### fillRect

.fillRect is a "stand-alone" command that draws and fills a rectangle.

So if you issue multiple .fillRect commands with multiple .fillStyle commands, each new rect will be filled with the preceeding fillstyle.

.fillRect 是一个**独立**的命令，它可以勾勒出一个矩形的形状。

如果你把多个.fillStyle命令和多个.fillRect命令搭配使用，每个矩形都会是离它最近的那个.fillStyle规定的颜色。

ctx.fillStyle="red";

ctx.fillRect(10,10,10,10); // filled with red 该矩形是红色

ctx.fillStyle="green";

ctx.fillRect(20,20,10,10); // filled with green 该矩形是绿色

ctx.fillStyle="blue";

ctx.fillRect(30,30,10,10); // filled with blue 该矩形是蓝色

### rect

.rect is part of the canvas's path commands.

Path commands are ****groups of drawings**** beginning with the beginPath() and continuing until another beginPath() is issued.

Within each group, only the last styling command wins.

So if you issue multiple .rect commands and multiple .fillStyle commands inside a path, only the last .fillStyle will be used on all the .rect's.

.rect 是canvas路径命令的**一部分**。

路径命令是你绘制的多个形状的集合，它从一个beginPath() 开始，直到遇到另一个beginPath() 结束，都只算一条路径。

在每一个集合内，只有最后一个fillStyle / strokeStyle 才起作用。

所以在同一个路径之内，如果你把多少.fillStyle命令和多个.rect命令搭配使用，只有**最后一个**fillStyle / strokeStyle起作用，它会被应用于之前**所有**的.rect。

ctx.beginPath(); // path commands must begin with beginPath

ctx.fillStyle="red";

ctx.rect(10,10,10,10); // blue

ctx.fillStyle="green";

ctx.rect(20,20,10,10); // blue

ctx.fillStyle="blue"; // this is the last fillStyle, so it "wins"

ctx.rect(30,30,10,10); // blue

// only 1 fillStyle is allowed per beginPath, so the last blue style fills all

ctx.fill()

表盘画在一个canvas里面，其他的画在另一个canvas里面，这些就互不影响，显示的时候把两个都显示出来。你可以搜下canvas嵌套试试。