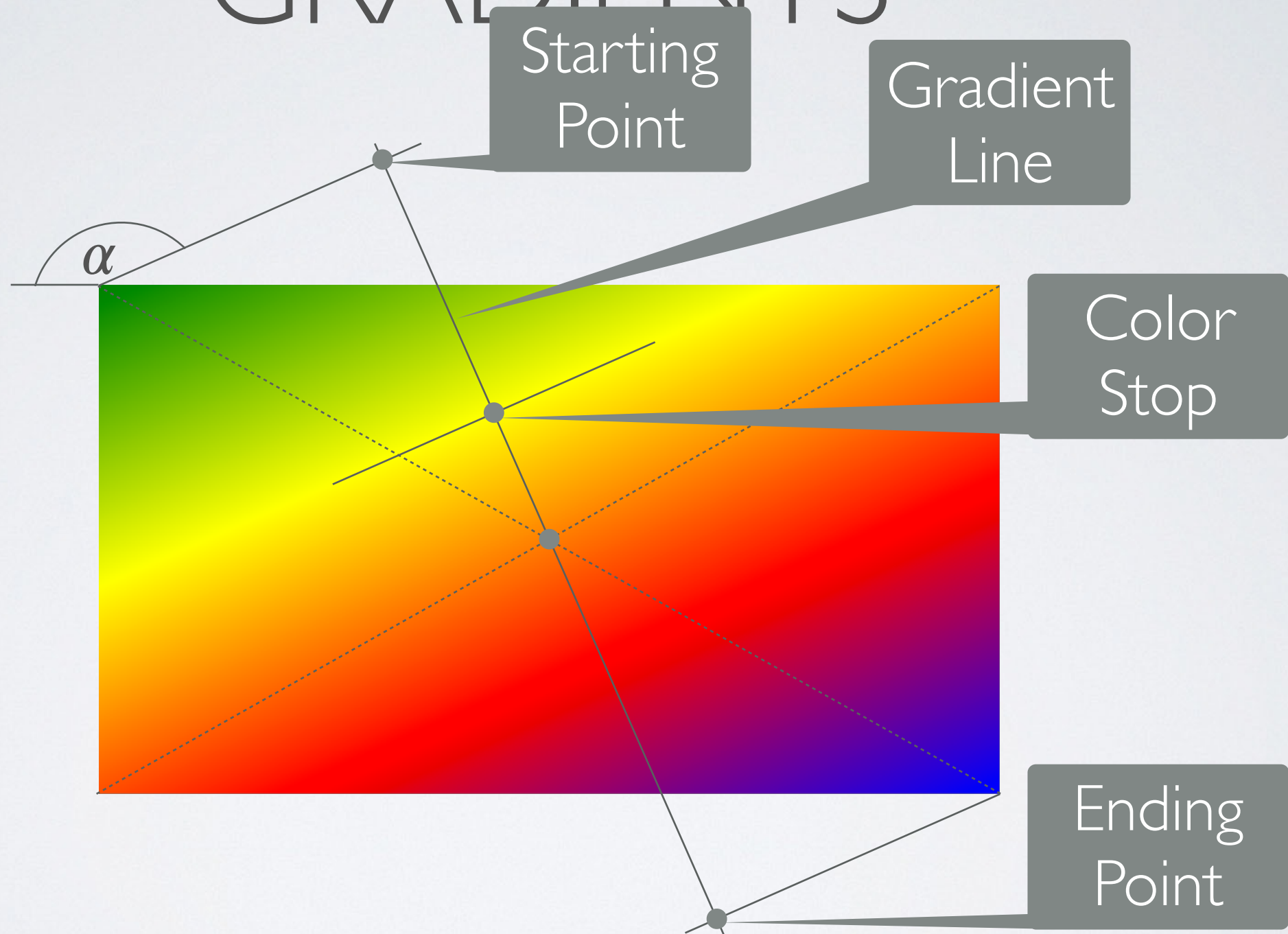
UNDERSTANDING LINEAR GRADIENTS



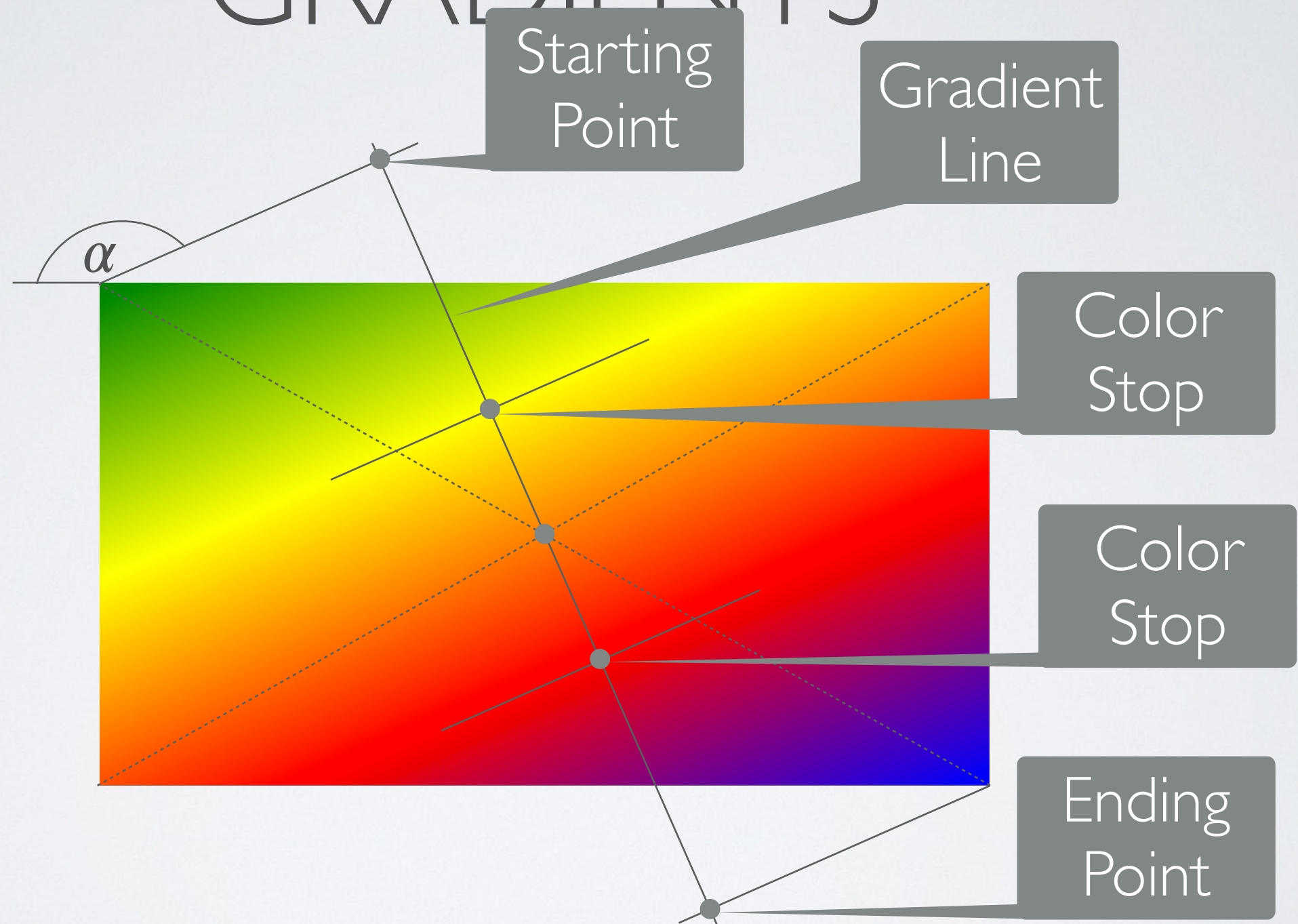
UNDERSTANDING LINEAR GRADIENTS



UNDERSTANDING LINEAR GRADIENTS



UNDERSTANDING LINEAR GRADIENTS




DIRECTIONS AND COLOR STOPS


DIRECTIONS AND COLOR STOPS

- Directions
 - to (left | right | top |
bottom)


DIRECTIONS AND COLOR STOPS

- Directions  or
 - to (left | right | top |
bottom)


DIRECTIONS AND COLOR STOPS

- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)


DIRECTIONS AND COLOR STOPS

- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)
 - angle

DIRECTIONS AND COLOR STOPS

- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)
 - angle
- Color stop

DIRECTIONS AND COLOR STOPS

- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)
 - angle
- Color stop
 - color

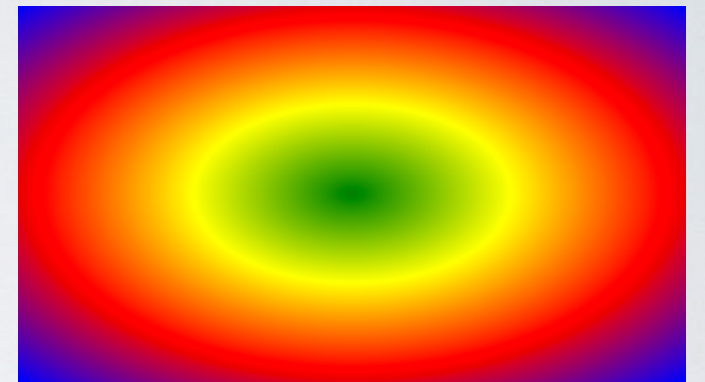
DIRECTIONS AND COLOR STOPS

- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)
 - angle
- Color stop
 - color
 - color percentage

DIRECTIONS AND COLOR STOPS

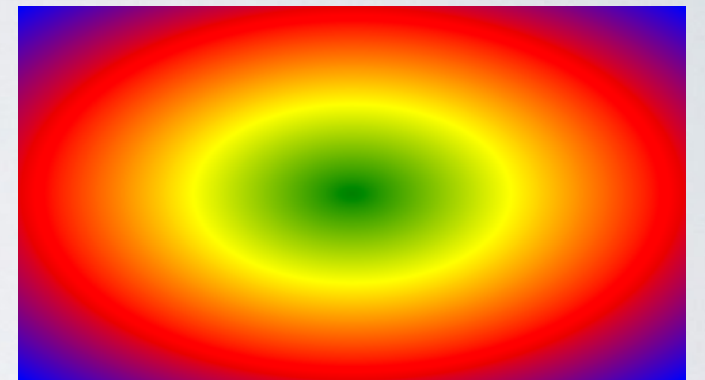
- Directions 
 - to (left | right | top | bottom)
 - to (top left | top right | bottom left | bottom right)
 - angle
- Color stop
 - color
 - color percentage
 - color length

RADIAL GRADIENT SHAPE



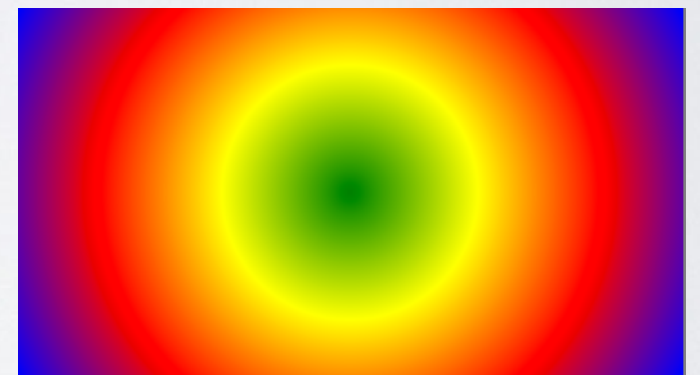
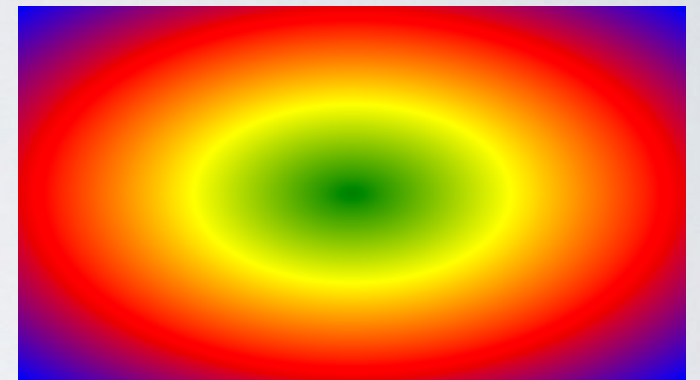
RADIAL GRADIENT SHAPE

```
#oval-radialgradient {  
  background-image:  
    radial-gradient(ellipse, red, green, blue);  
}
```



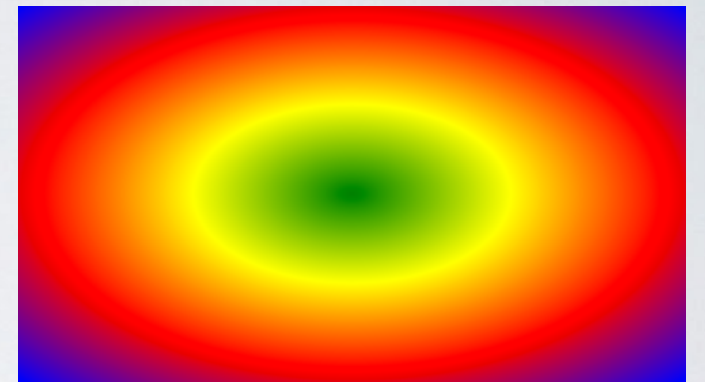
RADIAL GRADIENT SHAPE

```
#oval-radialgradient {  
  background-image:  
    radial-gradient(ellipse, red, green, blue);  
}
```

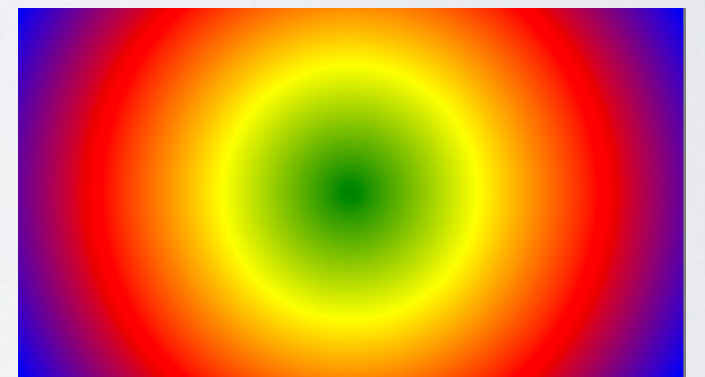


RADIAL GRADIENT SHAPE

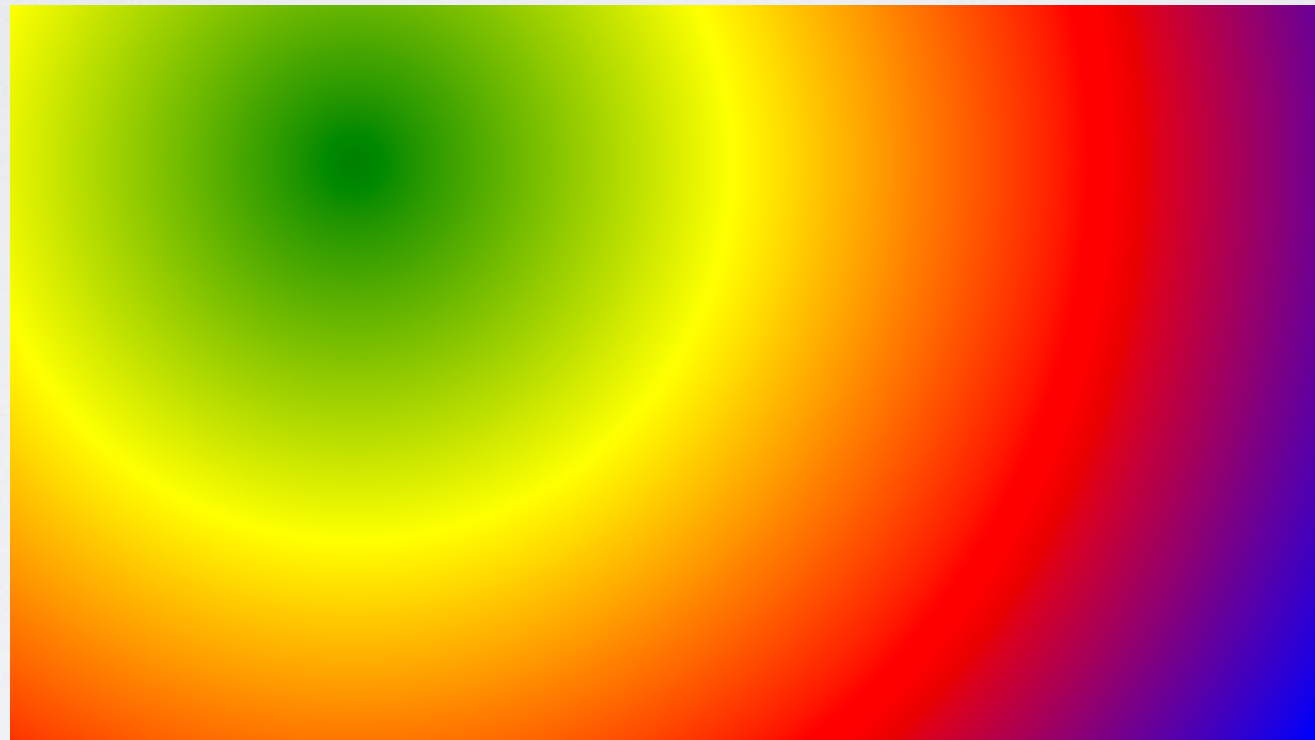
```
#oval-radialgradient {  
  background-image:  
    radial-gradient(ellipse, red, green, blue);  
}
```



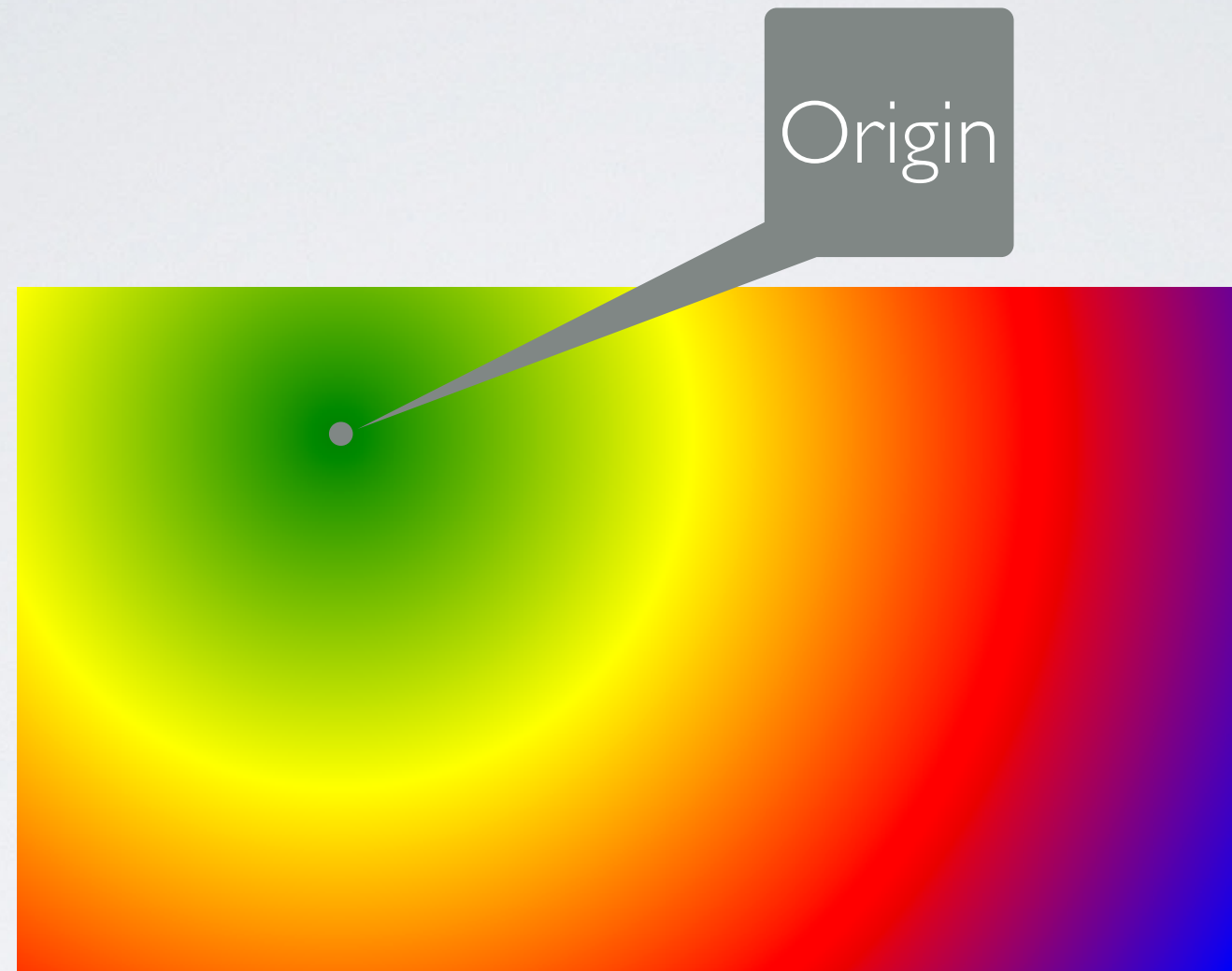
```
#circular-radialgradient {  
  background-image:  
    radial-gradient(circle, red, green, blue);  
}
```



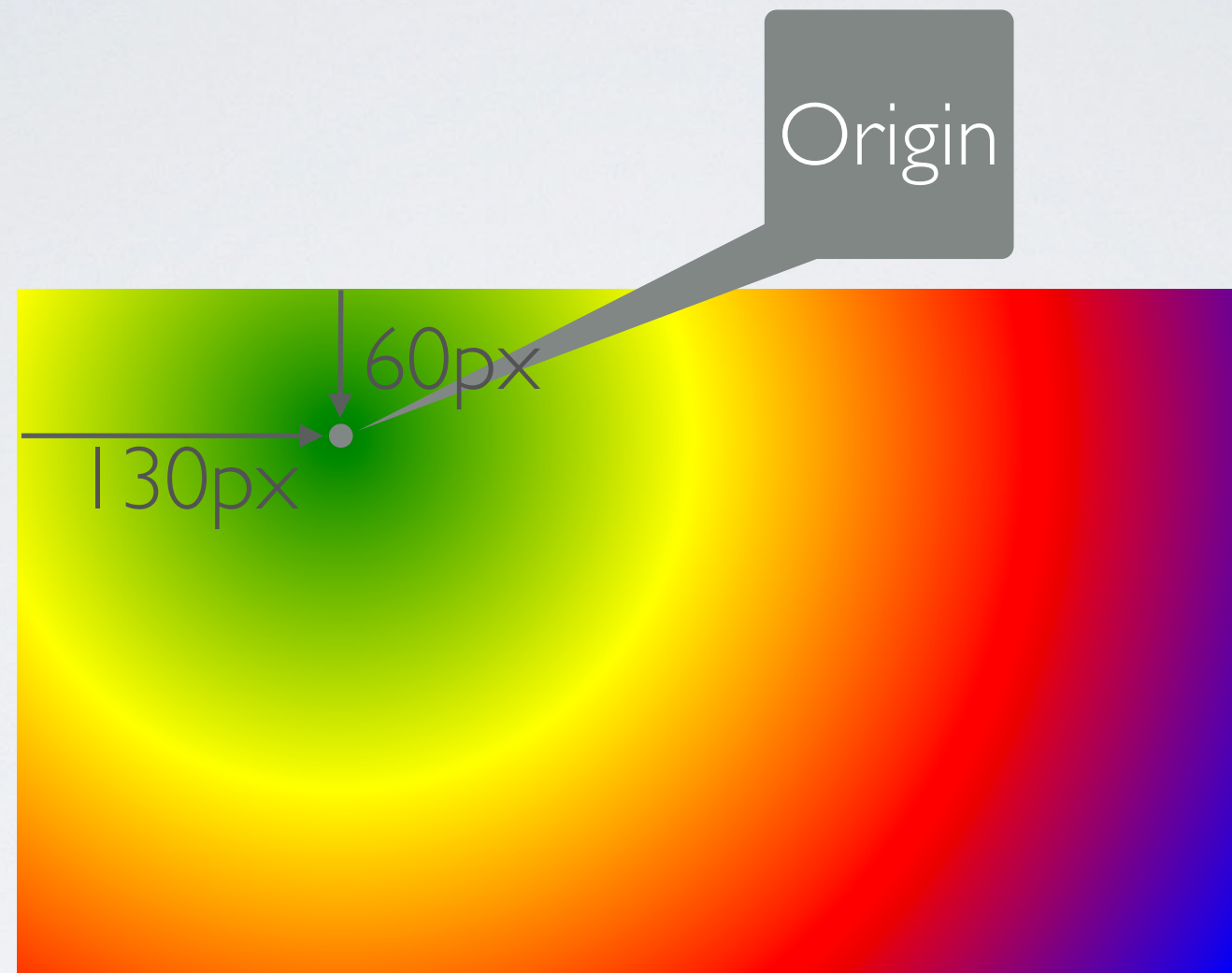
RADIAL GRADIENTS ORIGIN



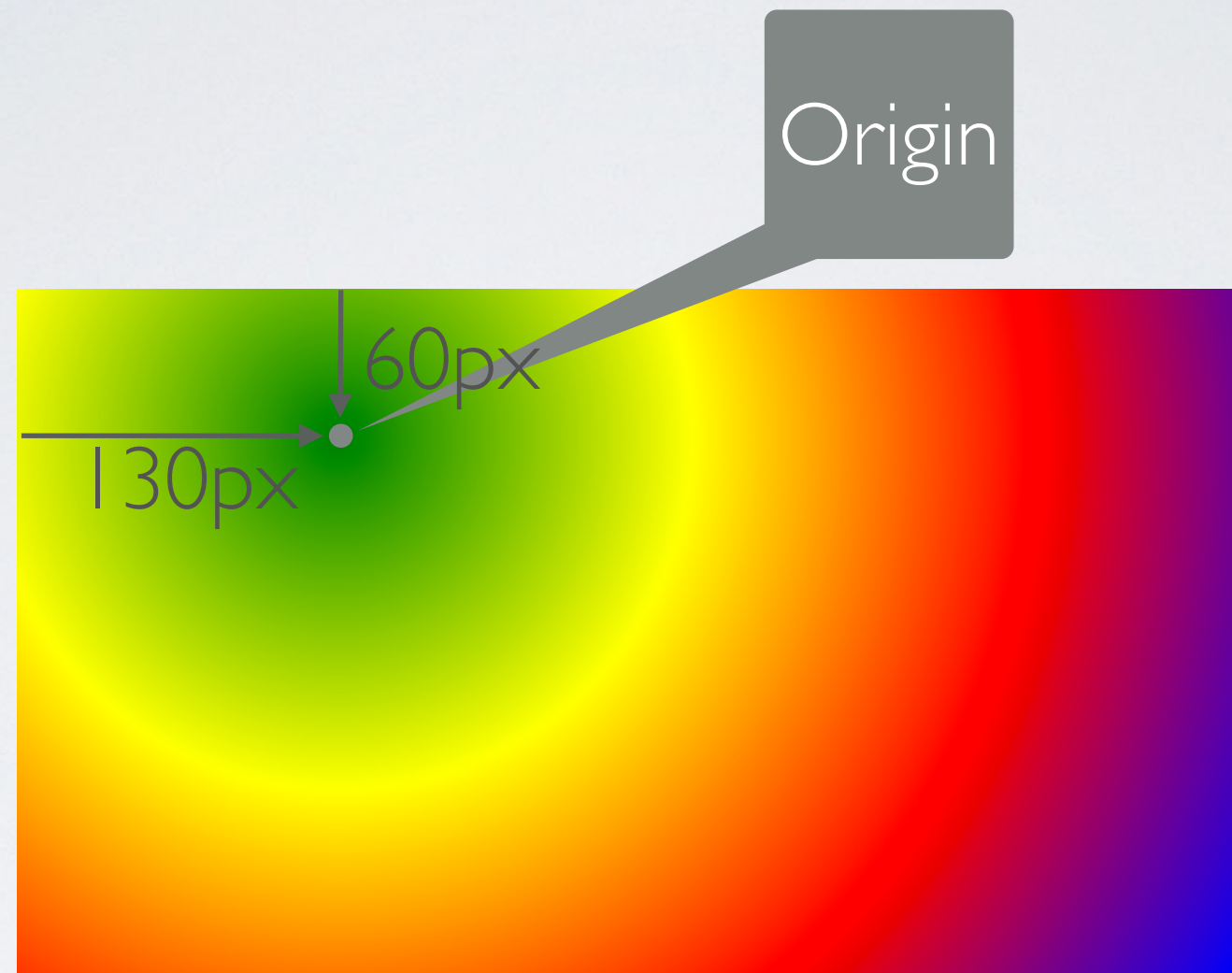
RADIAL GRADIENTS ORIGIN



RADIAL GRADIENTS ORIGIN

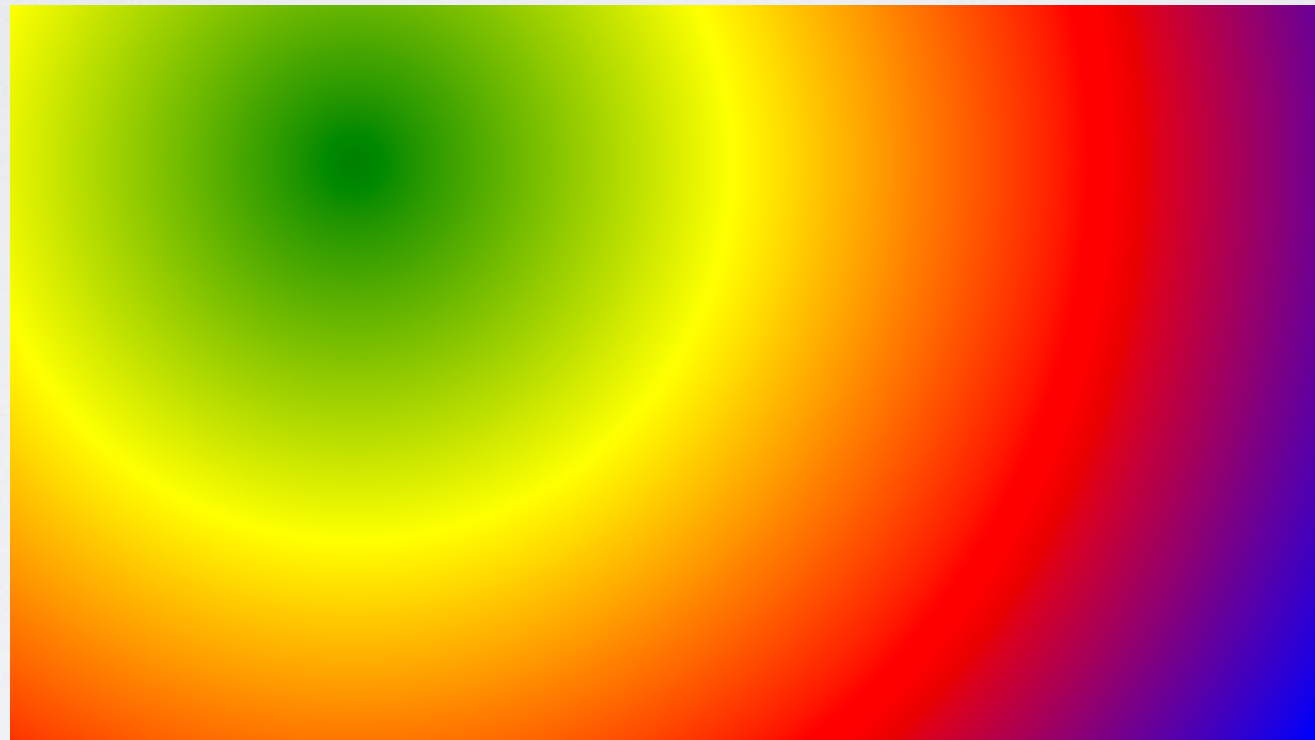


RADIAL GRADIENTS ORIGIN

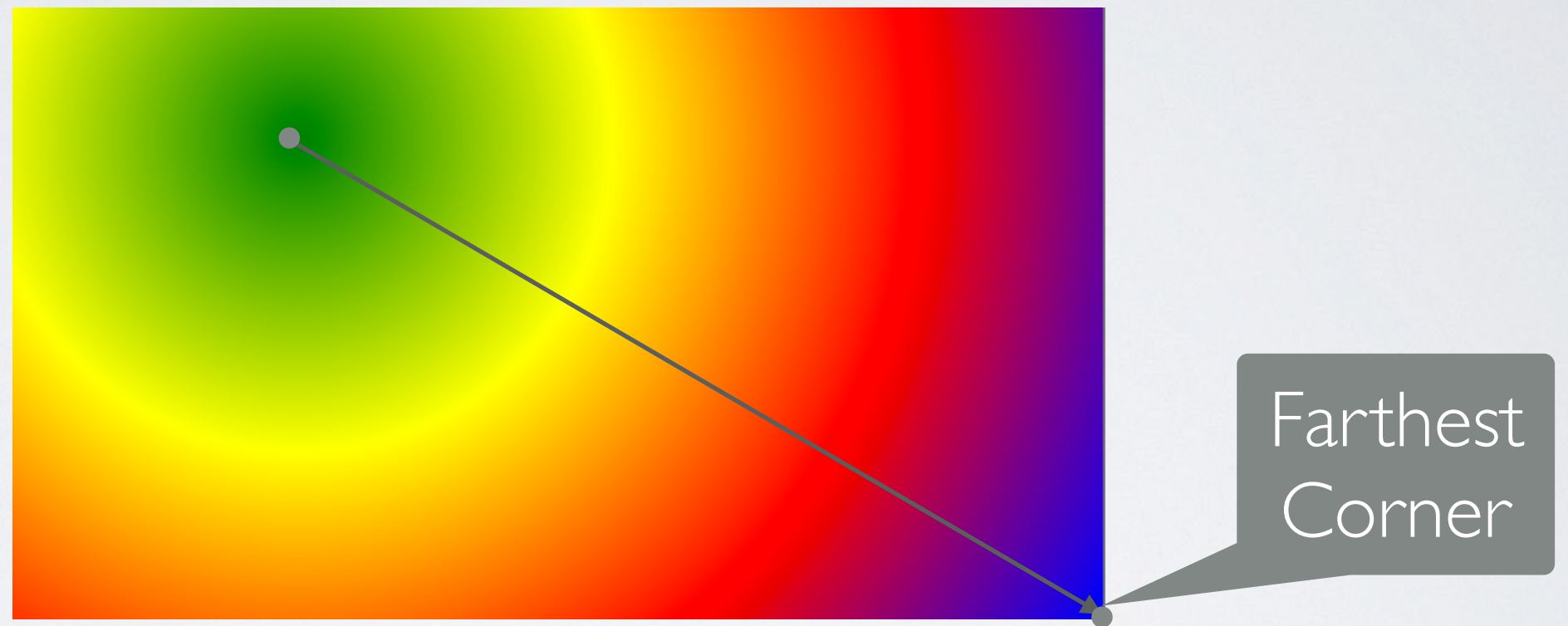


```
radial-gradient(circle at 130px 60px,  
green, yellow, red, blue);
```

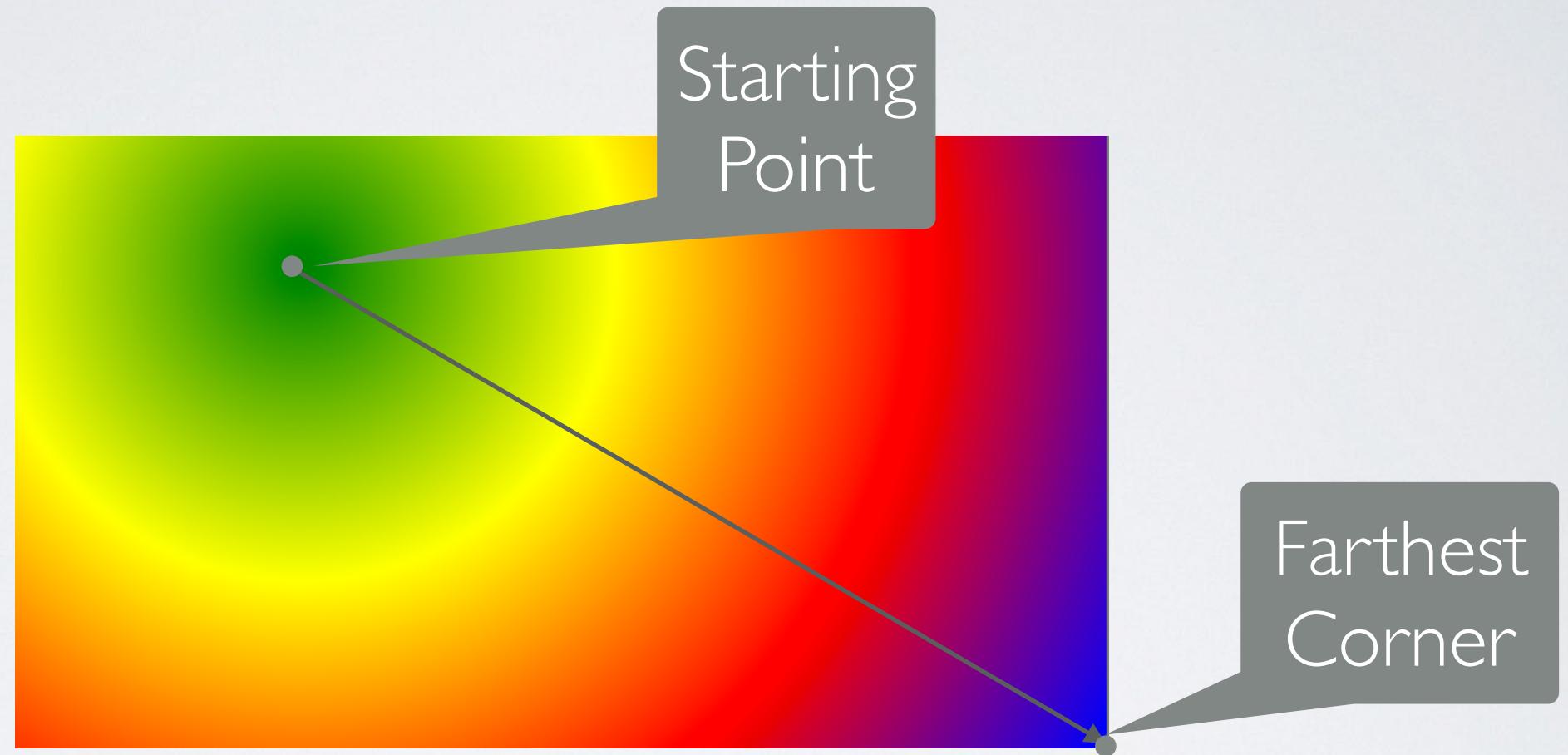

RADIAL GRADIENTS SIZE



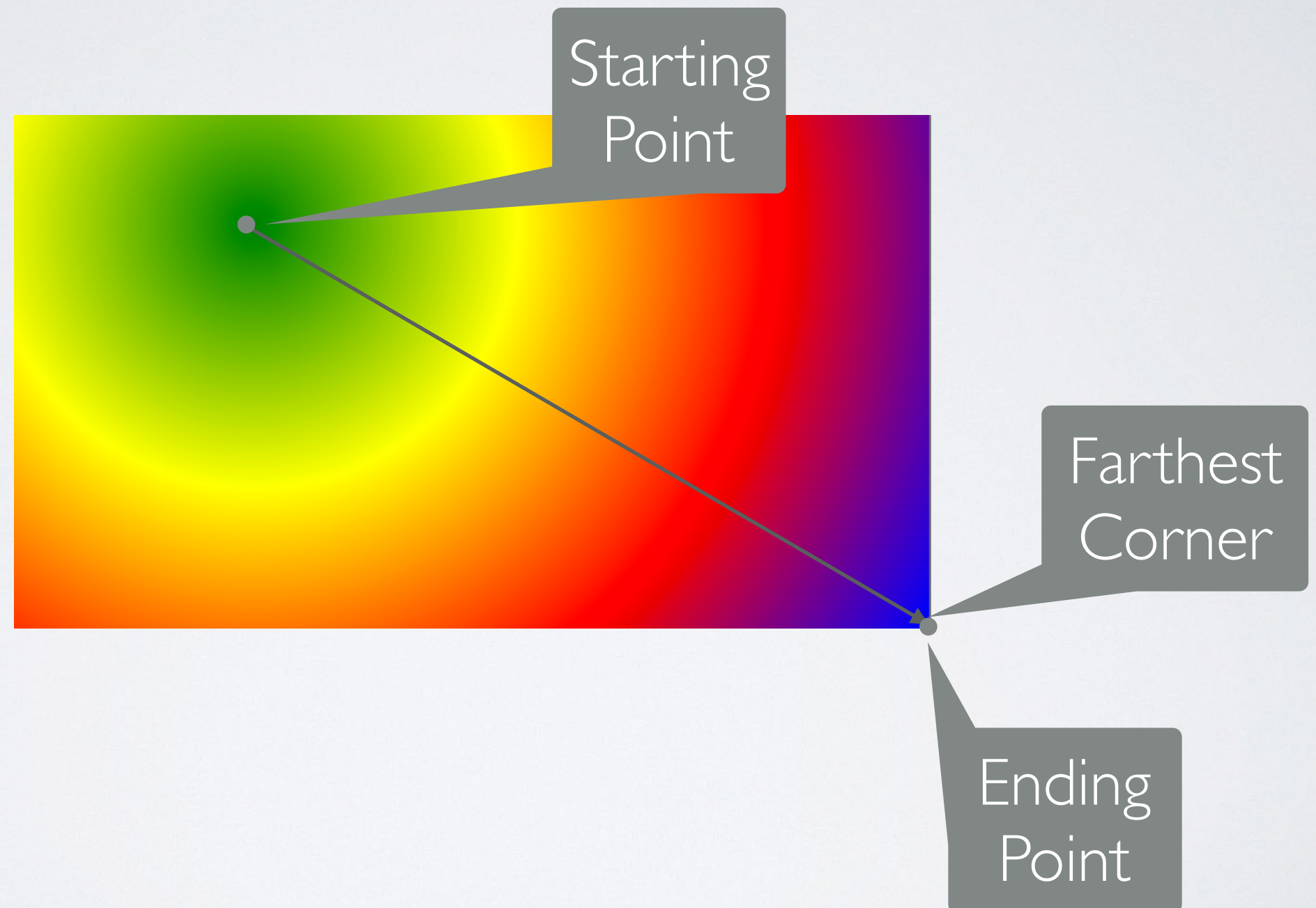
RADIAL GRADIENTS SIZE



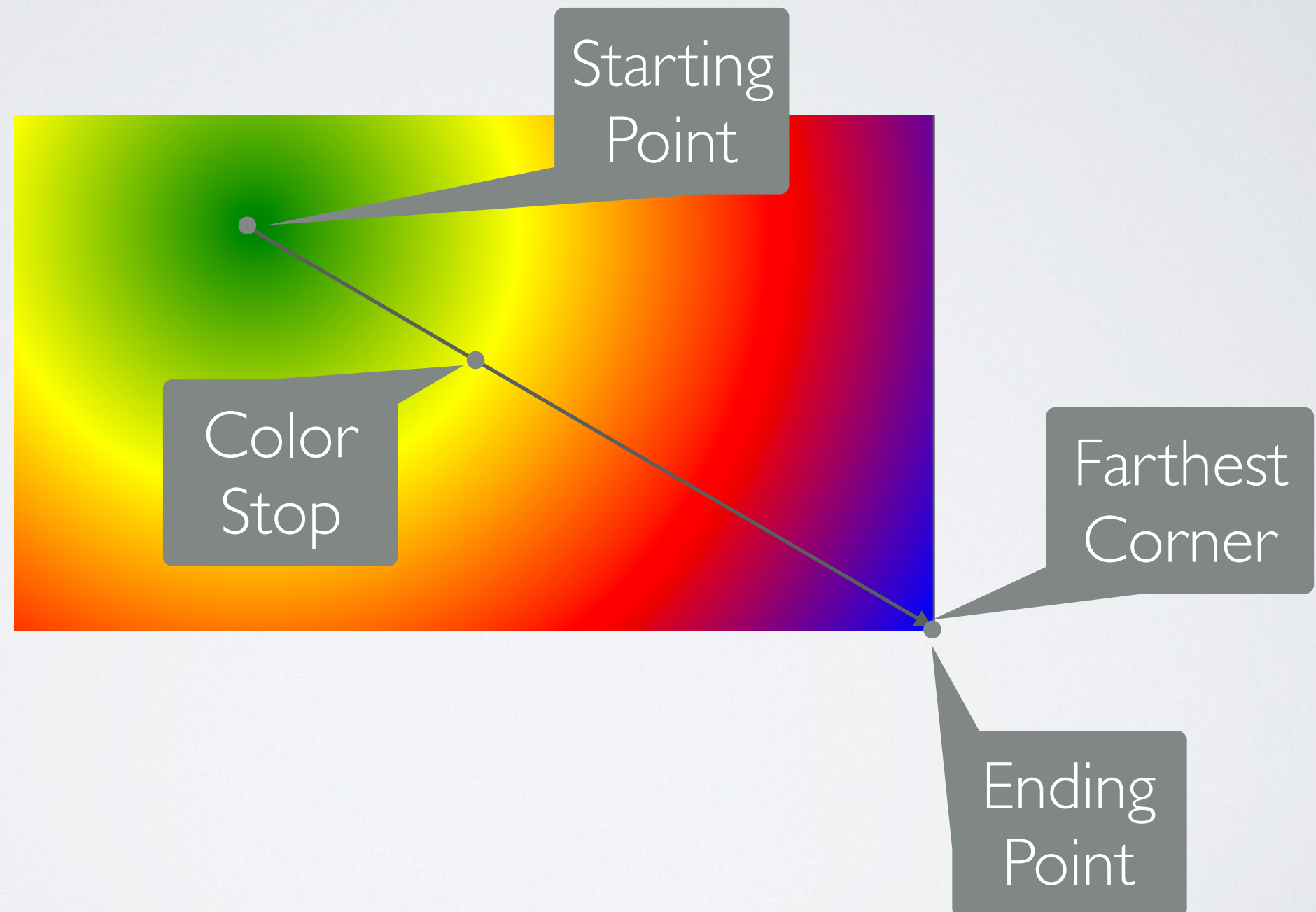
RADIAL GRADIENTS SIZE



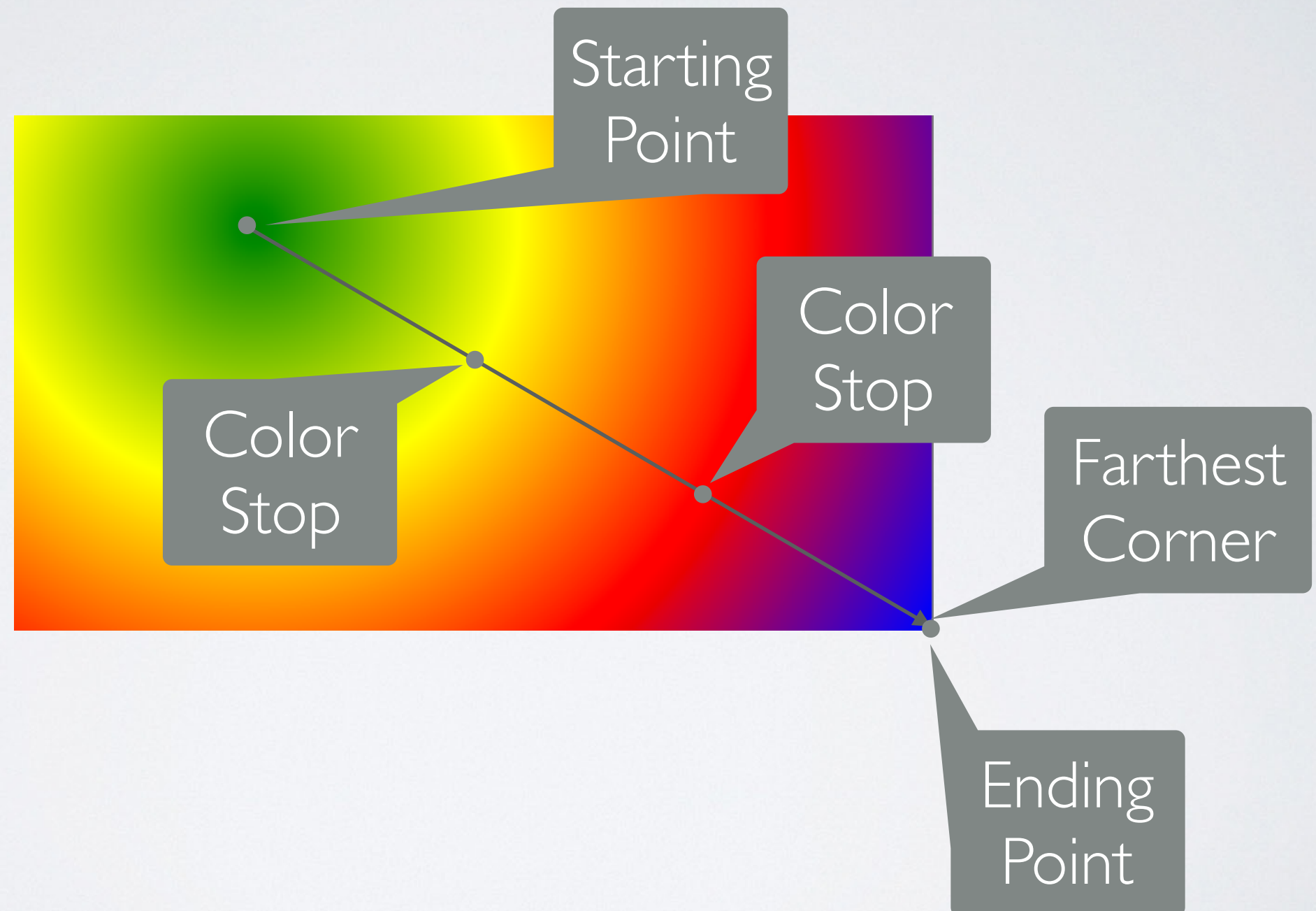
RADIAL GRADIENTS SIZE



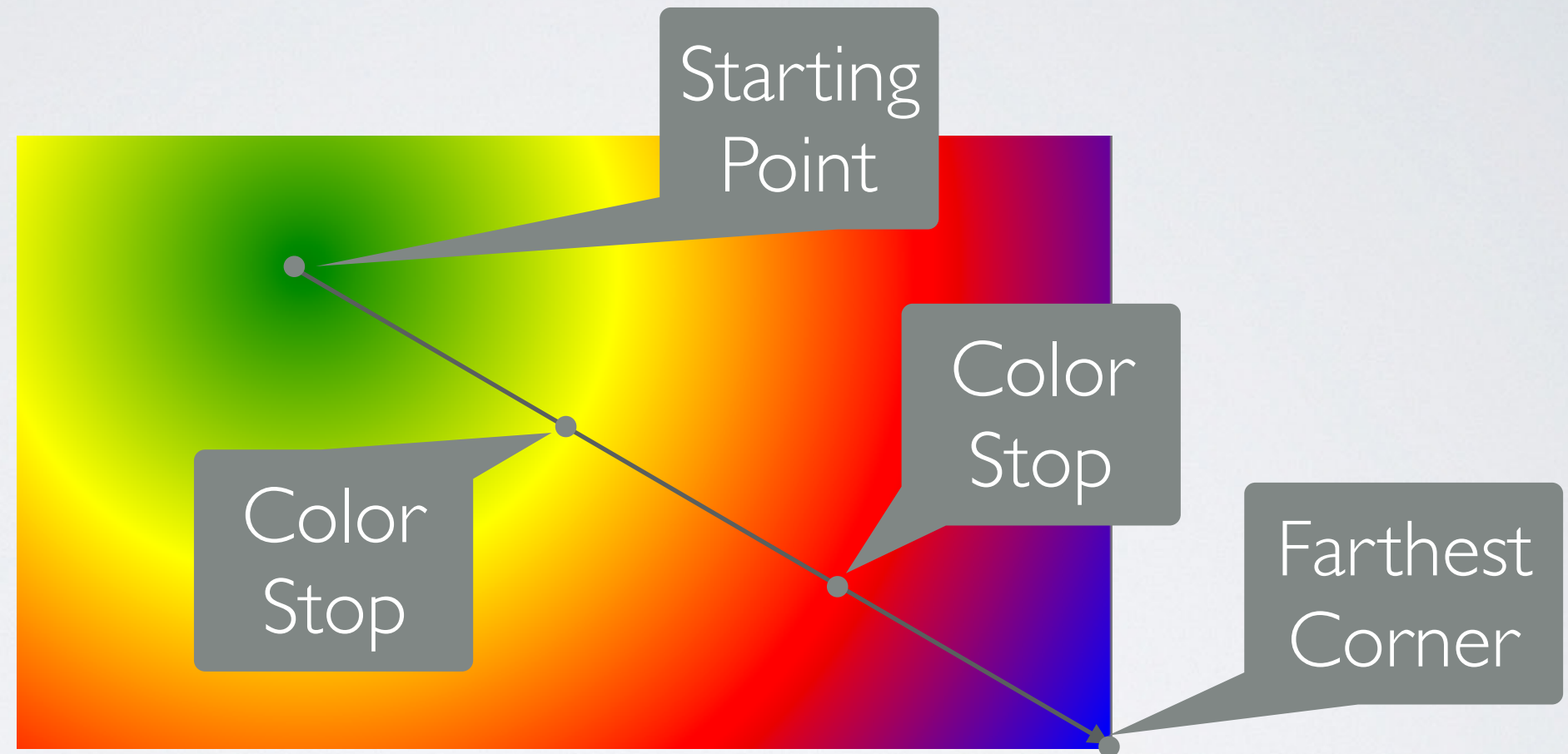
RADIAL GRADIENTS SIZE



RADIAL GRADIENTS SIZE

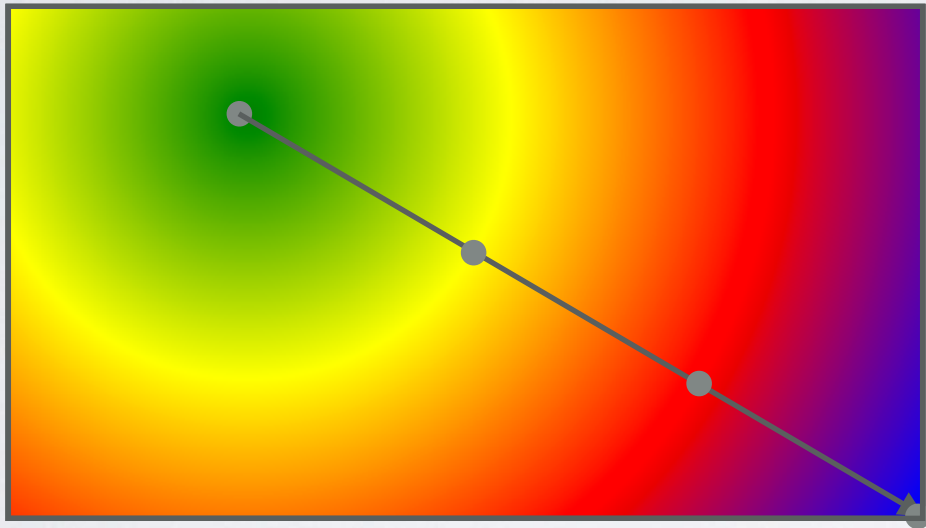


RADIAL GRADIENTS SIZE

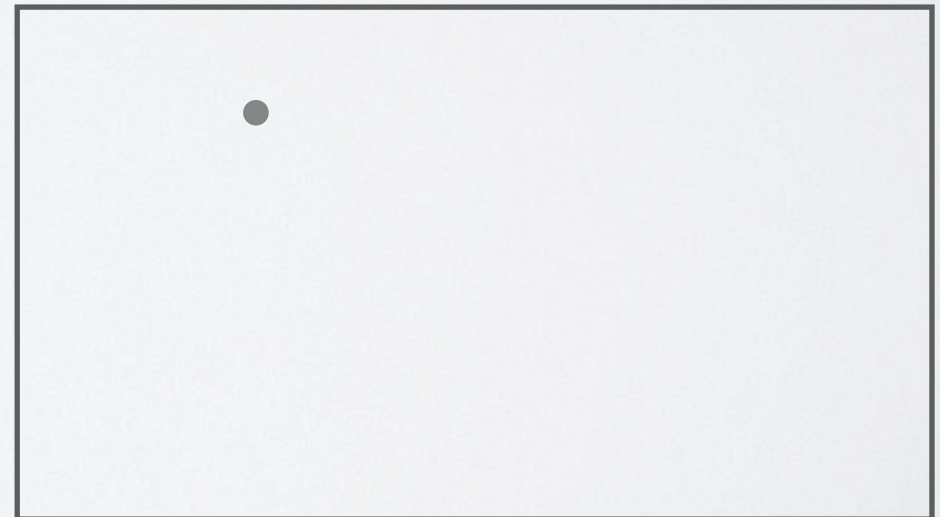
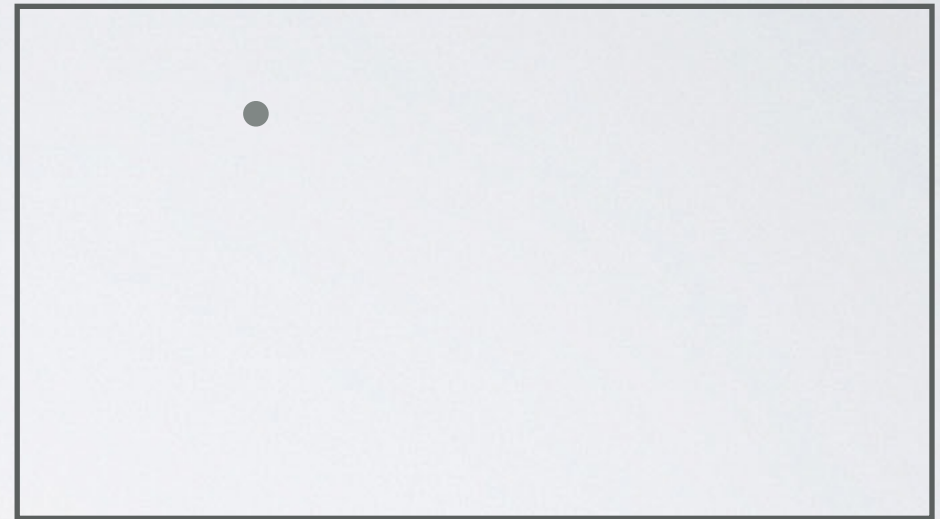


```
radial-gradient(circle farthest-corner  
at 130px 60px,  
green, yellow, red, blue);
```

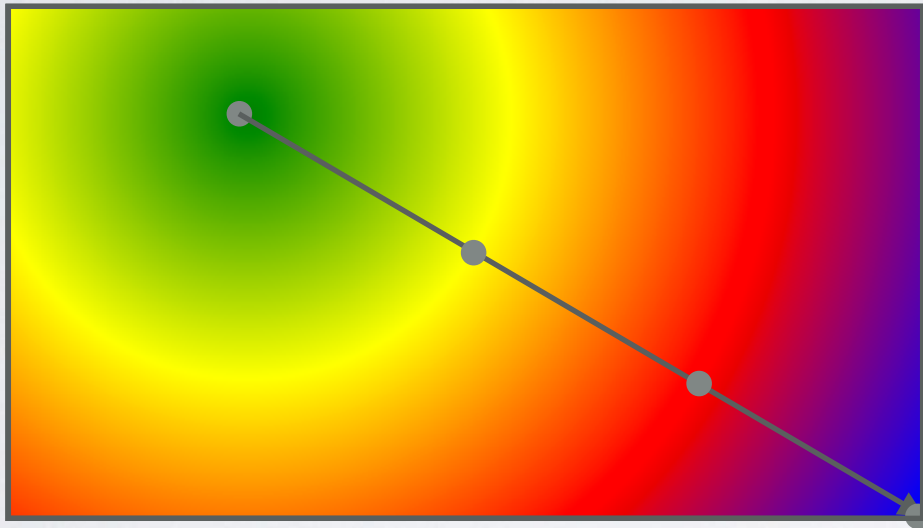
HOW WILL THE OTHERS LOOK LIKE?



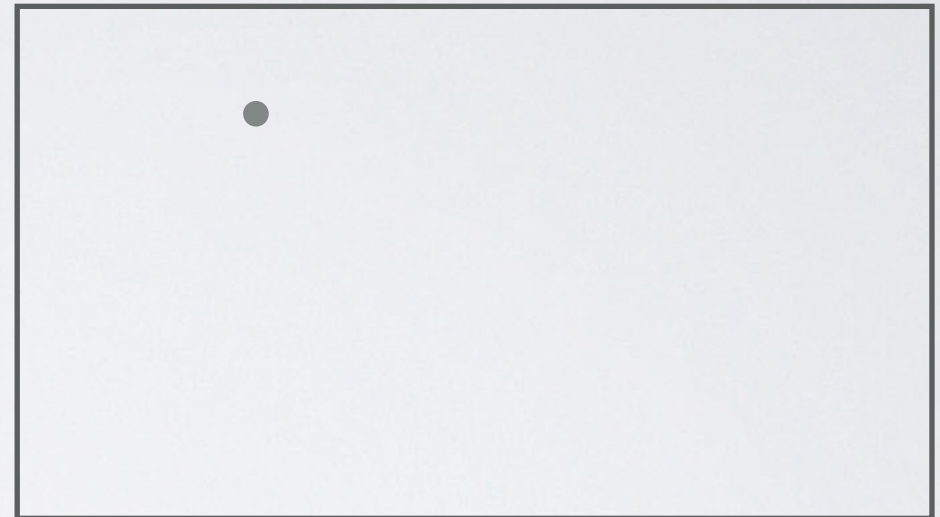
```
radial-gradient(circle farthest-corner  
at 130px 60px,  
green, yellow, red, blue);
```



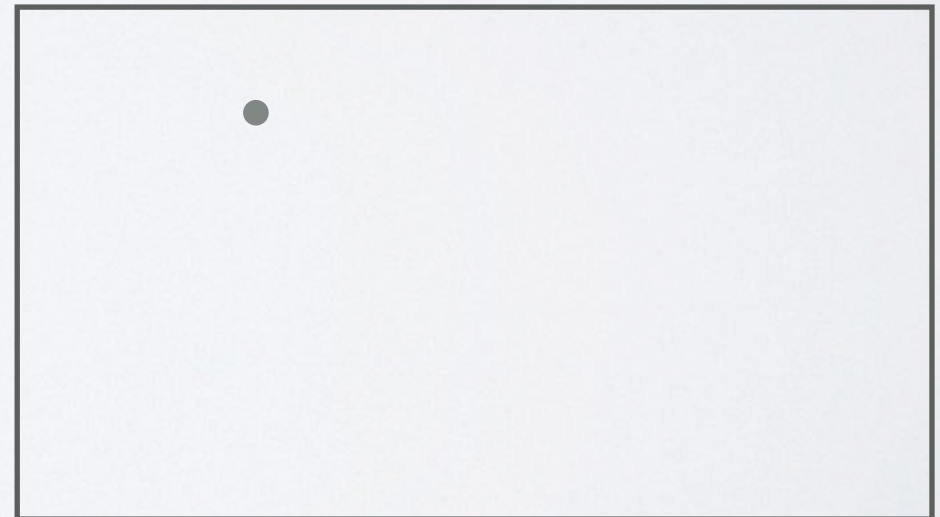
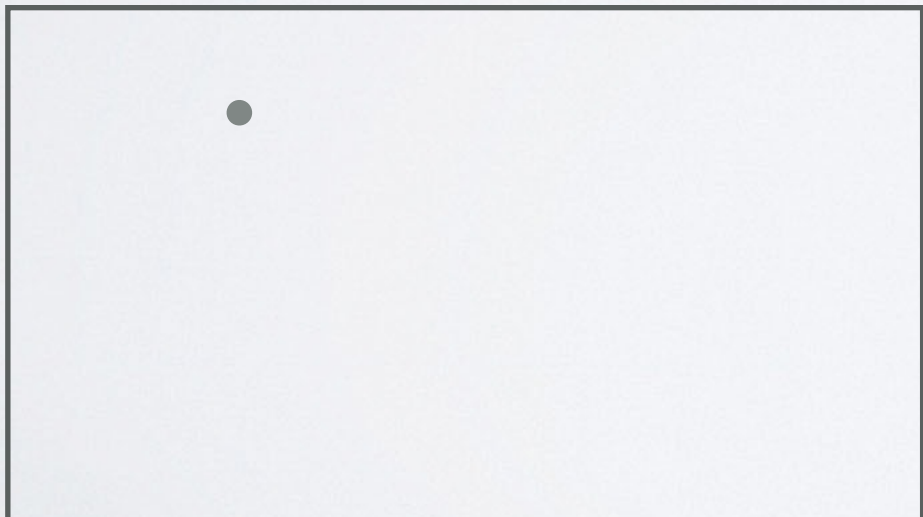
HOW WILL THE OTHERS LOOK LIKE?



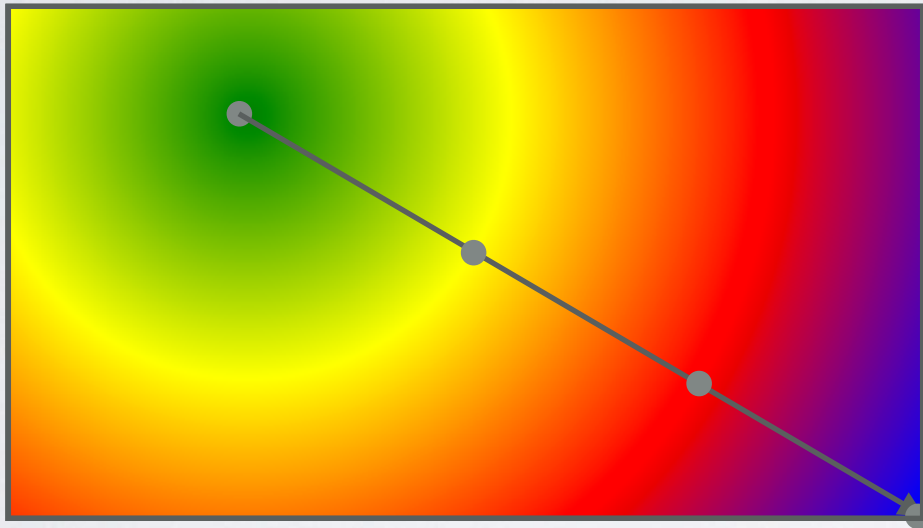
radial-gradient(circle **farthest-corner**
at 130px 60px,
green, yellow, red, blue);



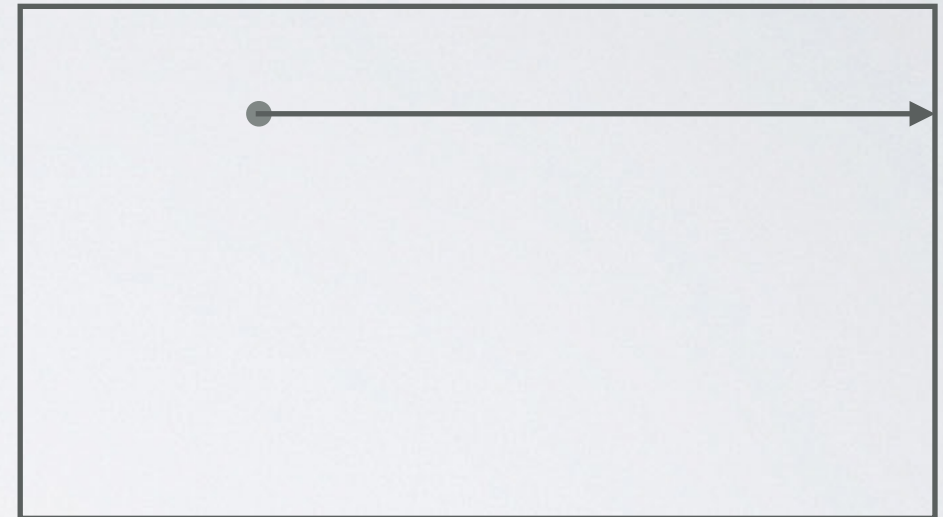
radial-gradient(circle **farthest-side**
at 130px 60px,
green, yellow, red, blue);



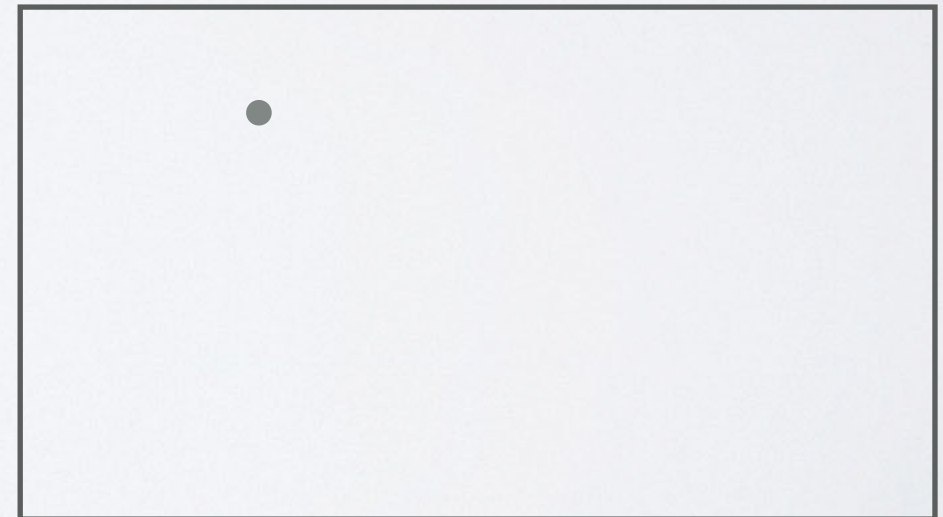
HOW WILL THE OTHERS LOOK LIKE?



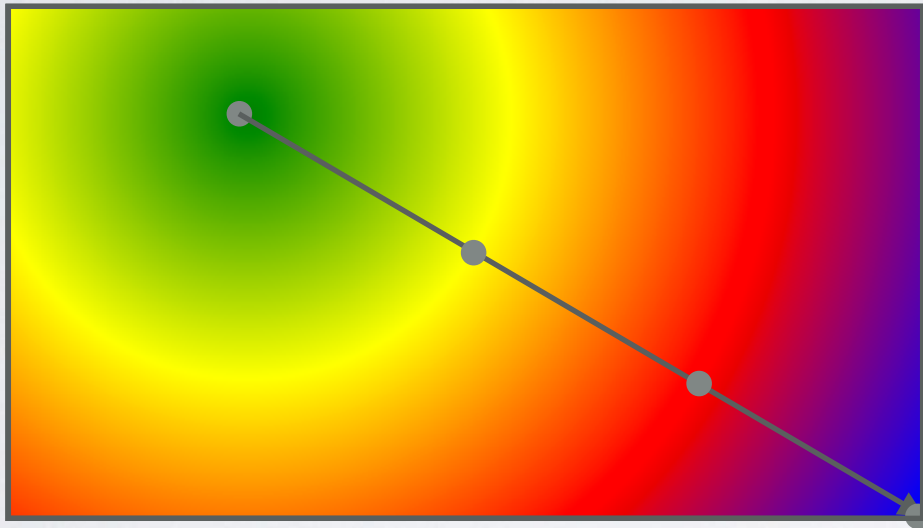
radial-gradient(circle **farthest-corner**
at 130px 60px,
green, yellow, red, blue);



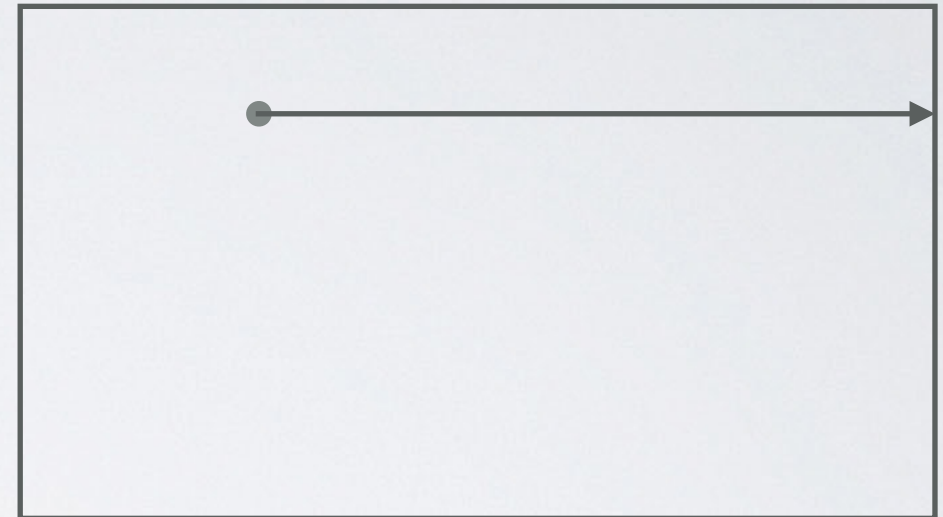
radial-gradient(circle **farthest-side**
at 130px 60px,
green, yellow, red, blue);



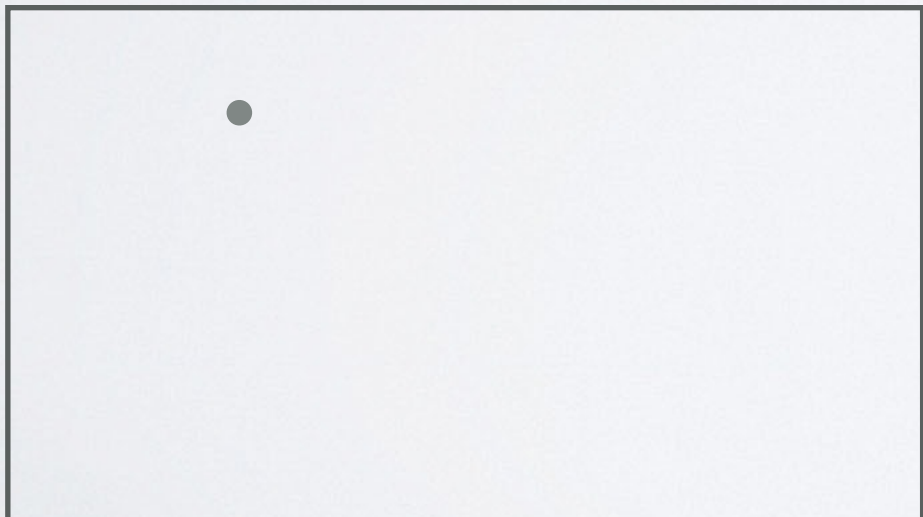
HOW WILL THE OTHERS LOOK LIKE?



```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



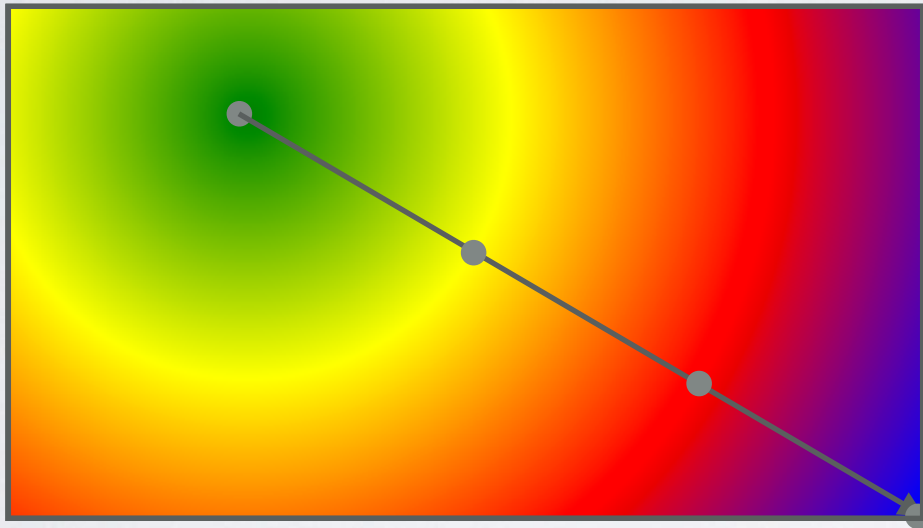
```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```



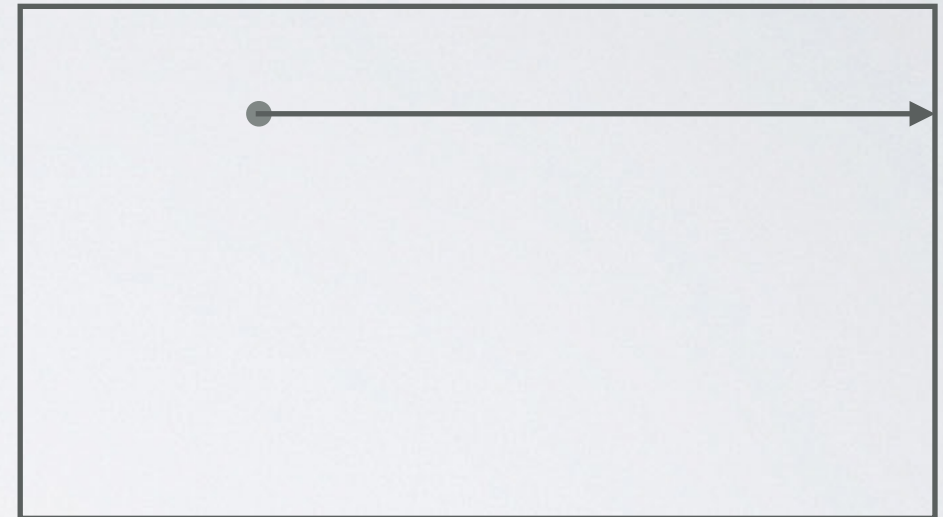
```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



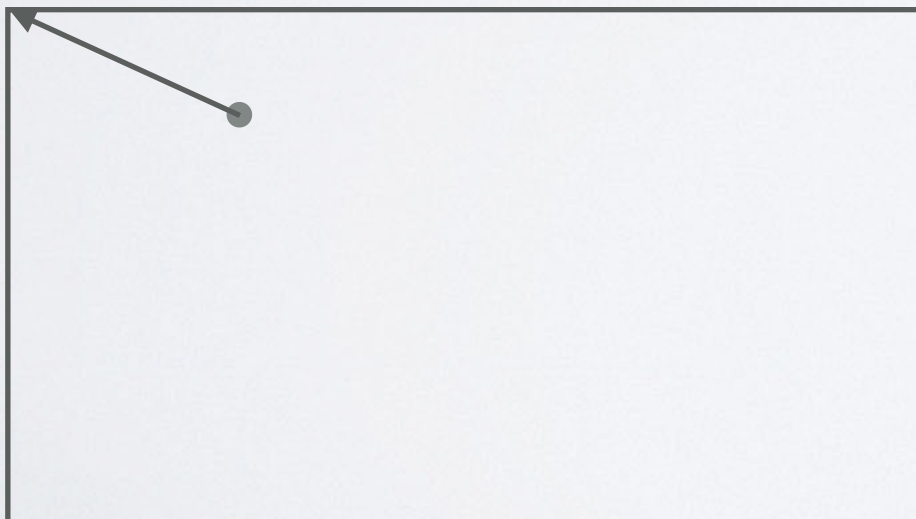
HOW WILL THE OTHERS LOOK LIKE?



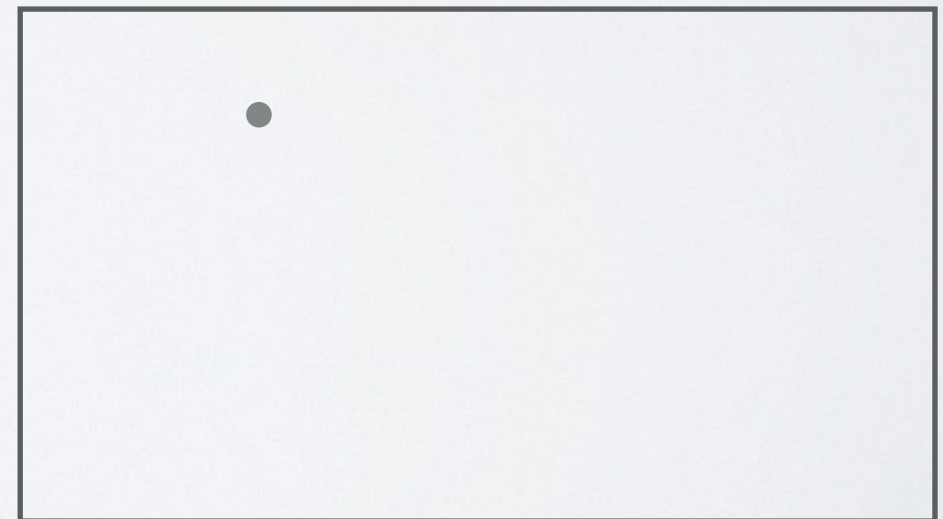
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



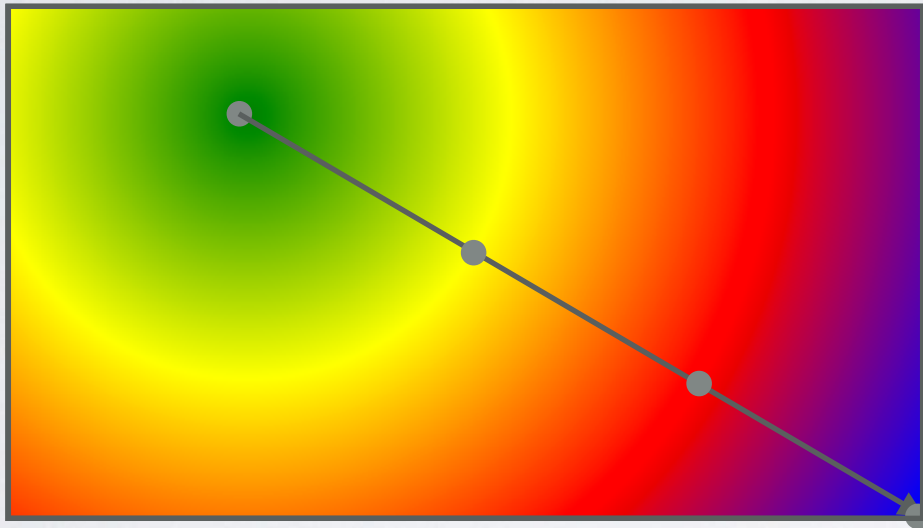
```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```



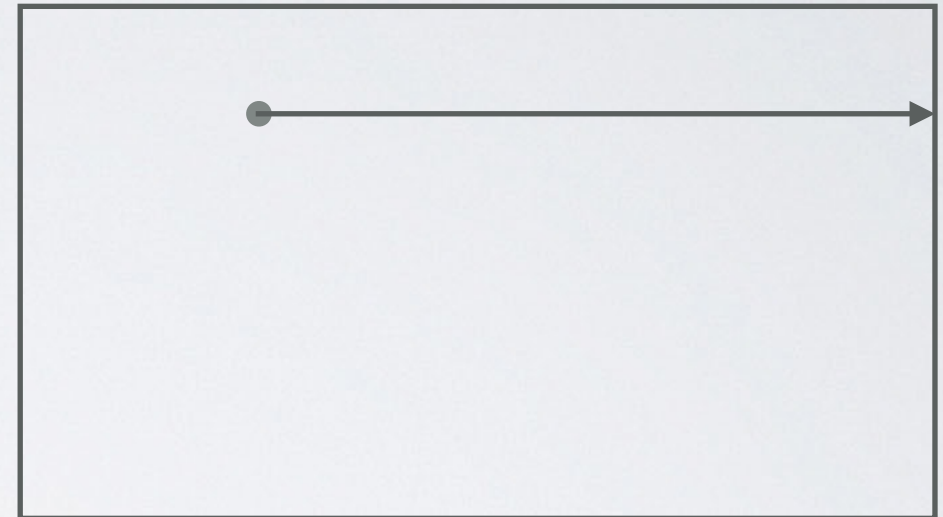
```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



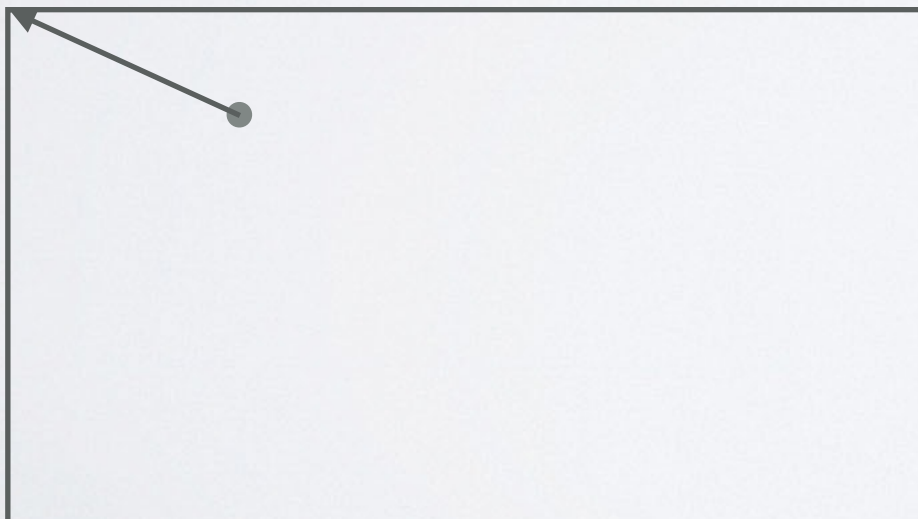
HOW WILL THE OTHERS LOOK LIKE?



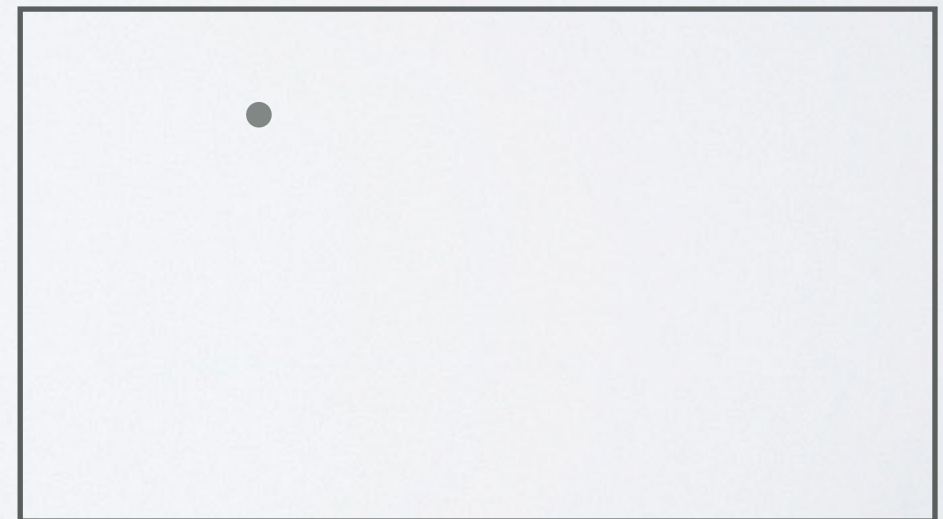
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

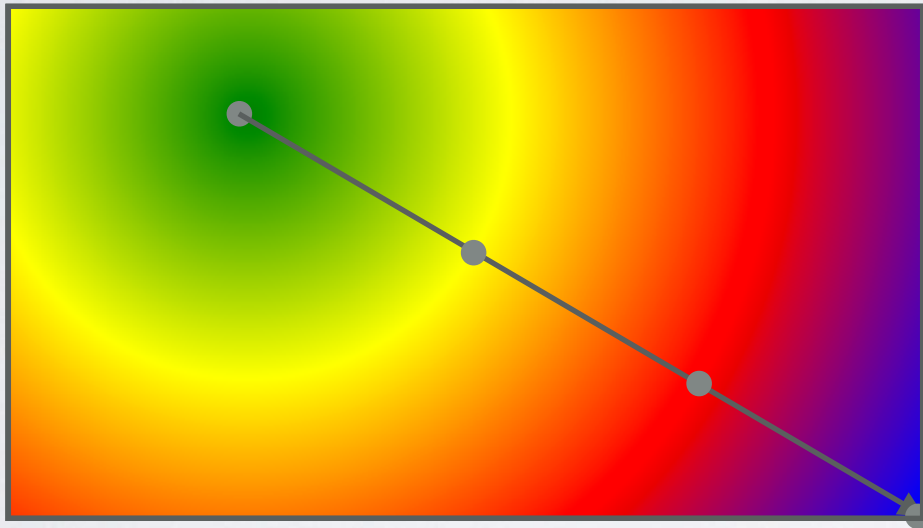


```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```

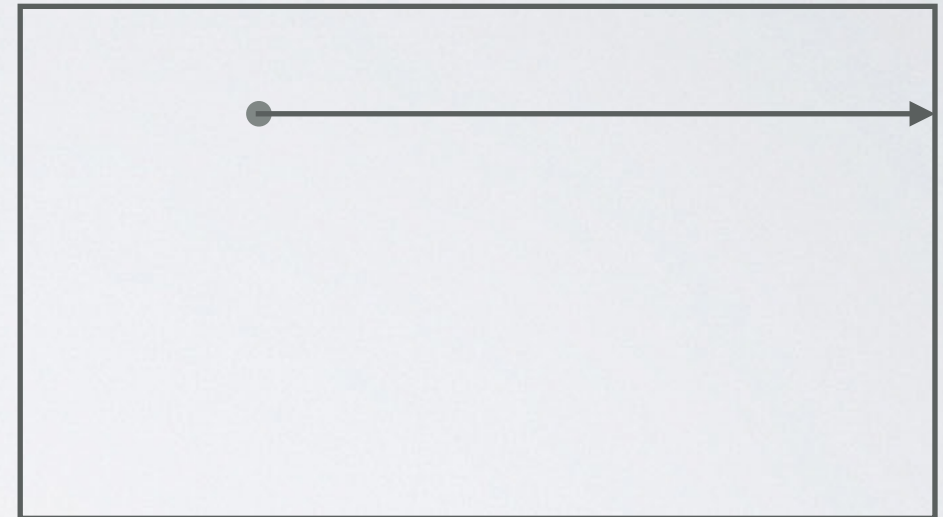


```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

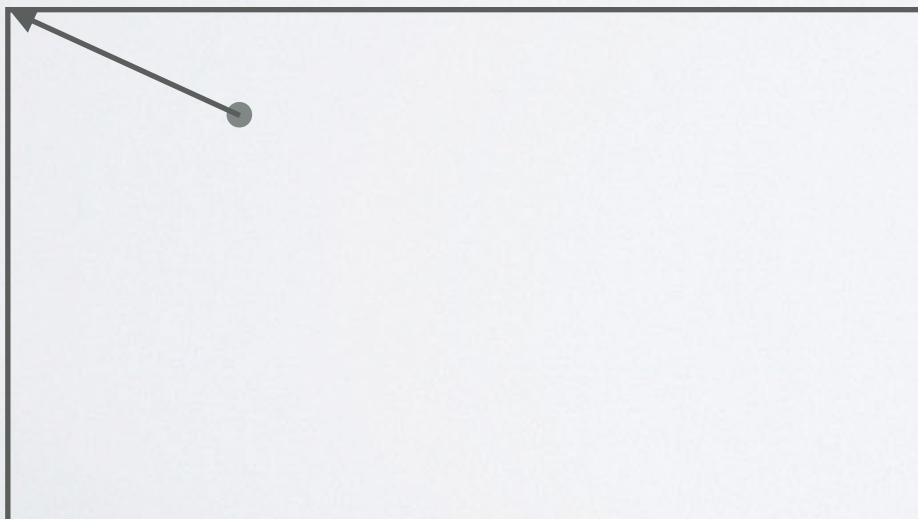
HOW WILL THE OTHERS LOOK LIKE?



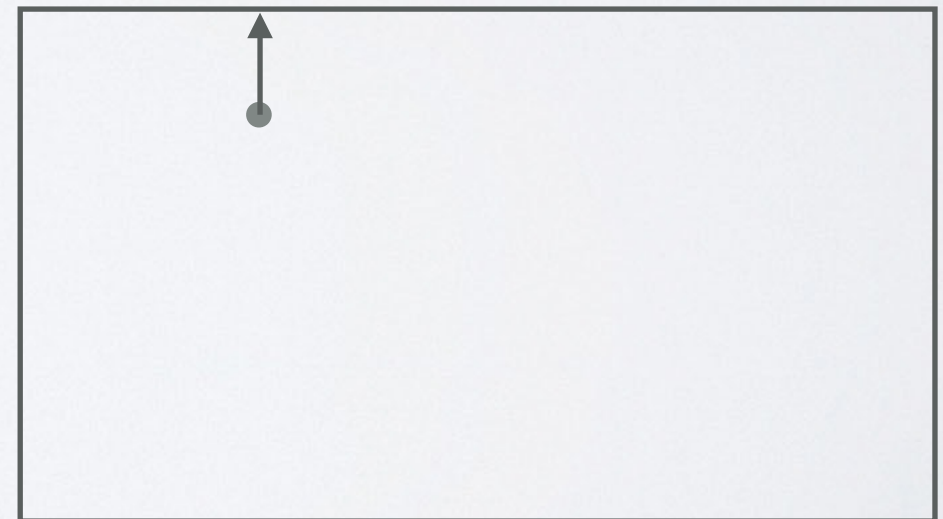
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

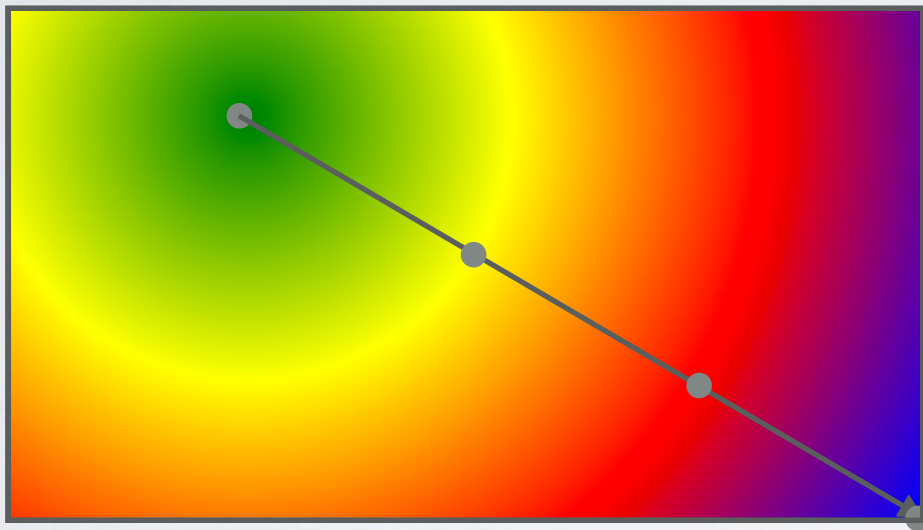


```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```

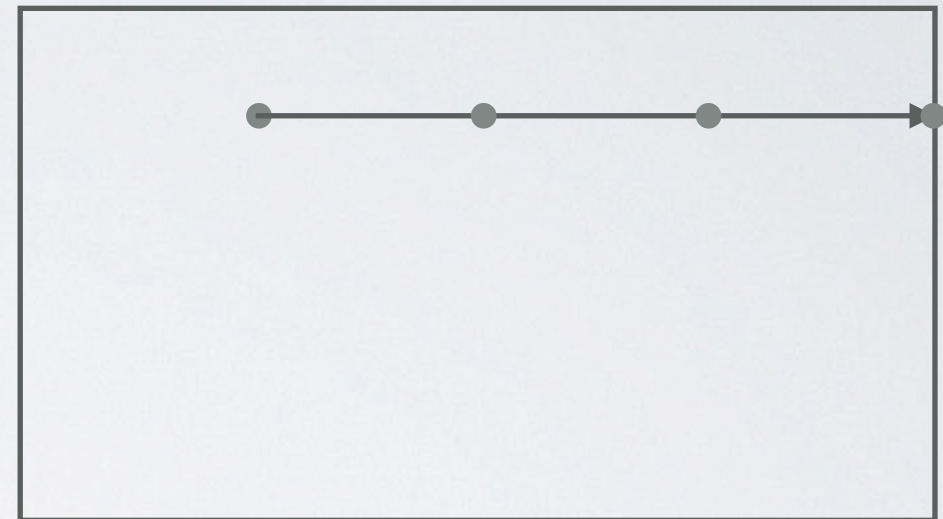


```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

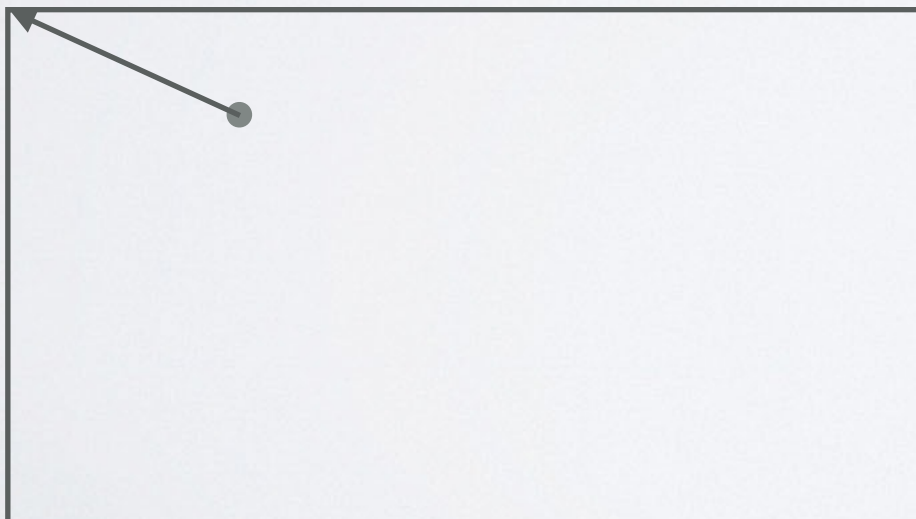

HOW WILL THE OTHERS LOOK LIKE?



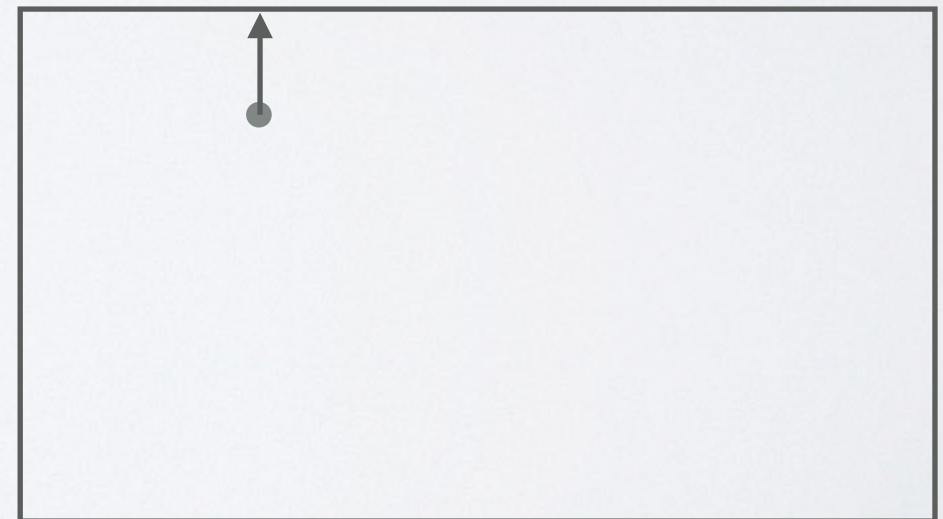
radial-gradient(circle **farthest-corner**
at 130px 60px,
green, yellow, red, blue);



radial-gradient(circle **farthest-side**
at 130px 60px,
green, yellow, red, blue);

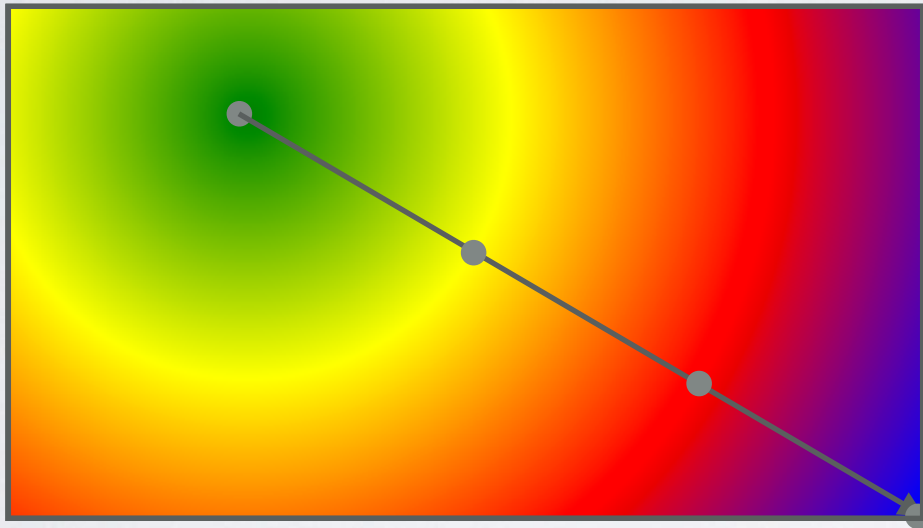


radial-gradient(circle **closest-corner**
at 130px 60px,
green, yellow, red, blue);

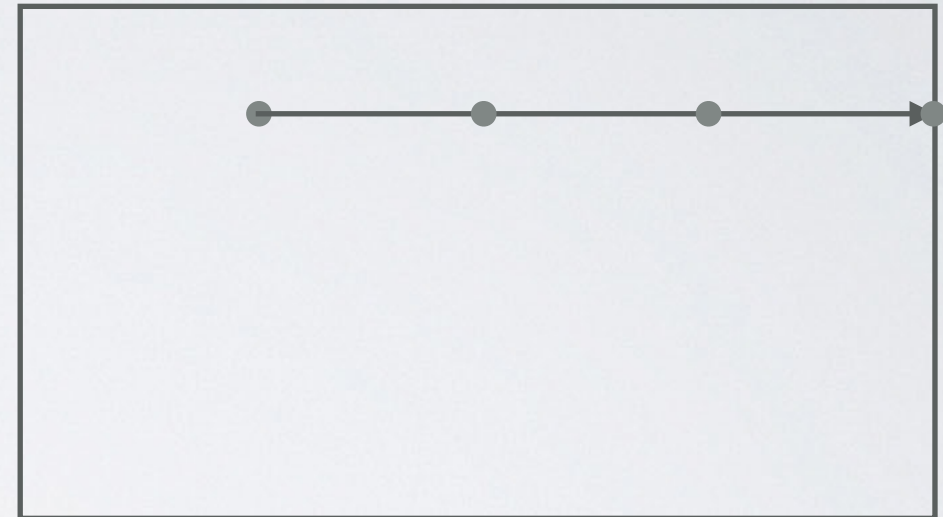


radial-gradient(circle **closest-side**
at 130px 60px,
green, yellow, red, blue);

HOW WILL THE OTHERS LOOK LIKE?



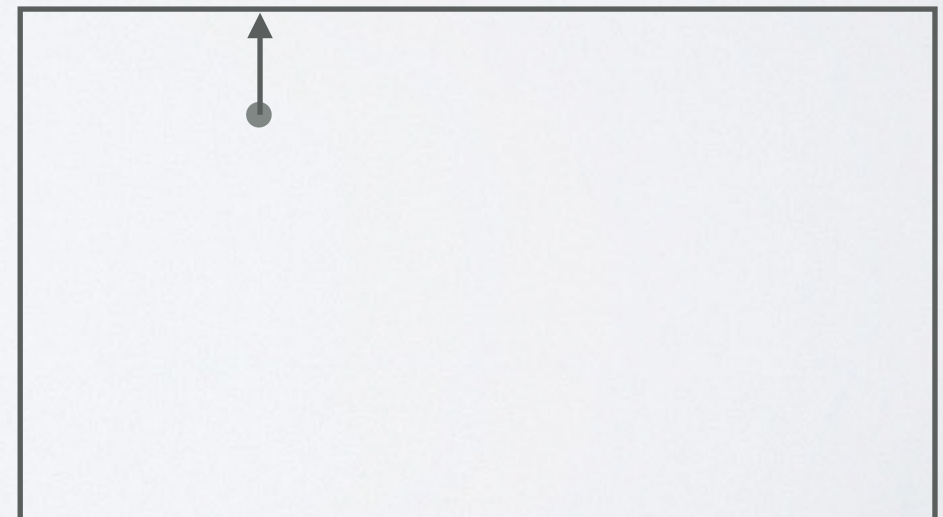
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

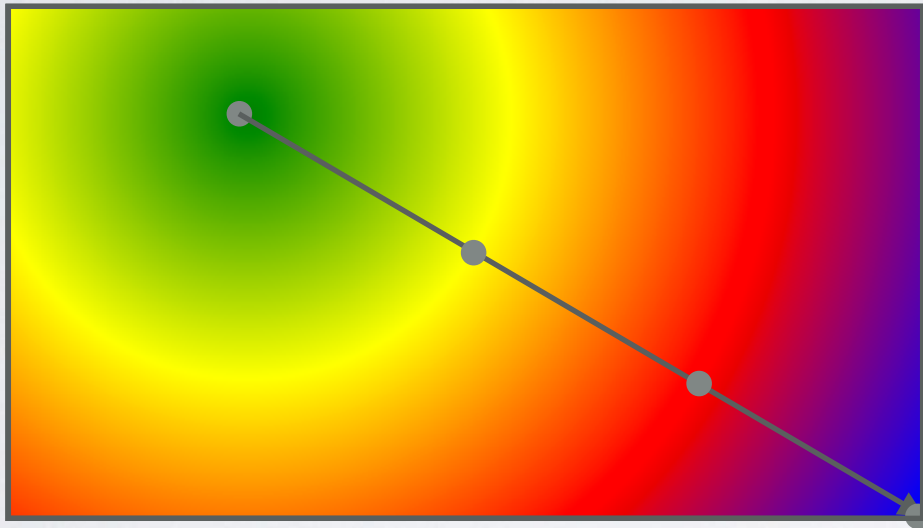


```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```

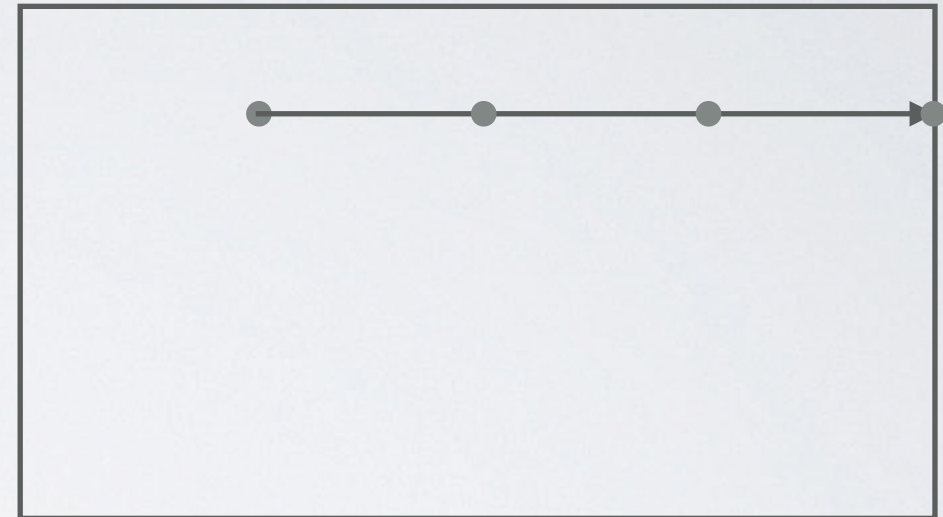


```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```


HOW WILL THE OTHERS LOOK LIKE?



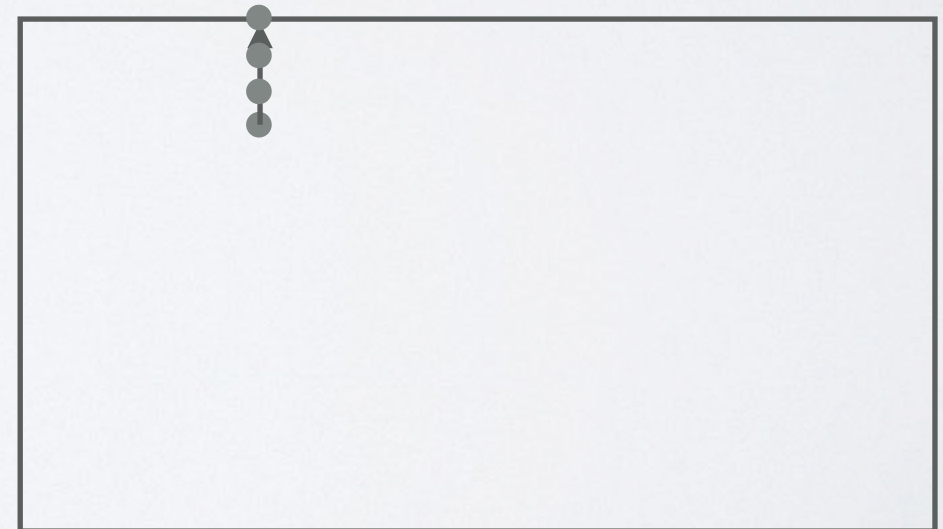
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

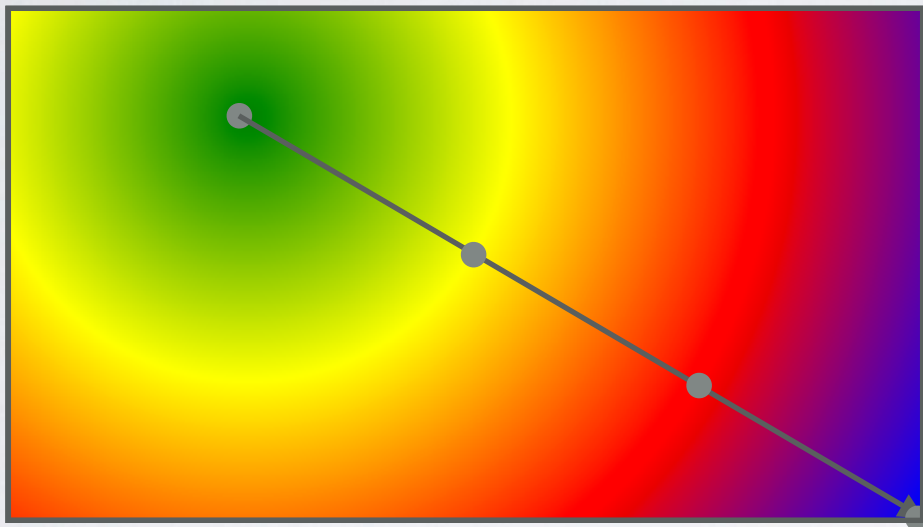


```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```

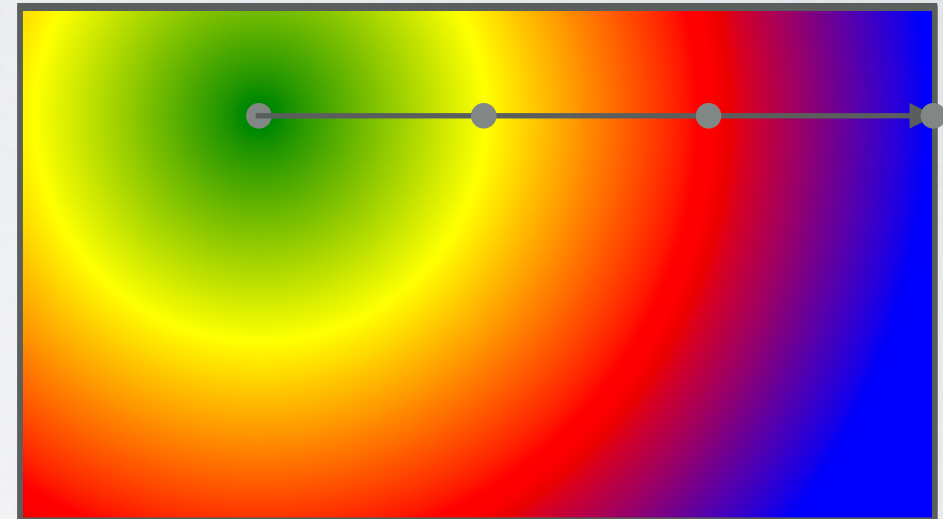


```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

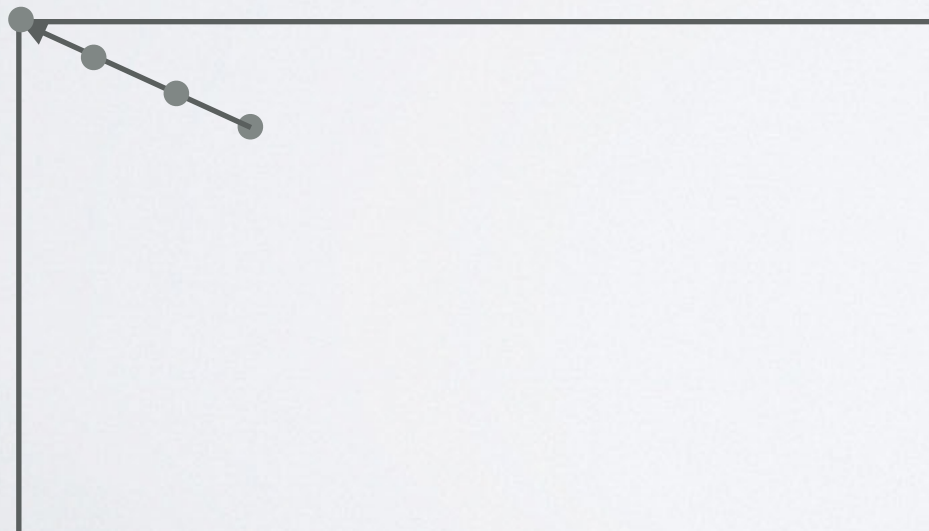
HOW WILL THE OTHERS LOOK LIKE?



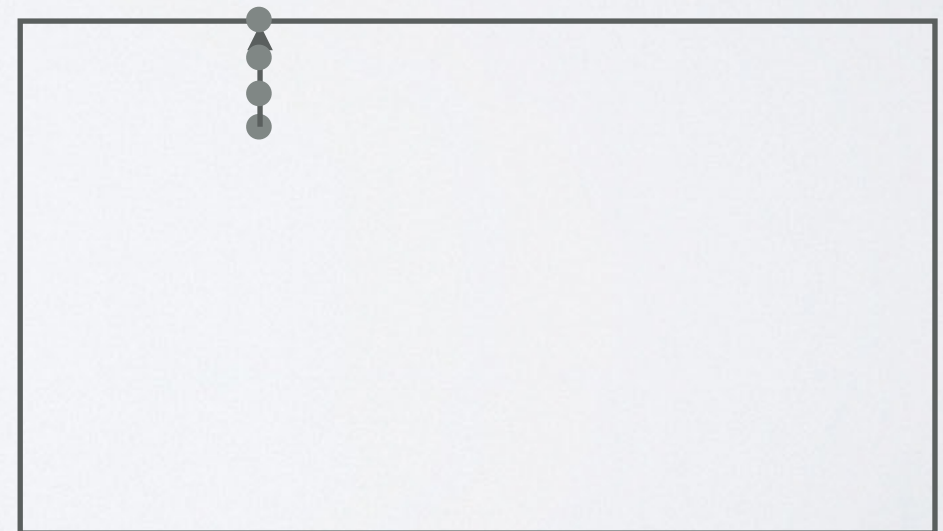
radial-gradient(circle **farthest-corner**
at 130px 60px,
green, yellow, red, blue);



radial-gradient(circle **farthest-side**
at 130px 60px,
green, yellow, red, blue);

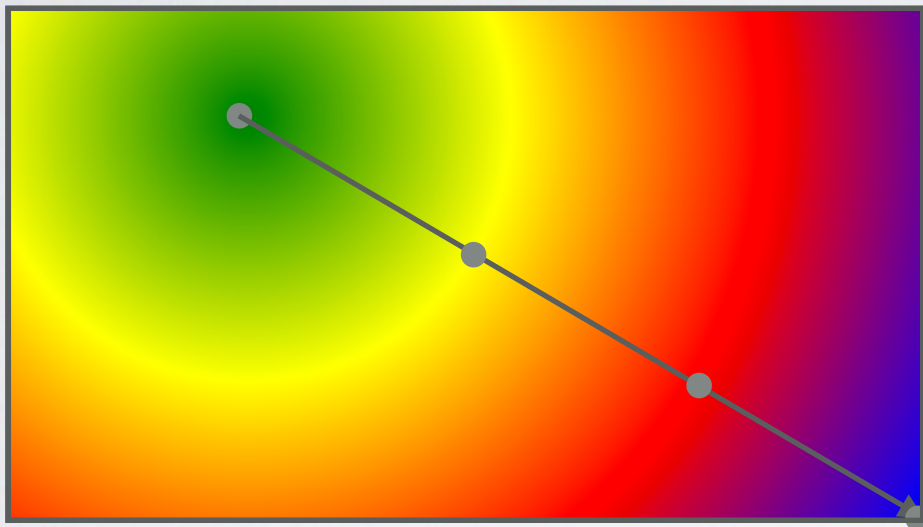


radial-gradient(circle **closest-corner**
at 130px 60px,
green, yellow, red, blue);

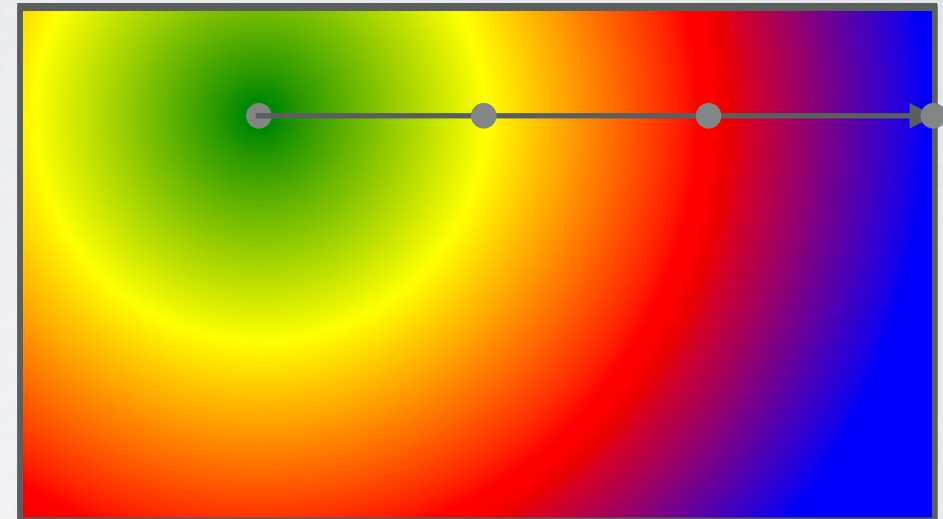


radial-gradient(circle **closest-side**
at 130px 60px,
green, yellow, red, blue);

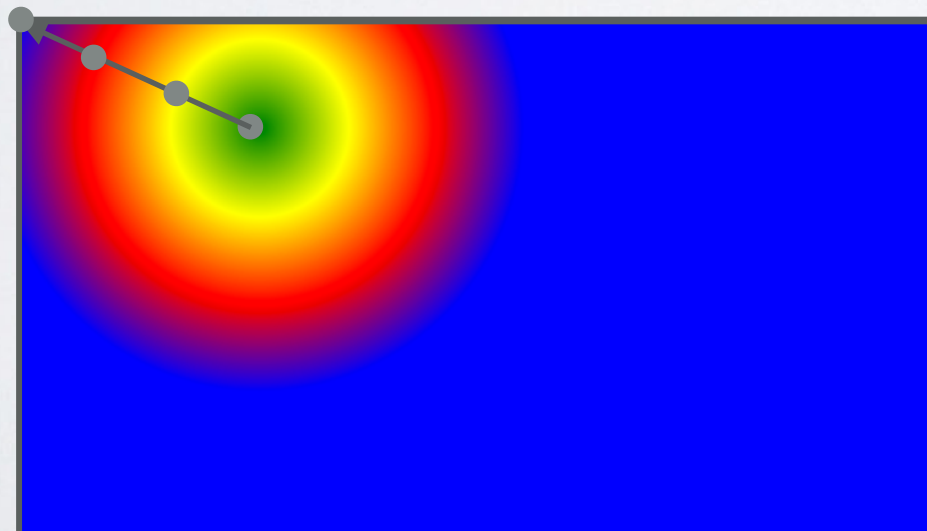
HOW WILL THE OTHERS LOOK LIKE?



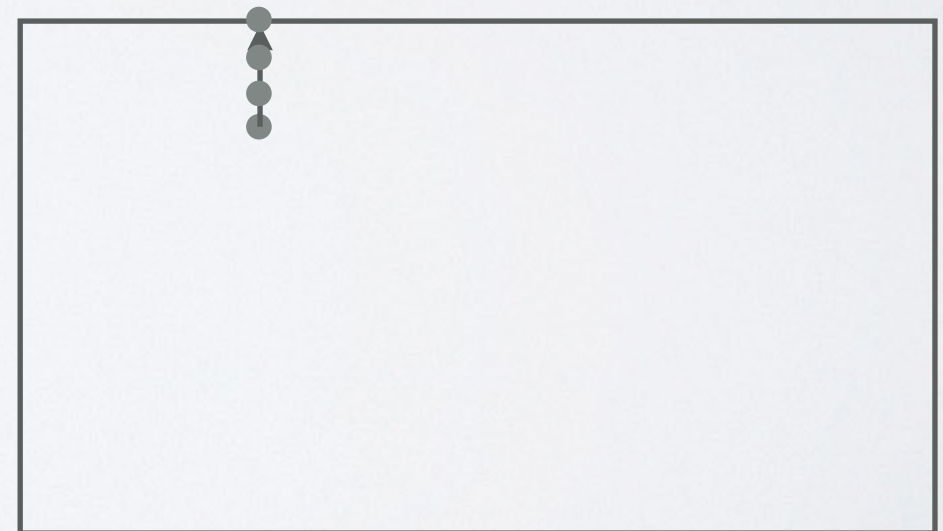
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

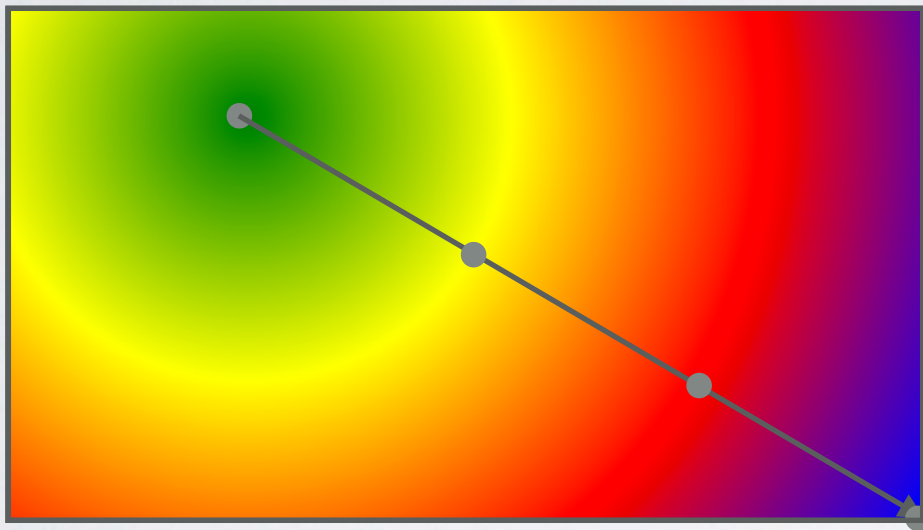


```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```

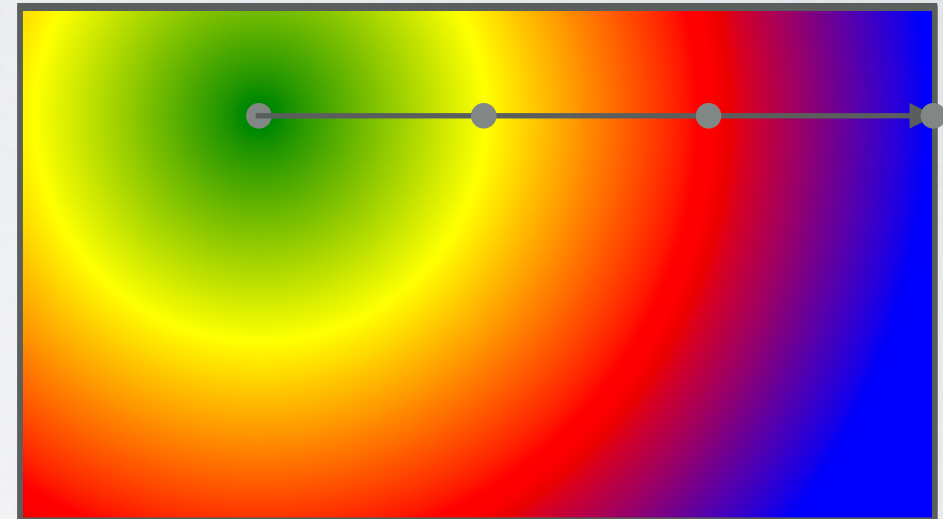


```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```

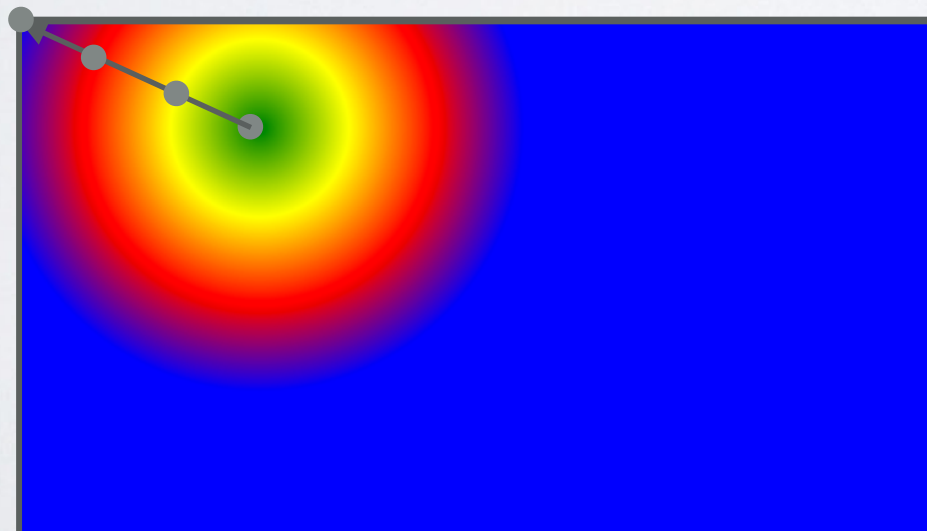
HOW WILL THE OTHERS LOOK LIKE?



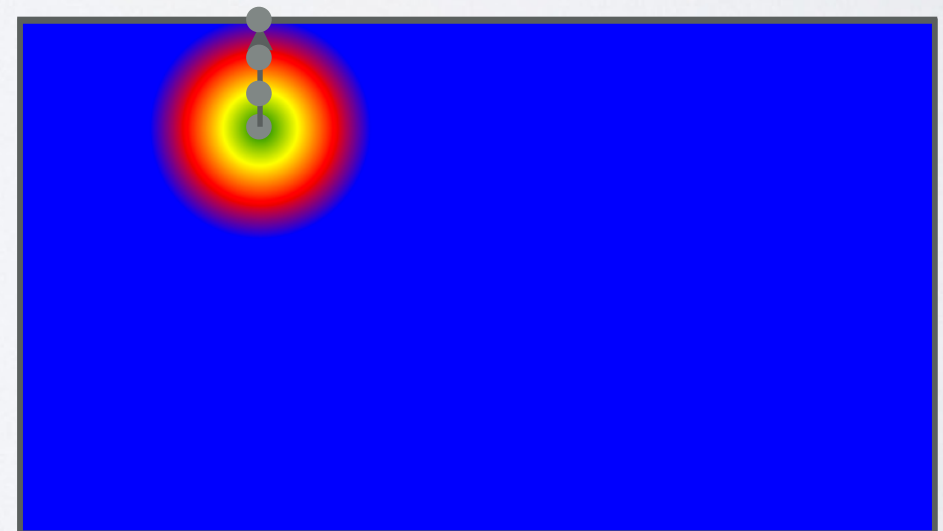
```
radial-gradient(circle farthest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle farthest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle closest-corner  
  at 130px 60px,  
  green, yellow, red, blue);
```



```
radial-gradient(circle closest-side  
  at 130px 60px,  
  green, yellow, red, blue);
```


ROUNDED CORNERS

Rounded Corners

ROUNDED CORNERS

```
div {  
  padding: 10px 40px;  
  background: #dddddd;  
  width: 300px;
```

Rounded Corners

ROUNDED CORNERS

```
div {  
  padding: 10px 40px;  
  background: #dddddd;  
  width: 300px;
```



ROUNDED CORNERS

```
div {  
    padding: 10px 40px;  
    background: #dddddd;  
    width: 300px;  
    border-radius: 15px;  
}
```



Rounded Corners

SHADOWS



SHADOWS

```
#element {
```

```
    box-shadow: 10px 10px 5px #888888
```

```
}
```



SHADOWS

Horizontal
offset

```
#element {
```

```
  box-shadow: 10px 10px 5px #888888
```

```
}
```



SHADOWS

Horizontal
offset

Vertical offset

```
#element {
```

```
  box-shadow: 10px 10px 5px #888888
```

```
}
```



SHADOWS

```
#element {
```

```
  box-shadow: 10px 10px 5px #888888
```

Horizontal
offset

Vertical offset

Blur distance

```
}
```



SHADOWS

```
#element {
```

```
  box-shadow: 10px 10px 5px #888888
```

Horizontal
offset

Vertical offset

Blur distance

```
}
```

Shadow
color



DEMO

- Linear gradient as background
- Different directions and color stops
- Move color stops on gradient line
- A rounded box with a border and a shadow
- Radial gradient
- Play around with sizes, moving origins, etc.

TRANSITIONS, ANIMATIONS, AND TRANSFORMATIONS

PURPOSE OF TRANSITIONS IN CSS

PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes

PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes
 - Change should not happen immediately

PURPOSE OF TRANSITIONS IN CSS

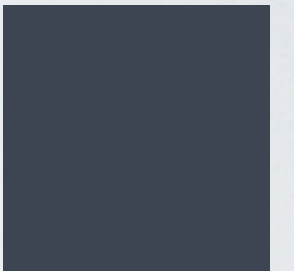
- If a value of a CSS property of an element changes
 - Change should not happen immediately
 - Rather it should happen over a specified duration

PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes
 - Change should not happen immediately
 - Rather it should happen over a specified duration
 - Change should be animated

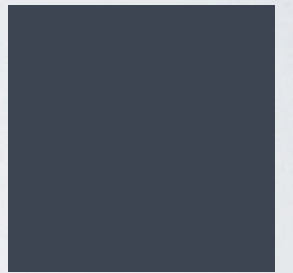
PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes
 - Change should not happen immediately
 - Rather it should happen over a specified duration
 - Change should be animated



PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes
 - Change should not happen immediately
 - Rather it should happen over a specified duration
 - Change should be animated



Mind the
change of color

PURPOSE OF TRANSITIONS IN CSS

- If a value of a CSS property of an element changes
 - Change should not happen immediately
 - Rather it should happen over a specified duration
 - Change should be animated




Mind the
change of color

WHAT TO SPECIFY IN A TRANSITION

- Property
- Duration
- Timing function

WHAT TO SPECIFY IN A TRANSITION



width, height,
color, ...

- Property
- Duration
- Timing function

WHAT TO SPECIFY IN A TRANSITION

width, height,
color, ...

- Property
- Duration
- Timing function

see <https://www.w3.org/TR/css3-transitions/> for a full list

WHAT TO SPECIFY IN A TRANSITION

- Property
 - width, height, color, ...
- Duration
 - see <https://www.w3.org/TR/css3-transitions/> for a full list
- Timing function
 - s ... seconds
ms ... milliseconds

WHAT TO SPECIFY IN A TRANSITION

- Property

width, height,
color, ...

- Duration

see <https://www.w3.org/TR/css3-transitions/> for a full list

- Timing function

s ... seconds
ms ... milliseconds

ease, linear, ease-in,
ease-out, ease-in-out, ...

EXAMPLE OF A TRANSITION

```
div {  
  width:100px;  
  height:100px;  
  background-color:red;  
  transition:width 2s;  
}
```

```
div:hover {  
  width:200px;  
}
```


EXAMPLE OF A TRANSITION

```
div {  
  width:100px;  
  height:100px;  
  background-color:red;  
  transition:width 2s;  
}
```

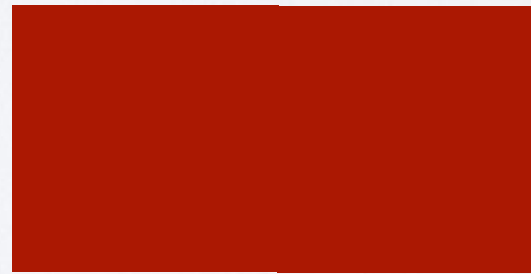
```
div:hover {  
  width:200px;  
}
```



EXAMPLE OF A TRANSITION

```
div {  
  width:100px;  
  height:100px;  
  background-color:red;  
  transition:width 2s;  
}
```

```
div:hover {  
  width:200px;  
}
```



PURPOSE OF CSS 3 ANIMATIONS

- Sometimes more control over the animation of a transition is needed
 - Intermediate color values are required
 - The animation should start-over after having finished
 - Bouncing effects
 - ...

EXAMPLE OF AN ANIMATION

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: rainbow;  
    animation-duration: 5s;  
    animation-iteration-count: infinite;  
}
```

EXAMPLE OF AN ANIMATION

```
div {  
    width: 100px;  
    height: 100px;  
    background-color: red;  
    animation-name: rainbow;  
    animation-duration: 5s;  
    animation-iteration-count: infinite;  
}
```

Refers to keyframes
which specify exact
behavior of animation

CSS3 ANIMATIONS

```
@keyframes rainbow {  
    0%      { background-color: red; }  
    50%     { background-color: yellow; }  
    75%     { background-color: blue; }  
    100%    { background-color: green; }  
}
```


PURPOSE OF CSS3 TRANSFORMATIONS



Text

PURPOSE OF CSS3 TRANSFORMATIONS

- Scale



PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

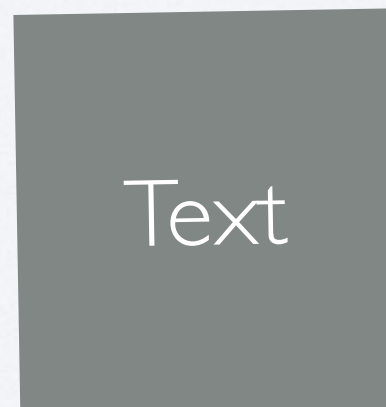


PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate



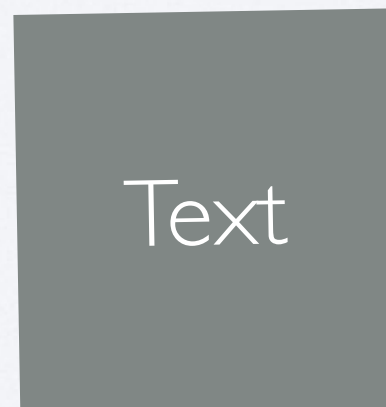
PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate

```
.element {  
  transform: rotate(360deg);  
}
```



PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate

```
.element {  
  transform: rotate(360deg);  
}
```

- Translate



Text

PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate

```
.element {  
  transform: rotate(360deg);  
}
```

- Translate

```
.element {  
  transform: translate(20px, 0px);  
}
```



Text

PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate

```
.element {  
  transform: rotate(360deg);  
}
```

- Translate

```
.element {  
  transform: translate(20px, 0px);  
}
```

- Skew



Text

PURPOSE OF CSS3 TRANSFORMATIONS

- Scale

```
.element {  
  width: 20px;  
  height: 20px;  
  transform: scale(20);  
}
```

- Rotate

```
.element {  
  transform: rotate(360deg);  
}
```

- Translate

```
.element {  
  transform: translate(20px, 0px);  
}
```

- Skew

Watch out for
it during demo

Text

BROWSER SPECIFIC STUFF

BROWSER SPECIFIC STUFF

- Internet Explorer < 10 doesn't support transitions, animations, or transformations

BROWSER SPECIFIC STUFF

- Internet Explorer < 10 doesn't support transitions, animations, or transformations
- For animations and transformations the following has to be taken care of
 - Firefox needs the prefix -moz
 - Chrome and Safari need the prefix -webkit
 - Opera needs the prefix -o

BROWSER SPECIFIC STUFF

- Internet Explorer < 10 doesn't support transitions, animations, or transformations
- For animations and transformations the following has to be taken care of
 - Firefox needs the prefix -moz
 - Chrome and Safari need the prefix -webkit
 - Opera needs the prefix -o



Example see
the next slides

BROWSER SPECIFIC STUFF

- Internet Explorer < 10 doesn't support transitions, animations, or transformations
- For animations and transformations the following has to be taken care of
 - Firefox needs the prefix -moz
 - Chrome and Safari need the prefix -webkit
 - Opera needs the prefix -o
- Transitions work without the above prefixes



Example see
the next slides

ANIMATION WITH PREFIXES

(I)

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-animation: rainbow 5s;  
    -moz-animation: rainbow 5s;  
    -o-animation: rainbow 5s;  
    animation: rainbow 5s;  
}
```


ANIMATION WITH PREFIXES

(2)

@-webkit-keyframes rainbow {	@keyframes rainbow {
0% {background:red;}	0% {background:red;}
50% {background:yellow;}	25% {background:yellow;}
75% {background:blue;}	50% {background:blue;}
100% {background:green;}	100% {background:green;}
}	}

ANIMATION WITH PREFIXES

(2)

-moz, -o, etc.
analogously

```
@-webkit-keyframes rainbow {
```

```
  0% {background:red;}
```

```
  50% {background:yellow;}
```

```
  75% {background:blue;}
```

```
  100% {background:green;}
```

```
}
```

```
@keyframes rainbow {
```

```
  0% {background:red;}
```

```
  25% {background:yellow;}
```

```
  50% {background:blue;}
```

```
  100% {background:green;}
```

```
}
```


DEMO

- Transition
 - Resize
- Animation
 - Rainbow-Colors
- Freemover (needs absolute positioning comes at end)
- Transformation
 - Concentrate on skew.

POSITIONING

POSITIONING

POSITIONING

- All html elements are positioned within the text flow

POSITIONING

- All html elements are positioned within the text flow
- One exception: floating elements

POSITIONING

- All html elements are positioned within the text flow
- One exception: floating elements
- Sometimes you want to place an element on a specific location
 - Explicit positioning can help here

DIFFERENT WAYS OF POSITIONING

DIFFERENT WAYS OF POSITIONING

- Position static
 - Default, element appears in the document flow

DIFFERENT WAYS OF POSITIONING

- Position static
 - Default, element appears in the document flow
- Position fixed
 - Element position is relative to the browser window

DIFFERENT WAYS OF POSITIONING

- Position static
 - Default, element appears in the document flow
- Position relative
 - Element position is relative to its normal position
- Position fixed
 - Element position is relative to the browser window

DIFFERENT WAYS OF POSITIONING

- Position static
 - Default, element appears in the document flow
- Position relative
 - Element position is relative to its normal position
- Position absolute
 - Element position is relative to its first parent element which position is not static
- Position fixed
 - Element position is relative to the browser window

TRY AND TAKE CARE

- http://www.w3schools.com/cssref/playit.asp?filename=playcss_position
- Absolute position has to be used with care
 - Can easily break flexibility to adjust to different screen sizes


```
#titanic {  
  position: absolute;  
  bottom: 0;  
}
```

```
.iceberg {  
  display: block;  
}
```

