

> sketch.js

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(255, 0, 0);  
7 }
```

宽 高  
① 宽  
② 高  
③ 宽  
④ 高  
⑤ 宽  
⑥ 高  
⑦ 宽  
⑧ 高

① function 函数: blocks of code 代码块  
② setup 设置: a series of things that we want to do when we first load the code. 第一次加载代码时想做的一系列事情。(只想做一遍)

③ draw 画: 相当于一个循环动作, 可以接下来继续画. pool 循环  
④ RGB 颜色: 红. 绿. 蓝组成颜色. 在RGB颜色中, 通常有3个值, 它们代表红. 绿. 蓝, 从0-255 (总共有256个值, 因为是从0开始的, 所以0也包含在内). 可以添加第四个值来代表透明度(同样是0-225).  
也可以写单数字, 单数字相当于从黑(0)到纯白(255).

```
4  
5 function draw() {  
6   background(220);  
7 }  
8 rect(50, 100, 40, 60);  
9 }
```

⑨ 宽 高  
⑩ 宽 高  
⑪ 宽 高  
⑫ 宽 高

⑤ 形状: rectangle 长方形的缩写.  
⑥ 从左上角开始的, 首先是形状在画布上的位置(xy轴), 其次是形状的宽和高.

⑦ 完成一项要加分号.

Preview

得到一个描黑边的白色长方形

Play Sketch

Preview

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7 }  
8 stroke(255, 0, 0);  
9 fill(255, 255, 0, 255);  
10 rect(50, 100, 40, 60);  
11  
12 }
```

描边(RGB颜色)  
填充(RGB颜色)

```
1 function setup() {  
2   createCanvas(400, 400);  
3 }  
4  
5 function draw() {  
6   background(220);  
7 }  
8 stroke(255, 0, 0);  
9 fill(255, 255, 0);  
10 rect(50, 100, 40, 60);  
11  
12 fill(0, 0, 255);  
13 rect(150, 100, 40, 60);  
14 }
```



另外一个形状就另起一行重复以上操作.

# key coding: variable 变量

```
> sketch.js•  
1 let x = 0; ①  
2  
3 function setup() {  
4   createCanvas(400, 400);  
5 }  
6  
7 function draw() {  
8   background(220);  
9  
10  fill(255, 255, 0);  
11  rect(x, 100, 40, 60);  
12  
13  x = x + 3; ②  
14  
15  console.log(x);  
16 } 按控制台日志  
③ Console ④ 出现矩形的移动  
522 525 528
```

① 让  $x=0$ , 其中,  $x$  就是 variable 的名字. 当我们将其其中一个值用  $x$  代表, 那么, 那个值就是 0.  
(这里呈现的图像是长方形在画布的最左边, 因为  $x$  轴的值被 variable 替换了).

②  $x=x+3$  是下一个循环, 再下一个循环中,  $x$  将变成 3. 然后又变成  $x=x+3$  ( $x=3+3, x=6+3, x=9+3, x=12+3 \dots \dots$ ) 这里呈现的图像是矩形一直向右移动, 并再也不回来.

③ 得到每次经过时  $x$  的值 (默认帧率是 30 or 60).

## system variable (不需要额外顶格写)

```
7 function draw() {  
8   background(220);  
9  
10  fill(255, 255, 0);  
11  rect(mouseX, mouseY, 40, 60);  
12  ④ ⑤ T
```

④ 这就是 system variable 系统变量.  
 $mouseX$  是图形跟着鼠标横向移动.  
 $mouseY$  就是纵向移动.

## key coding 2: conditionals (检查某事是否正确, 然后根据答案做些事情)

```
13  x = x + 3;  
14  
15  ⑤ if(x > width) {  
16    x = 0;  
17  } 这里不是 mouseX  
18  
//  x = x + 3;  
//  if(x > width) {  
//    x = 0;  
//  }
```

⑤ 如果 ( $x >$  宽/数字) { 大括号并另起一行  
就让  $x=0$ ;  
} 这段代码可以让图形重新回起点  
选中这段代码然后按 (ctrl) 可以让它  
们暂时不运行.

```
1 let x = 0;
2
3 function setup() {
4   createCanvas(400, 400);
5 }
6
7 function draw() {
8   background(220);
9
10  if (mouseX > width / 2) { ①
11    fill(255, 0, 0);
12  } else {
13    fill(255, 255, 0);
14  }
15
16  rect(50, 100, 40, 60);
17 }
```

图像呈现的是当鼠标划过宽的一半后，就……  
长方形变成黄色。

①如果鼠标的位置由  $x > width / 2$   
写作  $width / 2$  (width/2) 也可以。  
就……  
else 如果不。