**Pygame 官方文档 - pygame.PixelArray**

pygame.PixelArray

pygame对象用于surfaces的直接像素访问  
[pygame.PixelArray.surface](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#surface) - 获取PixelArray使用的Surface。  
[pygame.PixelArray.itemsize](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#itemsize) - 返回像素数组项的字节大小  
[pygame.PixelArray.ndim](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#ndim) - 返回维数。  
[pygame.PixelArray.shape](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#shape) - 返回数组大小。  
[pygame.PixelArray.strides](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#strides) - 返回每个数组维度的字节偏移量。  
[pygame.PixelArray.make\_surface](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#make_surface%28%29) - 从当前的PixelArray创建一个新的Surface。  
[pygame.PixelArray.replace](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#replace%28%29) - 将PixelArray中传递的颜色替换为另一个。  
[pygame.PixelArray.extract](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#extract%28%29) - 从PixelArray中提取传递的颜色。  
[pygame.PixelArray.compare](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#compare%28%29) - 将PixelArray与另一个进行比较。  
[pygame.PixelArray.transpose](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#transpose%28%29) - 交换x和y轴。  
[pygame.PixelArray.close](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#close%28%29) - 关闭PixelArray，并释放Surface锁。

PixelArray包裹Surface并提供对曲面像素的直接访问。 像素数组可以是一维或二维的。 二维数组（就像它的的surface）被索引为[column，row]。 像素数组支持切片，用于返回子数组或用于赋值。 在单个列或行上切片的像素数组返回一维像素数组。 不支持算术运算和其他运算。 像素数组可以安全地赋值给自己。 最后，像素数组导出一个数组结构接口，允许它们与pygame.pixelcopy方法和NumPy数组交互。

可以为PixelArray像素项指定原始整数值，pygame.Color实例，或者（r，g，b [，a]）元组。

pxarray[x, y] = 0xFF00FF

pxarray[x, y] = pygame.Color(255, 0, 255)

pxarray[x, y] = (255, 0, 255)

但是，仅返回像素的整数值。 因此，要将像素与特定颜色进行比较，需要首先使用为其创建PixelArray的Surface对象的Surface.map\_rgb()方法来映射颜色。

pxarray = pygame.PixelArray(surface)

#检查左上角的第一个像素是否为蓝色

if pxarray[0, 0] == surface.map\_rgb((0, 0, 255)):

...

分配一系列像素时，可以使用非元组颜色序列或PixelArray作为值。 对于序列，长度必须与PixelArray宽度匹配。

pxarray[a:b] = 0xFF00FF # 把所有的像素设置为 0xFF00FF

pxarray[a:b] = (0xFF00FF, 0xAACCEE, ... ) # 第一个像素 = 0xFF00FF,

# 第二个像素 = 0xAACCEE, ...

pxarray[a:b] = [(255, 0, 255), (170, 204, 238), ...] # 和上面一样

pxarray[a:b] = [(255, 0, 255), 0xAACCEE, ...] # 和上面一样

pxarray[a:b] = otherarray[x:y] # 切片大小必须匹配

对于PixelArray赋值，如果右侧数组的行长度为1，则在目标数组的行上散布该列。 高度为1的数组在目标列上散布，相当于分配1D PixelArray。

下标切片也可用于分配目标PixelArray的矩形子视图。

＃创建一些新的PixelArray对象，提供原始数组/表面的不同视图。

newarray = pxarray[2:4, 3:5]

otherarray = pxarray[::2, ::2]

下标切片也可用于进行快速矩形像素操作，而不是在x或y轴上迭代。

pxarray[::2, :] = (0, 0, 0) # 偶数列变为黑色

pxarray[::2] = (0, 0, 0) # 和 [::2, :] 一样

在它的生命周期中，PixelArray锁定表面，因此一旦不再使用它就明确地必须关闭它（）表面应该在相同的范围内执行操作。 最好将它作为上下文管理器使用with PixelArray（surf）作为pixel\_array：style。 所以它也适用于pypy。

一个简单的：列的切片索引可以省略。

pxarray[::2, ...] = (0, 0, 0) # 和 pxarray[::2, :]一样

pxarray[...] = (255, 0, 0) # 和 pxarray[:]一样

关于PixelArray到PixelArray赋值的注解，对于项大小为3（从24位surface创建）的数组，像素值将从源转换为目标格式。每个像素的红色，绿色和蓝色元素被移位以匹配目标surface的格式。对于所有其他像素大小，不会发生这种重新映射。这应该在以后的pygame版本中更改，其中对所有像素大小执行格式转换。为了避免在实现完全映射复制时代码破坏，建议PixelArray到PixelArray副本仅在相同格式的surfaces之间。

pygame 1.9.4中的新功能

1. 添加了close()方法。明确清理。
2. 能够使用PixelArray作为清理的上下文管理器。
3. 在没有引用计数（pypy）的情况下工作时，这两者都很有用。

pygame 1.9.2中的新功能

1. 数组结构接口
2. 转置方法
3. 散布长度为1维

pygame 1.9.2的改进

1. 2D PixelArray可以具有1维长度。 2D PixelArray上只有一个整数索引返回一维数组。
2. 对于赋值，元组只能是一种颜色。任何其他序列类型都是一系列颜色。

pygame.PixelArray.surface [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

获取PixelArray使用的Surface。  
surface -> Surface  
为PixelArray创建了Surface。  
[搜索pygame.PixelArray.surface的示例](https://github.com/search?q=pygame.PixelArray.surface&type=Code)

pygame.PixelArray.itemsize [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

返回像素数组项的字节大小  
itemsize -> int  
这与像素数组表面的Surface.get\_bytesize()相同。  
pygame 1.9.2中的新功能  
[搜索pygame.PixelArray.itemsize的示例](https://github.com/search?q=pygame.PixelArray.itemsize&type=Code)

pygame.PixelArray.ndim [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

返回维数。  
ndim -> int  
像素数组可以是1维或2维。  
pygame 1.9.2中的新功能  
[搜索pygame.PixelArray.ndim的示例](https://github.com/search?q=pygame.PixelArray.ndim&type=Code)

pygame.PixelArray.shape [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

返回数组大小。  
shape -> tuple of int’s  
给出每个维度长度的元组或长度ndim。 类似于Surface.get\_size()。  
pygame 1.9.2中的新功能  
[搜索pygame.PixelArray.shape的示例](https://github.com/search?q=pygame.PixelArray.shape&type=Code)

pygame.PixelArray.strides [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

返回每个数组维度的字节偏移量。  
strides -> tuple of int’s  
元组或长度ndim字节计数。 当stride乘以相应的索引时，它会给出该索引与数组开头的偏移量。 对于已反转的数组（具有负步长），stride为负。  
pygame 1.9.2中的新功能  
[搜索pygame.PixelArray.strides的示例](https://github.com/search?q=pygame.PixelArray.strides&type=Code)

pygame.PixelArray.make\_surface() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

从当前的PixelArray创建一个新的Surface。  
make\_surface() -> Surface  
从当前的PixelArray创建一个新的Surface。 根据当前PixelArray的大小，像素顺序等将与原始Surface不同。

# 创建一个垂直轴翻转的新surface

sf = pxarray[:,::-1].make\_surface ()

pygame 1.8.1的新功能  
[搜索pygame.PixelArray.make\_surface的示例](https://github.com/search?q=pygame.PixelArray.make_surface&type=Code)

pygame.PixelArray.replace() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

将PixelArray中传递的颜色替换为另一个。  
replace(color, repcolor, distance=0, weights=(0.299, 0.587, 0.114)) -> None  
使用PixelArray中传递的颜色替换像素，方法是将它们更改为传递的替换颜色。  
它使用简单的加权欧氏距离公式（ weighted Euclidean distance formula）来计算颜色之间的距离。 距离空间的范围为0.0到1.0，并用作颜色检测的阈值。 这使得替换也考虑具有类似但不完全相同的颜色的像素。  
这是一个直接影响PixelArray像素的就地操作。  
pygame 1.8.1中的新功能。  
[搜索pygame.PixelArray.replace的示例](https://github.com/search?q=pygame.PixelArray.replace&type=Code)

pygame.PixelArray.extract() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

从PixelArray中提取传递的颜色。  
extract(color, distance=0, weights=(0.299, 0.587, 0.114)) -> PixelArray  
通过将所有匹配像素更改为白色来提取传递的颜色，而将不匹配的像素更改为黑色。 这将返回带有黑/白色遮罩的新PixelArray。  
它使用简单的加权欧氏距离公式来计算颜色之间的距离。 距离空间的范围为0.0到1.0，并用作颜色检测的阈值。 这使得提取也考虑具有类似但不完全相同的颜色的像素。  
pygame 1.8.1中的新功能。  
[搜索pygame.PixelArray.extract的示例](https://github.com/search?q=pygame.PixelArray.extract&type=Code)

pygame.PixelArray.compare() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

将PixelArray与另一个进行比较。  
compare(array, distance=0, weights=(0.299, 0.587, 0.114)) -> PixelArray  
将PixelArray的内容与传递的PixelArray中的内容进行比较。 它返回一个带有黑/白色遮罩的新PixelArray，它指示两个数组的差异（白色）。 两个PixelArray对象必须具有相同的位深度和尺寸。  
它使用简单的加权欧氏距离公式来计算颜色之间的距离。 距离空间的范围为0.0到1.0，并用作颜色检测的阈值。 这导致比较将具有与黑色相似但不完全相同的颜色的像素标记为黑色。  
pygame 1.8.1中的新功能。  
[搜索pygame.PixelArray.compare的示例](https://github.com/search?q=pygame.PixelArray.compare&type=Code)

pygame.PixelArray.transpose() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

交换x和y轴。  
transpose() -> PixelArray  
此方法返回像素数组的新视图，其中包含交换的行和列。 因此，对于（w，h）大小的数组，返回（h，w）切片。 如果阵列是一维的，则添加长度为1 x的维度，从而产生2D像素数组。  
pygame 1.9.2中的新功能  
[搜索pygame.PixelArray.transpose的示例](https://github.com/search?q=pygame.PixelArray.transpose&type=Code)

pygame.PixelArray.close() [↶](https://blog.csdn.net/Enderman_xiaohei/article/details/88362408#title)

关闭PixelArray，并释放Surface锁。  
close() -> PixelArray  
此方法用于显式关闭PixelArray，并释放Suface上的锁定。  
pygame 1.9.4中的新功能  
[搜索pygame.PixelArray.close的示例](https://github.com/search?q=pygame.PixelArray.close&type=Code)

以上文档，自己翻译，可能有误，可参考：[pygame.PixelArray](https://www.pygame.org/docs/ref/pixelarray.html)