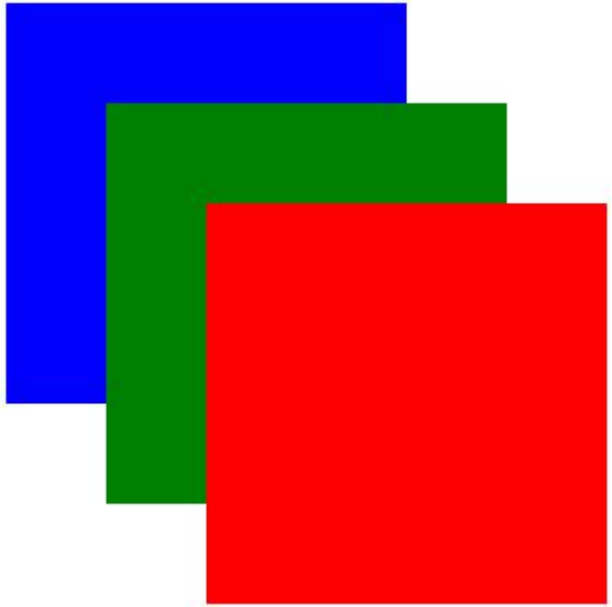
What?!?! Paint and Composite Gotchas

Chris Harrelson
and paint-dev@chromium.org

Paint order

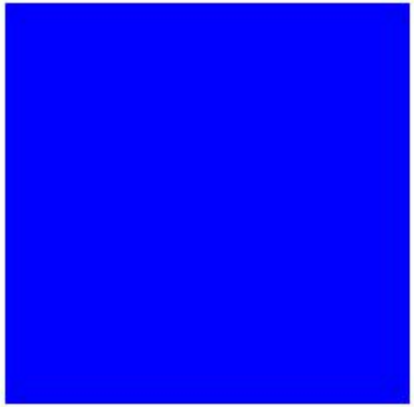


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }
</style>

<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

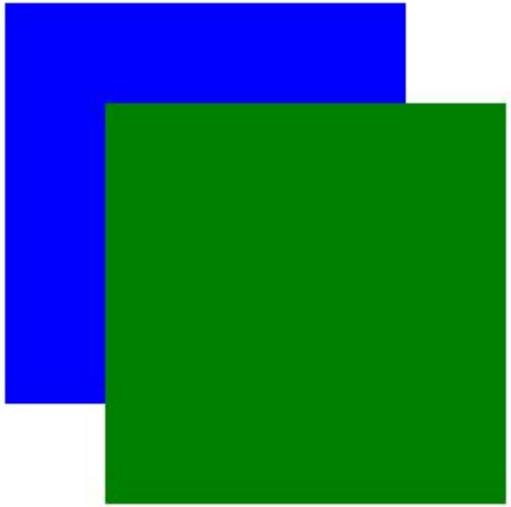
Paint order



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }
</style>

<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

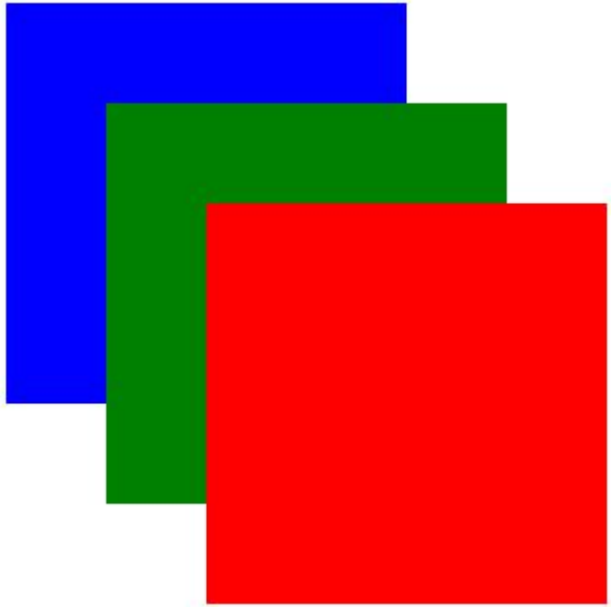
Paint order



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }
</style>

<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

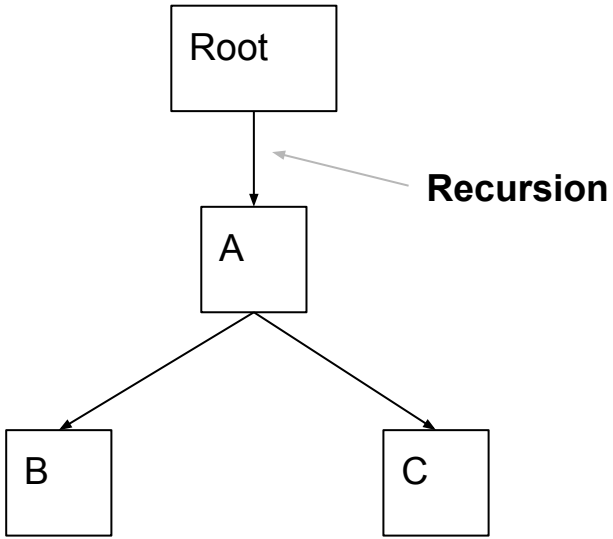
Paint order



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }
</style>

<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

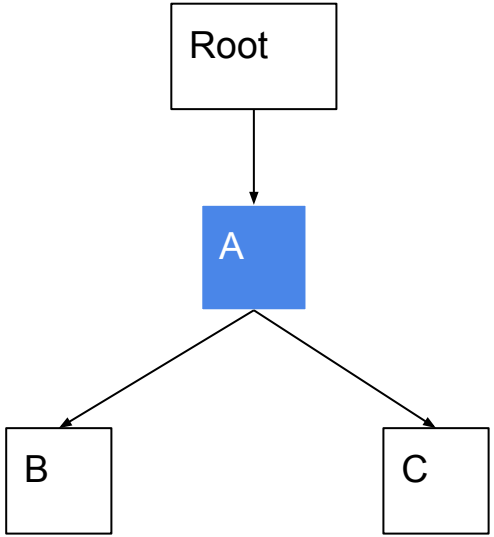
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

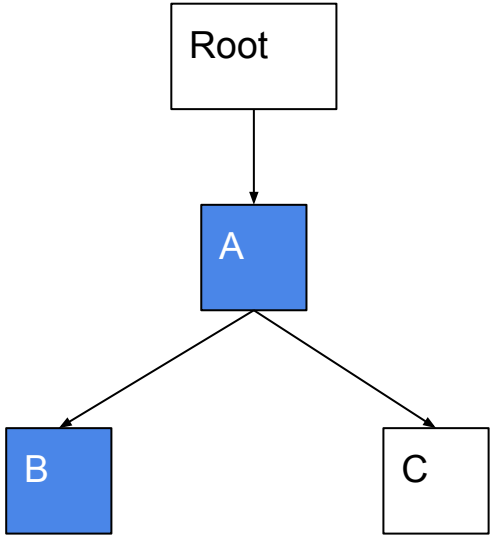
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

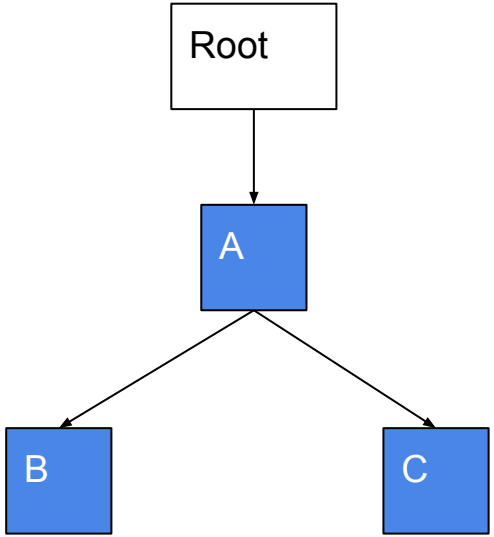
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

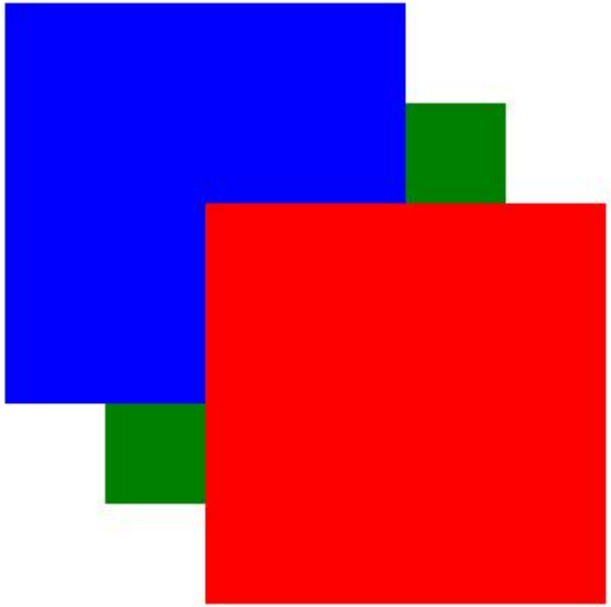
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

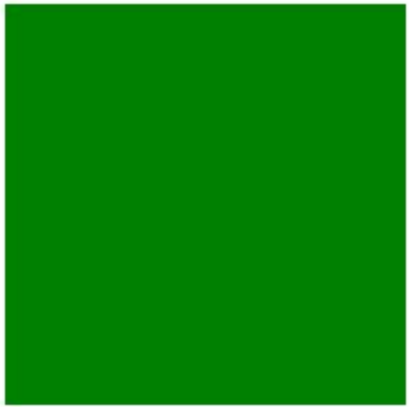
Paint order: z-index



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

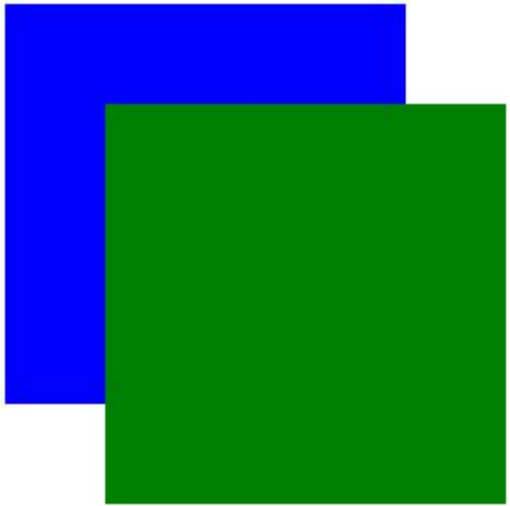
Paint order: z-index



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

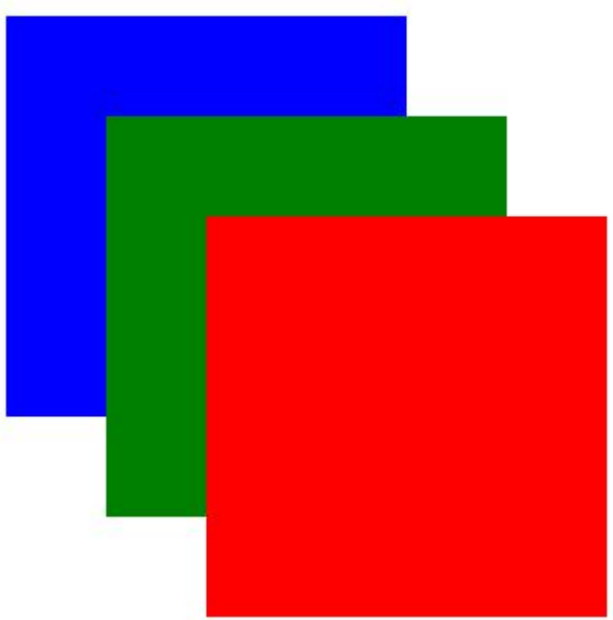
Paint order: z-index



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

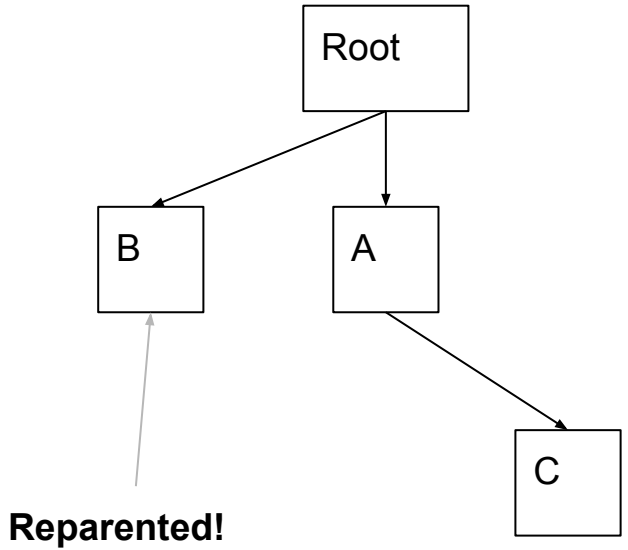
Paint order: z-index



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

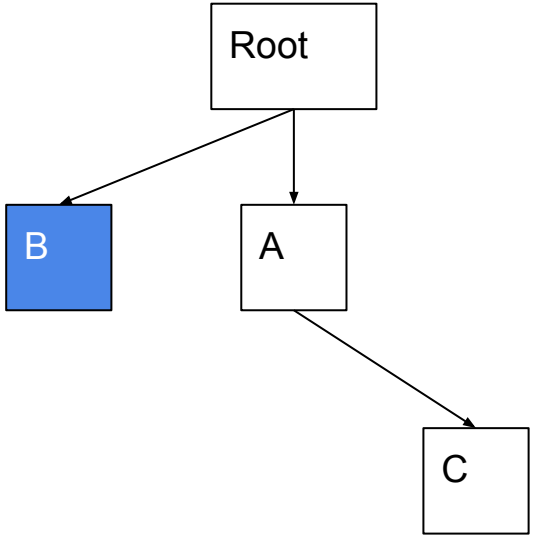
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

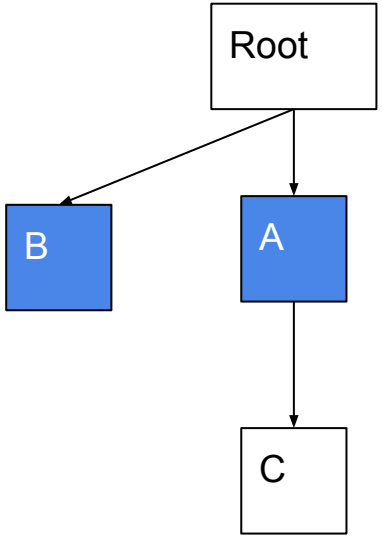
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

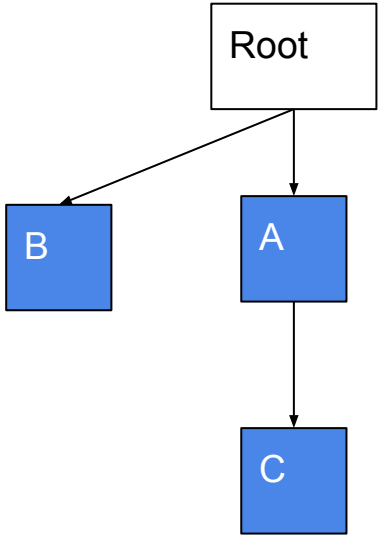
View from painting code




```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

View from painting code



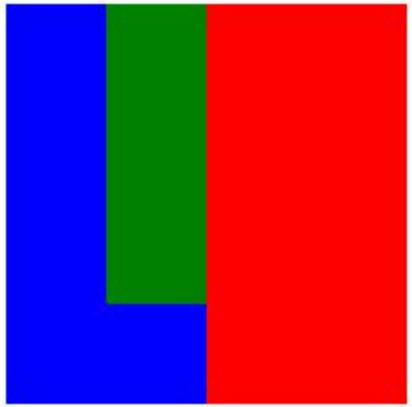

```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A { background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B" style=”z-index: -1”></div>
  <div id="C"></div>
</div>
```

Observations

- **The Paint tree is not the same topology as the DOM/Layout tree**

Overflow clip

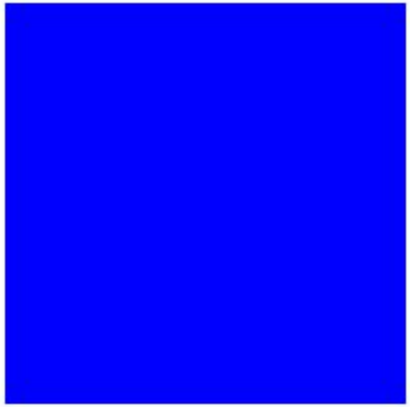


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/JYeYwL

Overflow clip

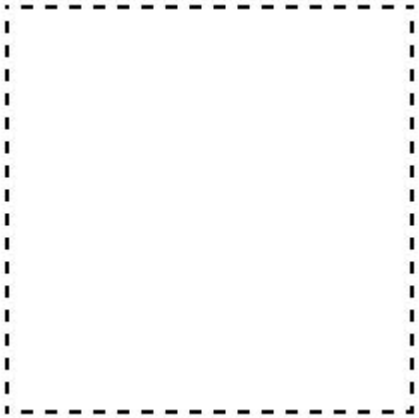


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip

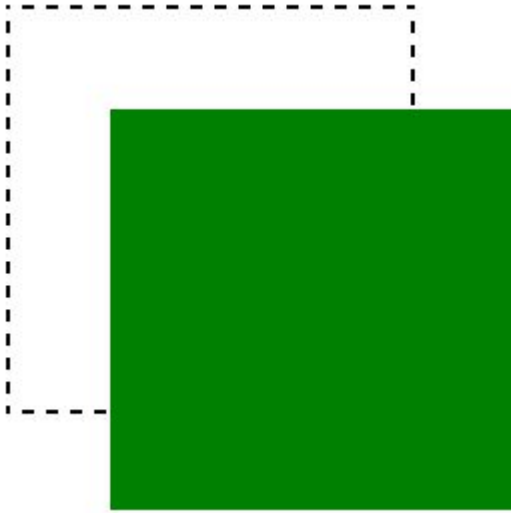


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip

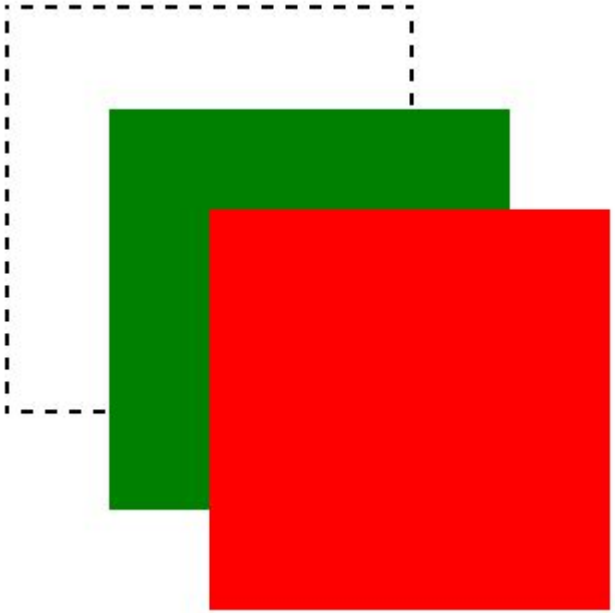


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip

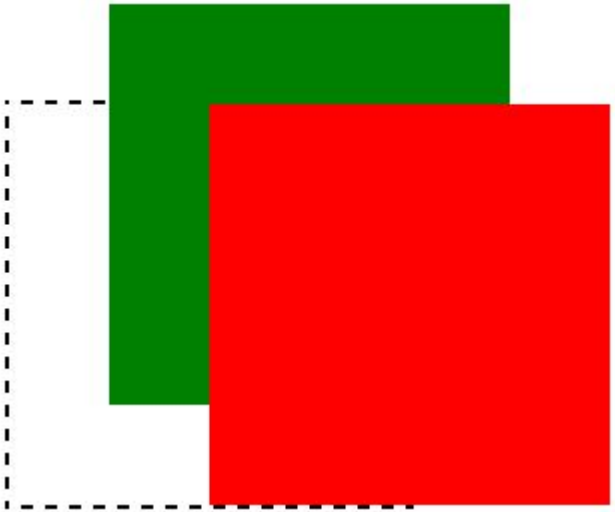


```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip



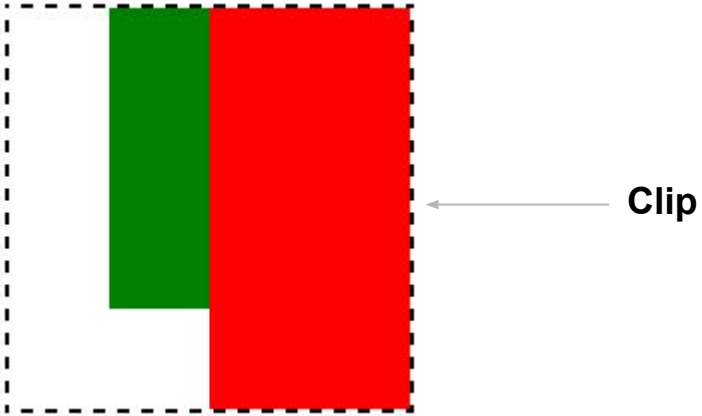
Scroll

```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip

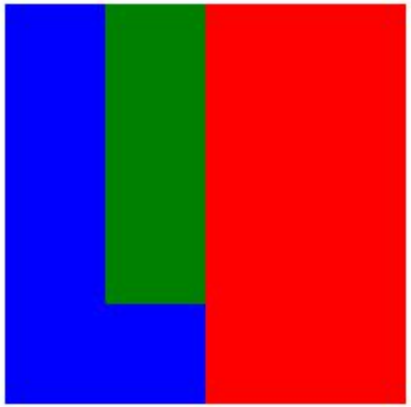



```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Overflow clip



Overlay

```
<style>
div {
  position:relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll;  background: blue;}
#B { background: green; top: 25px; left: 25px; }
#C {background: red;  top: -50px; left: 50px; }

</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Containing Block

```
<div id=”A”>
  <div id=”B”>
    <div id=”C”></div>
  </div>
<div>
```

The containing block of C is B. The containing block of B is A.

Containing Block

```
<div id=”A”>
  <div id=”B”>
    <div id=”C” style=”position: relative”></div>
  </div>
<div>
```

Same as the last example: The containing block of C is B. The containing block of B is A.

Containing Block

The containing block of a position:absolute is the container with non-default “position:”.

```
<div id=”A” style=”position: relative”>
  <div id=”B”>
    <div id=”C” style=”position: absolute”></div>
  </div>
<div>
```

The containing block of C is A.

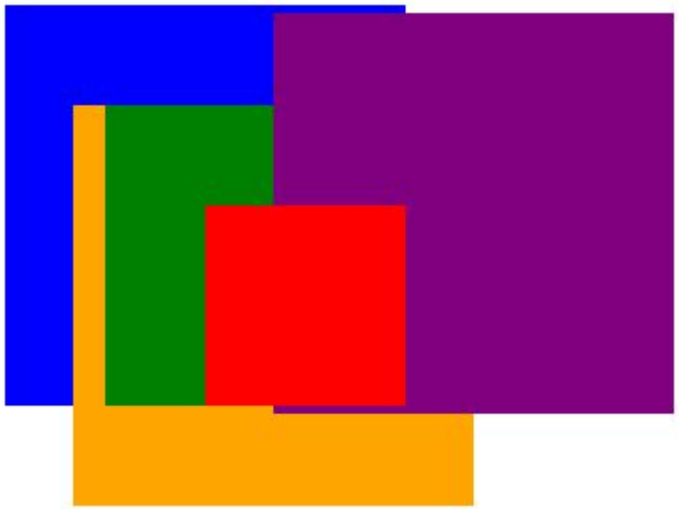
Containing Block

The containing block of a position: fixed element is the root of the frame (or transformed or SVG foreign element).

```
<div id=”A” style=”position: relative”>
  <div id=”B”>
    <div id=”C” style=”position: fixed”></div>
  </div>
<div>
```

The containing block of C is the root HTML element.

Containing block scroll

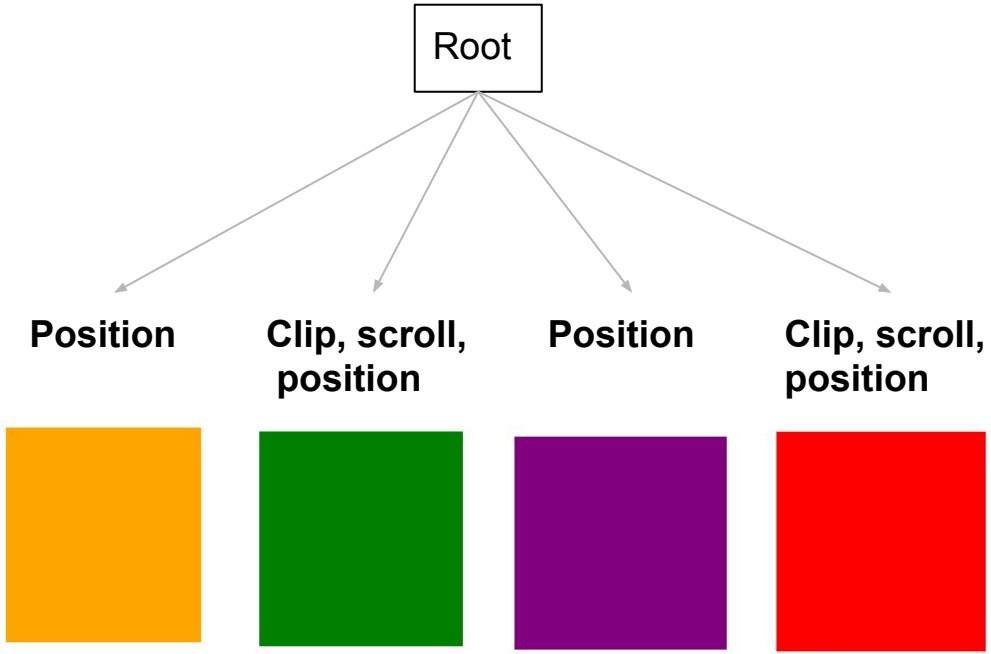


http://codepen.io/chrishtr/pen/jbQmLv

```
div {
  width: 100px;
  height: 100px;
}
#A {overflow:scroll; background: blue}
#B {position: relative; visibility: visible; background:
green; top: 25px; left: 25px; }
#C {position: relative; visibility: visible; background:
red;  top: -50px; left: 50px; }
#D { position: absolute; top: 33px; left: 25px;
background: orange;}
#E { position: fixed; top: 10px; left: 75px;
background: purple;}
</style>

<div id="A">
  <div id="D"></div>
  <div id="B"></div>
  <div id="E"></div>
  <div id="C"></div>
</div>
```

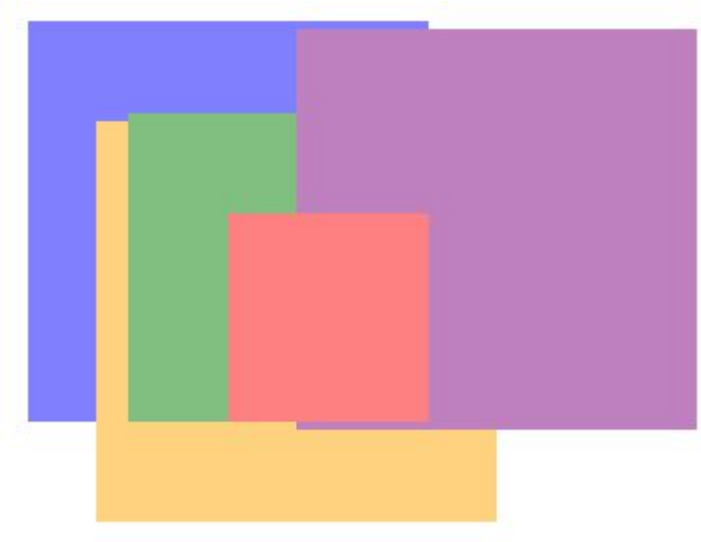
Containing block scroll




```
div {
  width: 100px;
  height: 100px;
}
#A {overflow:scroll; background: blue}
#B {position: relative;background: green; top: 25px; left: 25px; }
#C {position: relative; background: red;  top: -50px; left: 50px; }
#D { position: absolute; top: 33px; left: 25px; background: orange;}
#E { position: fixed; top: 10px; left: 75px; background: purple;}
</style>

<div id="A">
  <div id="D"></div>
  <div id="B"></div>
  <div id="E"></div>
  <div id="C"></div>
</div>
```

Containing block scroll



Adding a stacking context doesn't help. :(

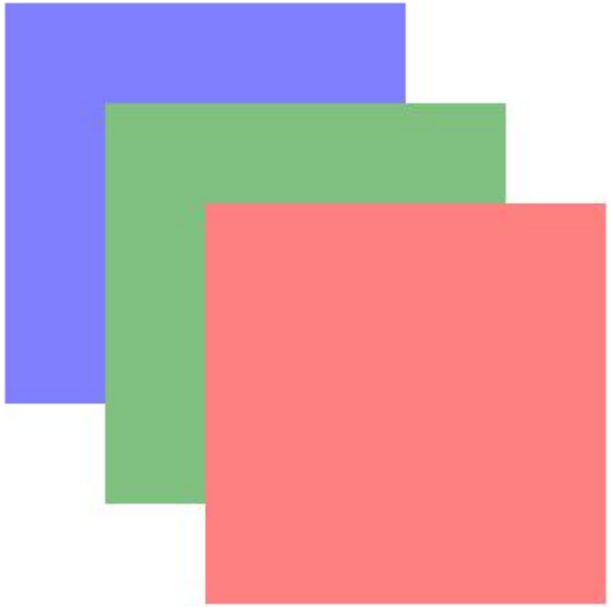
```
div {
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; overflow:scroll; background:
blue}
#B {position: relative; visibility: visible; background:
green; top: 25px; left: 25px; }
#C {position: relative; visibility: visible; background:
red;  top: -50px; left: 50px; }
#D { position: absolute; top: 33px; left: 25px;
background: orange;}
#E { position: fixed; top: 10px; left: 75px;
background: purple;}
</style>

<div id="A">
  <div id="D"></div>
  <div id="B"></div>
  <div id="E"></div>
  <div id="C"></div>
</div>
```

Observations

- The Paint tree is not the same topology as the DOM/Layout tree
- **Clipping and scrolling and containing block make for weird complexity**
- **Stacking contexts don’t fix it**

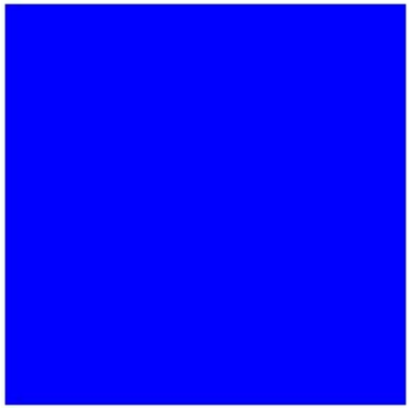
Opacity



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

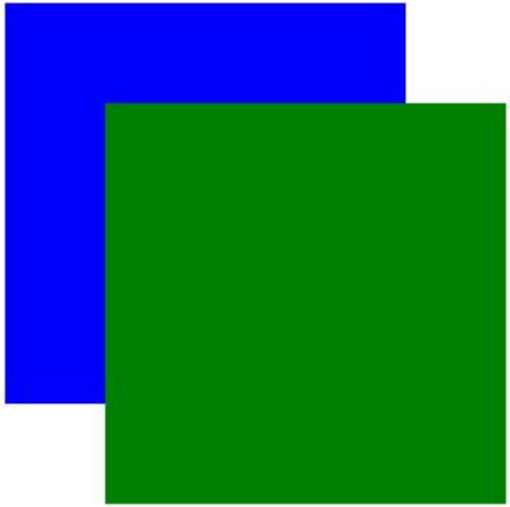
Opacity



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

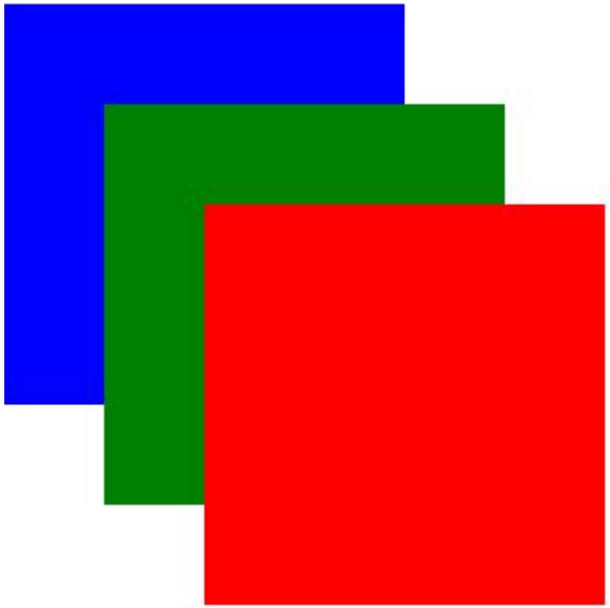
Opacity



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

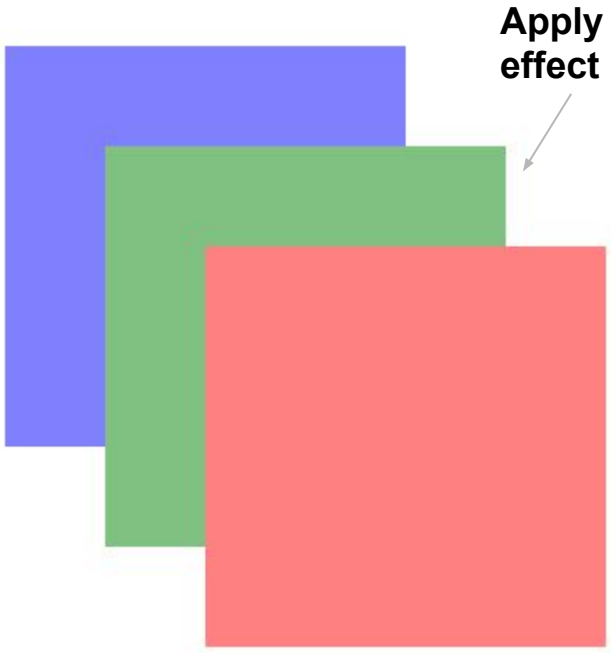
Opacity



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Opacity



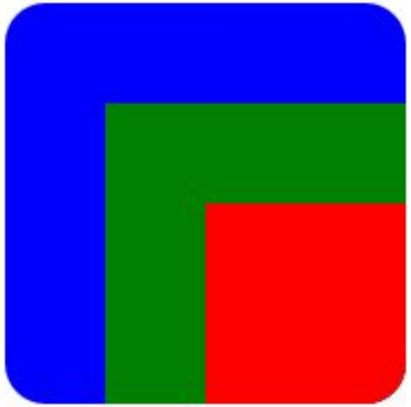

```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Observations

- The Paint tree is not the same topology as the DOM/Layout tree
- Clipping and scrolling and containing block make for weird complexity
- Stacking contexts don’t fix it
- **Some operations apply to groups of elements:**
  - **Clips**
  - **Scrolling**
  - **Opacity**
- **Caching opportunities:**
  - **Reuse picture on scroll change**
  - **Reuse picture on opacity change**

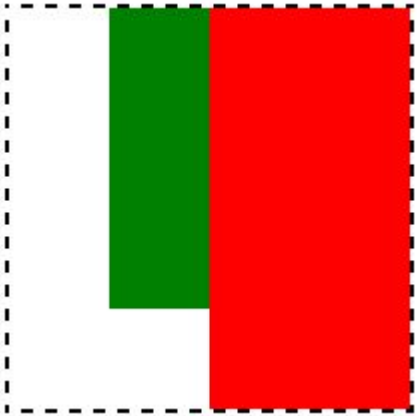
Rounded corners



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 20px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

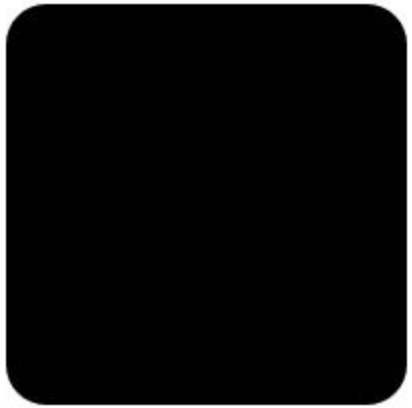
Rounded corners



```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 2px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Rounded corners



Mask

```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 2px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Rounded corners

Clip to mask

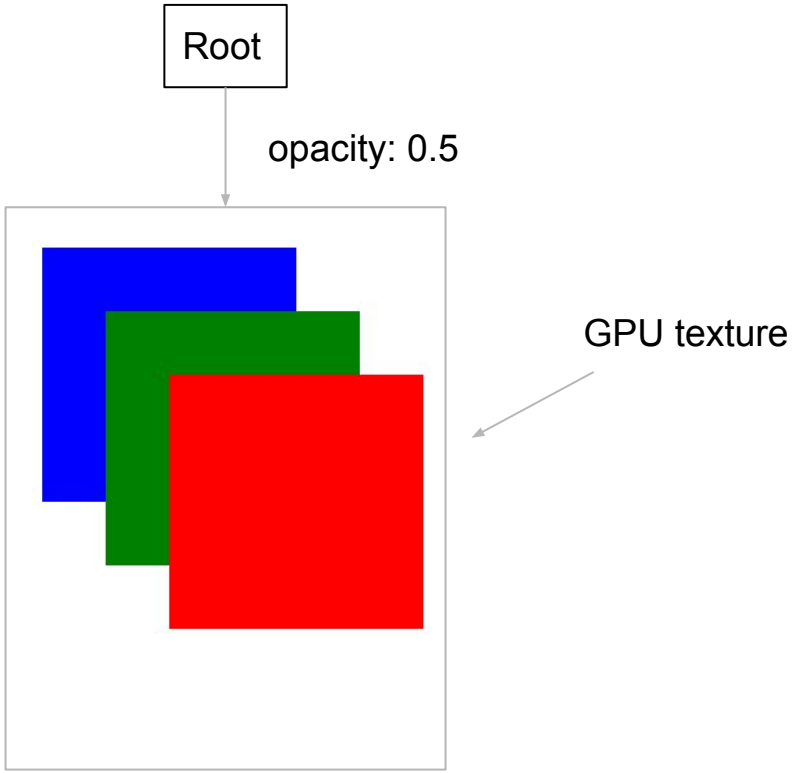
```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 2px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Observations

- The Paint tree is not the same topology as the DOM/Layout tree
- Clipping and scrolling and containing block make for weird complexity
- Stacking contexts don’t fix it
- Some operations apply to groups of elements:
  - Clips
  - Scrolling
  - Opacity
- Caching opportunities:
  - Reuse picture on scroll change
  - Reuse picture on opacity change
- **Rounded corners require an extra mask bitmap**

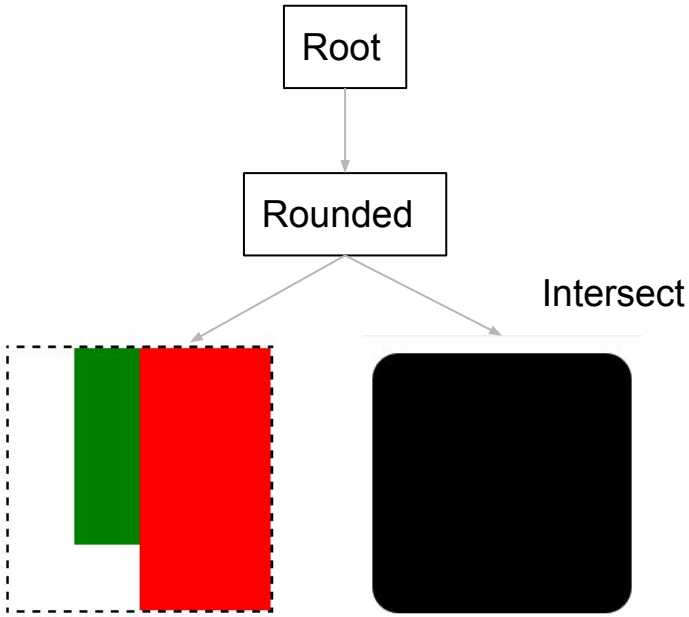
Compositing: Opacity




```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {opacity: 0.5; background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

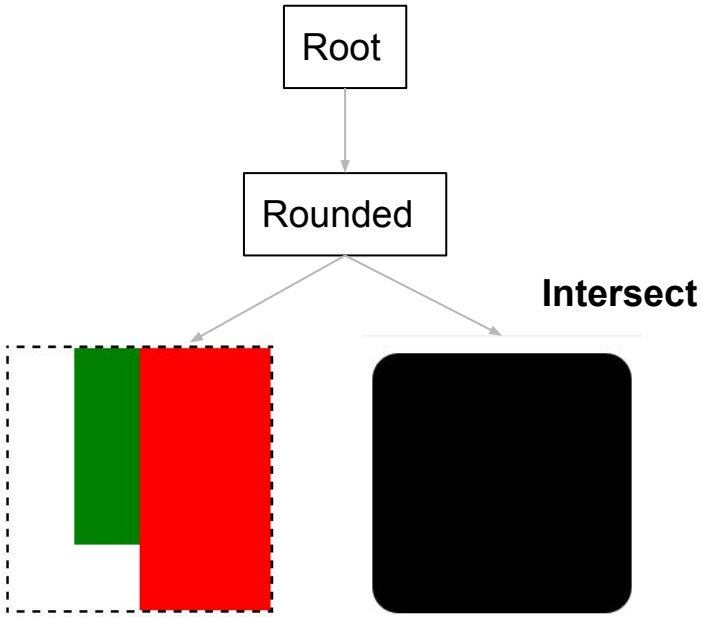
Compositing: Rounded corners




```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 20px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

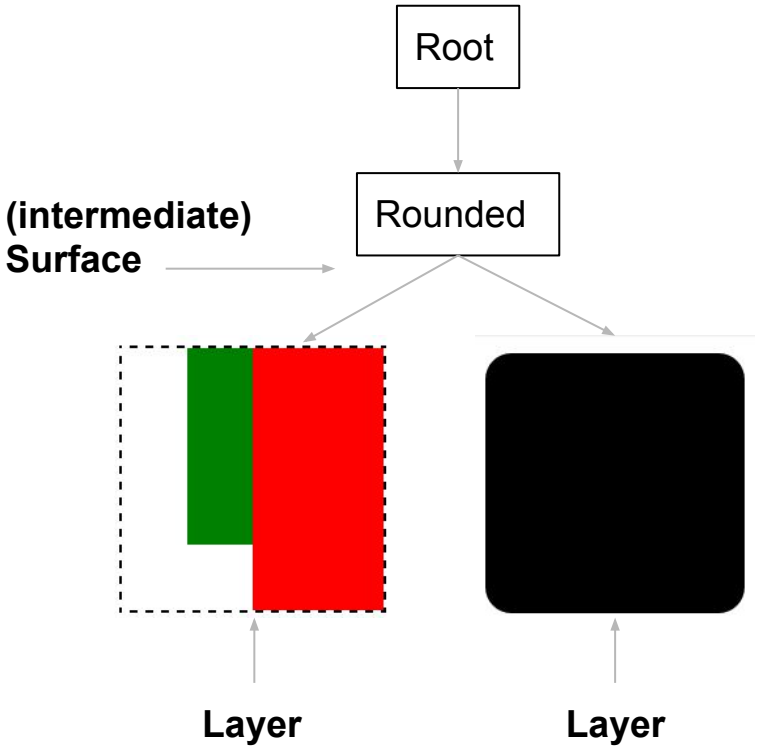
Compositing: Rounded corners




```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 20px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Compositing: Rounded corners




```
<style>
div {
  position: relative;
  width: 100px;
  height: 100px;
}
#A {overflow: scroll; border-radius: 20px;
background: blue}
#B { visibility: visible; background: green; top: 25px;
left: 25px; }
#C { visibility: visible; background: red;  top: -50px;
left: 50px; }
</style>
<div id="A">
  <div id="B"></div>
  <div id="C"></div>
</div>
```

http://codepen.io/chrishtr/pen/YyRyQv

Observations

- The Paint tree is not the same topology as the DOM/Layout tree
- Clipping and scrolling and containing block make for weird complexity
- Stacking contexts don’t fix it
- Some operations apply to groups of elements:
  - Clips
  - Scrolling
  - Opacity
- Caching opportunities:
  - Reuse picture on scroll change
  - Reuse picture on opacity change
- Rounded corners require an extra mask bitmap **and intersection output**
- **Compositing is about the same as regular painting, except on the GPU**

The pipeline

1. Compositing: deciding which pieces of content to cache into bitmaps (textures)
    a. Output: list of **composited layers**, and **mapping of content to layers**

The pipeline

1. Compositing: deciding which pieces of content to cache into bitmaps (textures)
2. Paint invalidation

The pipeline

1. Compositing
2. Paint invalidation
3. Paint: generating **low-level draw commands** for each LayoutObject
    a. Output: a display list of these commands

The pipeline

1. Compositing
2. Paint invalidation
3. Paint
4. Tile rastering: breaking composited layers down and drawing their textures
    a. Output: set of **textures for each tile** of each composited layer

The pipeline

1. Compositing
2. Paint invalidation
3. Paint
4. Tile rastering: breaking composited layers down and drawing their textures
5. Generation of surfaces / render passes / draw quads: making the box-arrow diagrams shown in earlier slides
    a. Output: **list of quads**

The pipeline

1. Compositing
2. Paint invalidation
3. Paint
4. Tile rastering: breaking composited layers down and drawing their textures
5. Generation of surfaces / render passes / draw quads
6. Drawing quads to the screen (in the browser process / ubercompositor)

Hard examples

Clipping under a scrolling element, with conflicting containing blocks.

http://jsfiddle.net/bgfzL0c3/1/

Blink does not paint or

composite this correctly

```
<style>
#scroller { position: relative; }
#clipper { position: absolute; clip: rect(...); }
#fixed { position: fixed }
</style>
<div id="scroller">
    <div id="clipper">
        <div id="fixed"></div>
    </div>
</div>
```

Problems

- Even if an element scrolls, **it can be clipped by elements that do scroll**

Hard examples

border-radius with descendant that has non-1 opacity, where some escape the clip

http://jsbin.com/cawucipawa/edit?html,css,output

```
<div id="clip">
    <div id="opacity">
      <div id="fixed"></div>
      <div id="abs"></div>
    </div>
 </div>
```

Problems

- Even if an element scrolls, it can be affected by elements that do scroll
- We are **forced to apply opacity separately** in order to apply clips correctly
  - Non-atomic opacity leads to rendering issues when content overlaps, because the opacities incorrectly add up in the overlapped region.

Hard examples

overflow clip with descendants that have a blur, where some escape the clip

http://jsbin.com/woroyoliqe/1/edit?html,css,output

```
<style>
#clip { position: absolute; overflow: hidden; }
#blur { -webkit-filter: blur(5px); }
#fixed { position: fixed; }
#abs { position: absolute; }
<div id="clip">
    <div id="blur">
      <div id="fixed"></div>
      <div id="abs"></div>
    </div>
  </div>
```

Problems

- Even if an element scrolls, it can be affected by elements that do scroll
- We are forced to apply opacity separately in order to apply clips correctly
- Same goes for blur (!!!!)

Thanks