# The Mondrain Project

The Mondrain Project is a fun way to create a work of Art with HTML and CSS!

## Step 1: Paint your gallery wall.



Before we do anything else let's give ourselves a nice big wall to hang our painting.

- Go into the CSS tab in your Mondrian project.
- Erase anything that's already there, we won't be needing that pagecontent style.
- Set the **body** and **html** styles to have a drywall-like background-color of **#f6f6f6**. A quick way to give two elements the same property is to write a style with multiple selectors separated by a comma.

*Our CSS code looks like this:*

```
body, html {
   background-color: #f6f6f6;
}
```

# Step 2: Hang your canvas



Let’s make a big box. This will be the canvas for our code painting and the container we will put everything else we will make into. To make our painting we will be using two new concepts: **#id** and ** **.

- In the **CSS** panel make a new #id for your big box. A CSS ID is type of selector that can be used to specify a style for a single, unique element in the HTML. You can call your ID whatever you like you just need to add an # in front of it. We’ve called ours **#painting**.
- Set the **height** and **width** attributes for your box. Keep in mind that we will dividing this box into sections using simple arithmetic so pick and number that is big but also easy to divide into quarters and thirds, etc. We made ours **400px**.
- Set the background color to white. You can also use a more canvas-like hex value. We used **#fff4db**.

- Now let's add a little positioning to this style. We want our painting to be centered on the wall with a little space from the ceiling. One easy way to do that is to set the left and right margins to **auto** and margin-top to **30px**.
- We also added a **box-shadow** to our **#painting** style to help it stand out from the wall.

*The new CSS code we added looks like this:*

```
#painting {
  background-color: #fff4db;
  width: 400px;
  height: 400px;
  margin-top: 30px;
  margin-left: auto;
  margin-right: auto;
  box-shadow: 10px 10px 0px #8d8d8d;
}
```

Now let's add that style to an element on our HTML page.

- Switch back to the **HTML** tab and in between the <body> tags add a new  tag. Give it the #id style you just created. To give an html element an ID style from our CSS add the text **id="painting"** inside the opening tag.

*Our HTML looks like this:*

```
<body>
  <div id="painting"></div>
</body>
```

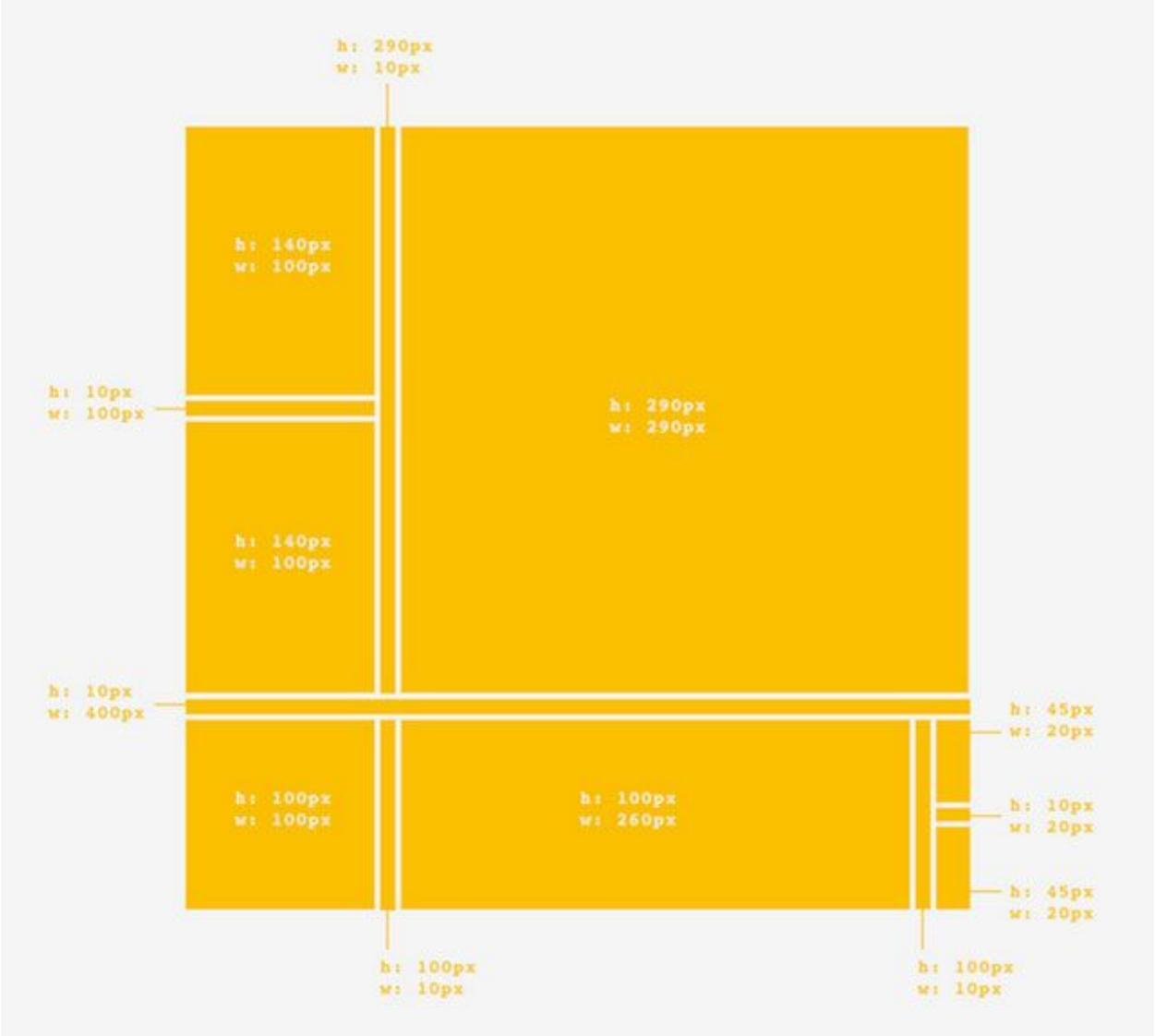
Divs are tags that help divide and segment a page. You can think of them like boxes that can be styled and positioned using CSS. You can also put stuff inside them, such as text, pictures, and even other divs. They are the basic building blocks of any website and a fundamental part of this project. So get ready to make a lot of them.

# Step 3: Let's start "div"-ing it up.

Now that we have our basic structure, it's time to define the structure of our painting. It's very helpful to plan out and divide the work into sections before we begin coding. Positioning and dividing up boxes with **DIV**s and **CSS** is much easier if you've already done the math.

We will be thinking about the works as rows, columns, and boxes. Each one of these will have their own CSS #id and one or more corresponding HTML s. Some of them will be visible and colored. Others will be invisible but serve to hold the boxes in proper position and alignment.

*Here's an illustration of how how these divisions will work and what the measurements for our finished piece will be:*




We'll start by making our top row the one with the big box on the right and two smaller boxes on the left.

- Go into the **CSS** tab and define a new #id for the top row. Since we made our **#painting** size **400px** our top row should have a height of **290px** to match the proportions of Mondrian's painting. Since we'll be nesting this inside our big  and want it to fill the full width we don't need to define that dimension. It will inherit that quality from its parent.

*Our CSS looks like this:*

```
#toprow {
   height: 290px;
}
```

- After you've done that then add a new ** ** in between the tags of your big painting on in the HTML page. Don't worry if you can't see it. This is one of those invisible divs we mentioned before. It will hold our boxes and columns in place. If you want to check its positioning add a temporary **background-color** to the **#toprow** style.

*Our HTML looks like this:*

```
<body>
   <div id="painting">
      <div id="toprow"></div>
   </div>
</body>
```

# Step 4: Make the big box

Alright, now let's start adding some of those boxes we talked about. We'll start with the biggest, most colorful one of them all: the big red box in the upper right.

- Let's create another #id for our big red box. This one is a square that takes up the full height of its row. Ours has a width and height of **290px**.
- Once you've done that add a corresponding ** ** to your HTML within the **#toprow ** you created in the previous step. Remember this box won't have any color yet. We'll show you a trick to color it in on the next step.

*Our HTML looks like this:*

```
<body>
   <div id="painting">
      <div id="toprow">
         <div id="bigbox"></div>
      </div>
   </div>
```

```
</body>
```

*The CSS addition looks like this:*

```
#bigbox {
   width: 290px;
   height: 290px;
}
```

## Step 5: Add some color and some class

Now let's give that box some color! Now you're probably thinking we could just add a **background-color** attribute to the **#bigbox** id we already created, and you would be correct! That would get the job done pretty well. But remember we want to give multiple boxes the same color later in the project. For that a CSS **.class** will be a better choice.

Classes are similar to the #id styles we've been working with except the same class can be used to style multiple elements, including DIVs that already have #ids in our layout. That will make more sense in just a bit. Let's make that .class first.

- In your CSS tab start a new line.

- Begin writing your first color class. It looks exactly the same as the other styles we've typed up so far except instead of **"#"** in front of the selector we put a **"."** (period). We are only defining color with these classes so the only attribute needs to be background-color. Try just "red" as the background-color.

```
.red {
   background-color: red;
}
```

If you'd prefer more control and variation in color you can use a hex value.

```
.red {
   background-color: #cc3333;
}
```

To apply the **.red** class to our **#bigbox** DIV, simply add the text **class="red"** to its tag in HTML.

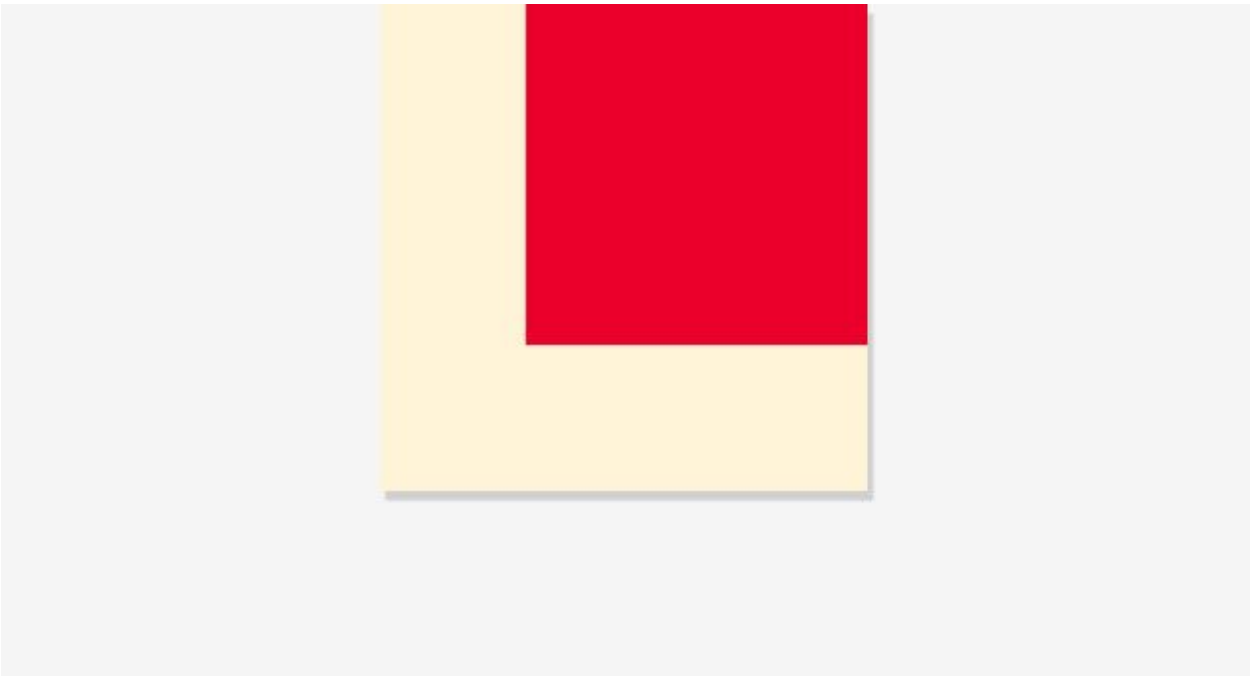
```
<body>
   <div id="painting">
      <div id="toprow">
         <div id="bigbox" class="red"></div>
      </div>
   </div>
</body>
```

- Now that you know how .class works, go ahead and make classes for the other colors in this painting: .blue, .yellow, and .black. You can make additional color classes as well.

```
.red {
   background-color: red;
}
.blue {
   background-color: blue;
}
.yellow {
   background-color: yellow;
}
```

```
.black {
   background-color: black;
}
```

## Step 6: Float it right



You probably noticed the big red box is on the wrong side. These things happen. Web elements will default to the upper left if they aren't assigned any other positioning. Let's move this box to the place its supposed to go.

There are many different ways of positioning an element the but method we will be using is called **float**. It makes things move (or float) either **right** or **left** until they bump into something. Normally a DIV will position itself underneath the previous element, but when DIVs are **"floated"**, they can align side-by-side until there is no more space and they are forced to wrap.

- Let's make a class called **.right** and use it to float the **bigbox** DIV to the right:

*Here's the relevant CSS:*

```
.right {
   float: right;
}
```

*And here's what changed in the HTML:*

```
<div id="bigbox" class="red right"></div>
```

Note that an element can only have a single **#id**, but it can have more than one **.class**.

Floated elements will stack up against each other. This means if we give all the boxes in our rows the .right .class they will float up next to each other and form a nice horizontal row. That's what we want for the next step.

## Step 7: Add a divider

Now let's add one of those black bars. We'll add them the same way we added the big red box. Make a new #id for the top divider and assign a color and position class to it in the HTML. our divider is 10px wide and 290px tall.

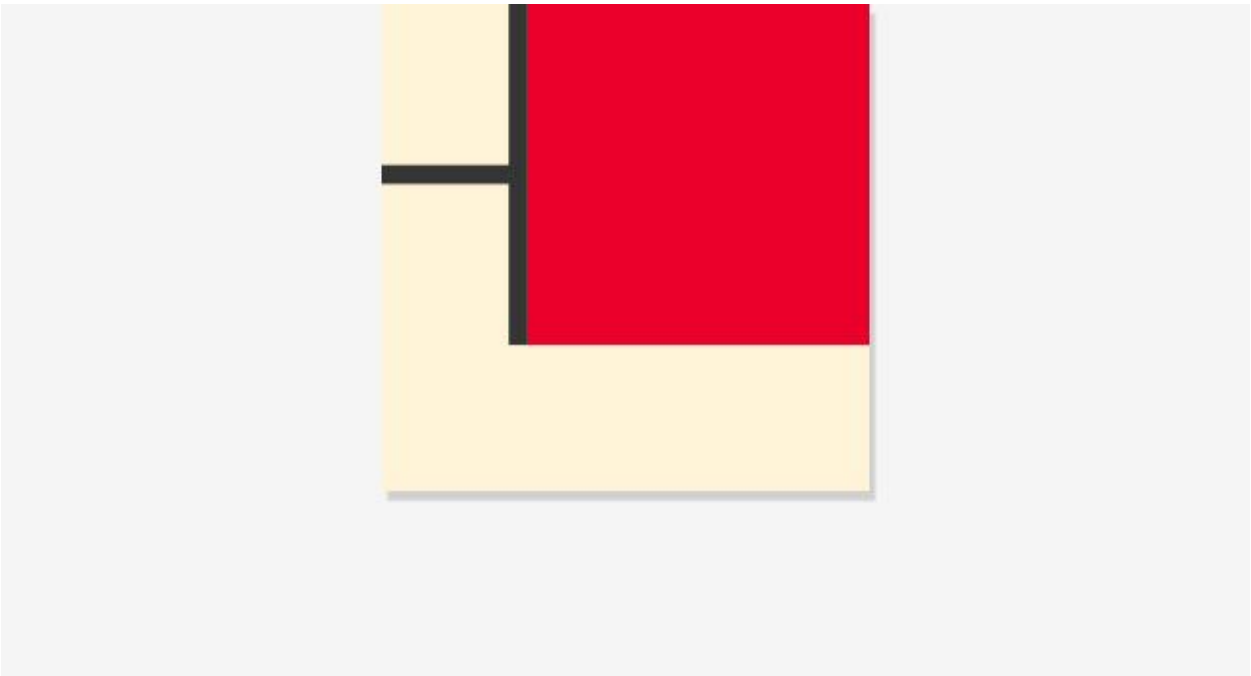
*Our CSS code looks like this:*

```
#divider1 {
   height: 290px;
   width: 10px;
}
```

*And our HTML:*

```
<body>
   <div id="painting">
      <div id="toprow">
         <div id="bigbox" class="red right"></div>
         <div id="divider1" class="black right"></div>
      </div>
   </div>
</body>
```

## Step 8: Make a column



Now let's make a column, called **#topleftcolumn**. We will be putting three boxes in it, two with the **.mediumbox** class, and a horizontal **#divider2** in between.

**#topleftcolumn** won't be visible, but it helps position and stack those three boxes vertically within that last bit of the row.

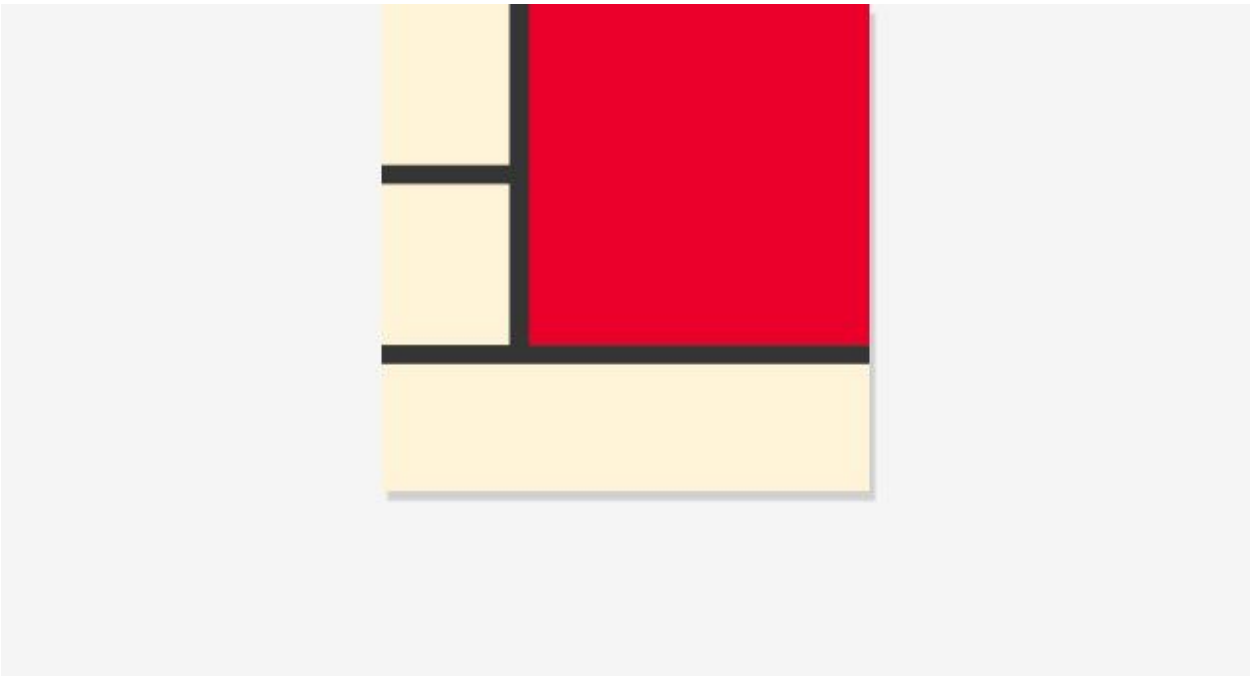
```
<body>
   <div id="painting">
      <div id="toprow">
         <div id="bigbox" class="red right"></div>
         <div id="divider1" class="black right"></div>
         <div id="topleftcolumn" class= "right">
            <div class="mediumbox"></div>
            <div id="divider2" class= "black"></div>
            <div class="mediumbox"></div>
         </div>
      </div>
   </div>
</body>
```

*Our CSS additions look like:*

```
.mediumbox {
    width: 100px;
    height: 140px;
}
#divider2 {
    height: 10px;
    width: 100px;
}
```

Since the two boxes are the exact same size we can use one **.class** for both of them instead of using a unique **#id**.

## Step 9: Do the middle row divider



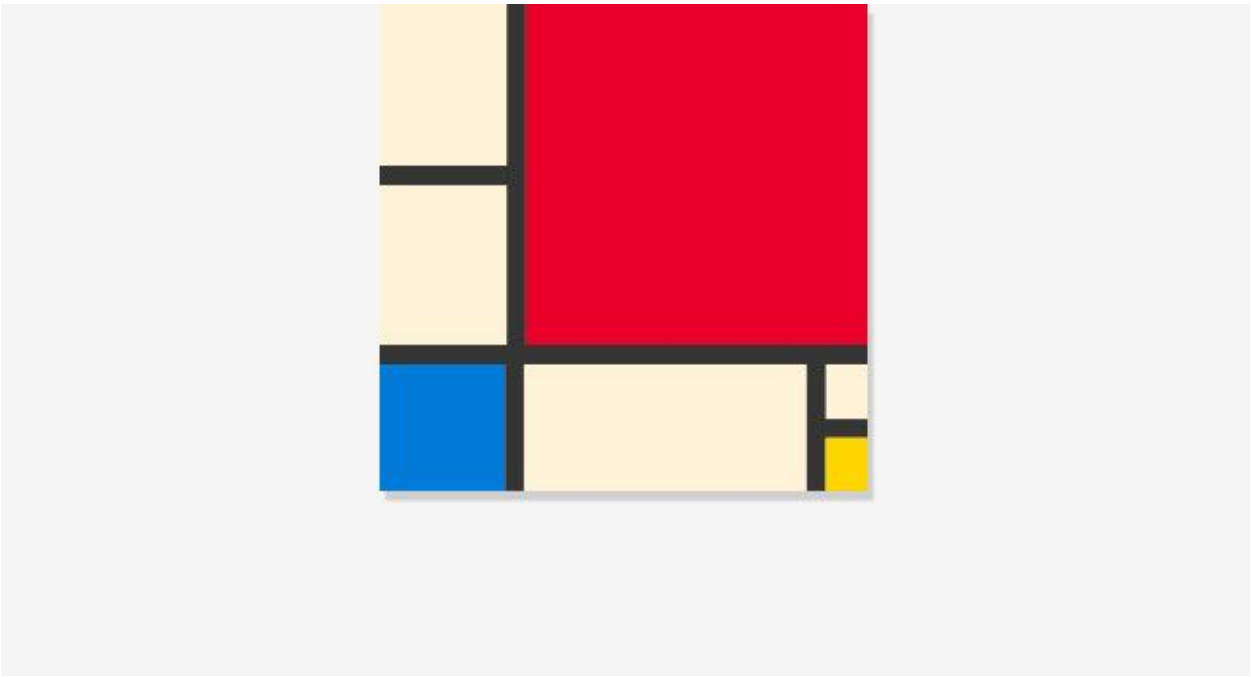
This is just another black divider, but it's not inside **#toprow**.

*Here's the HTML markup. Can you figure out the CSS?*

```
<body>
    <div id="painting">
```

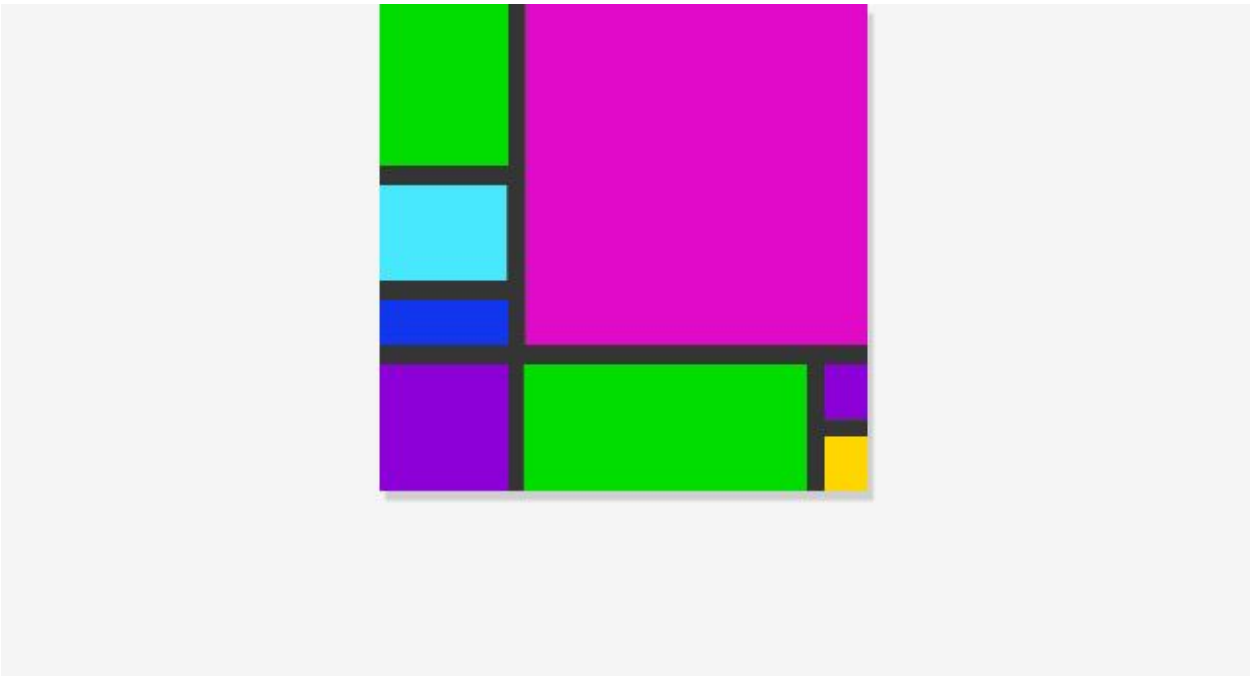
```
      <div id="toprow">
         <div id="bigbox" class="red right"></div>
         <div id="divider1" class="black right"></div>
         <div id="topleftcolumn" class= "right">
            <div class="mediumbox"></div>
            <div id="divider2" class= "black"></div>
            <div class="mediumbox"></div>
         </div>
      </div>
      <div id="middlerow"></div>
   </div>
</body>
```

## Step 10: Fill out the bottom row



Now you're ready to do the last row your self. You have all the pieces and knowledge at your disposal. The process is the same as the top row we just completed just with different sizes and colors. Keep an eye on those little boxes at the right. They'll need their own column.

# Step 11: Mess around with it



You’re done! Doesn’t it look great? Actually, maybe it doesn’t look that great. Maybe you want to make it wilder, add more colors or change the ones that are already there. Good thing that’s easy to do. Go back into the CSS tab and start changing the values of the color classes we made. If you’re feeling really crazy, maybe change some box sizes or add new ones. See if you can make it a whole new painting.