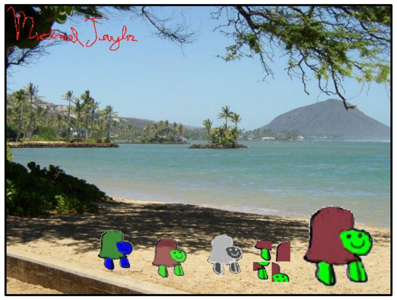
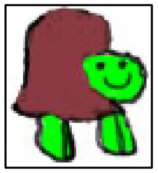
Michael Taylor

Completed



Originals



```
#Michael Taylor, March 14th 2021

import random

def scale(picture, factor):
  scaled = makeEmptyPicture(int(getWidth(picture)*factor), int(getHeight(picture)*factor))
  srcx = 0
  for x in range(int(getWidth(picture)*factor)):
    srcy = 0
    for y in range(int(getHeight(picture)*factor)):
      px = getPixel(picture, int(srcx), int(srcy))
      color = getColor(px)
      setColor(getPixel(scaled, x, y), color)
      srcy = srcy + 1.0/factor
    srcx = srcx + 1.0/factor
  return scaled

def colorSwap(pic):
  colorSwap = duplicatePicture(pic)
  for px in getPixels(colorSwap):
    r = getBlue(px)
    g = getRed(px)
    b = getGreen(px)
    setColor(px, makeColor(r, g, b))
  return colorSwap
```

```
def waffle(pic):
  waffled = duplicatePicture(pic)
  for x in range(0, getWidth(waffled), 2):
    for y in range(0, getHeight(waffled), 2):
      setColor(getPixel(waffled, x, y), white)
  return waffled

def inverse(pic, w, h):
  inverted = duplicatePicture(pic)
  for x in range(w):
    for y in range(h):
      px = getPixel(inverted, x, y)
      level = 255 - int(0.21*getRed(px) + 0.71*getGreen(px) + 0.07*getBlue(px))
      if (level < 15):
        color = makeColor(255, 255, 255)
      else:
        color = makeColor(level, level, level)
      setColor(px, color)
  return inverted

def swap(h, w, newPic, pic, startX, startY, endX, endY, goX, goY):
  for x in range(startX, endX):
    for y in range(startY, endY):
      color = getColor(getPixel(pic, x, y))
      setColor(getPixel(newPic, x+goX, y+goY), color)

def copy(turtle, canvas, startX, startY):
  srcx = 0
  for x in range(startX, startX+getWidth(turtle)):
    srcy = 0
    for y in range(startY, startY+getHeight(turtle)):
      color = getColor(getPixel(turtle, srcx, srcy))
      setColor(getPixel(canvas, x, y), color)
      srcy = srcy + 1
    srcx = srcx + 1
  return canvas

def chromakey(bkgr, canvas):
  for px in getPixels(canvas):
    x = getX(px)
    y = getY(px)
    if (getRed(px) > 230 and getGreen(px) > 230 and getBlue(px) > 230):
      bgpx = getPixel(bkgr, x, y)
      bgcol = getColor(bgpx)
      setColor(px, bgcol)
```

```
def randomTurtle(turtle, first, second, third, forth, fifth):
  if (turtle == 0):
    turtle = first
  elif (turtle == 1):
    turtle = second
  elif (turtle == 2):
    turtle = third
  elif (turtle == 3):
    turtle = forth
  elif (turtle == 4):
    turtle = fifth
  else:
    return
  return turtle

def collage():
  import random
  sig = makePicture(getMediaPath("signature.png"))
  pic = makePicture(getMediaPath("turtle.jpg"))
  bkgr = makePicture(getMediaPath("beach.jpg"))
  w = getWidth(pic)
  h = getHeight(pic)
  canvas = makeEmptyPicture(getWidth(bkgr), getHeight(bkgr))
  swapPic = makeEmptyPicture(w, h)
  first = inverse(pic, w, h)
  second = pic
  swap(h, w, swapPic, pic, 0, 0, w/2, h/2, w/2, 0)
  swap(h, w, swapPic, pic, w/2, 0, w, h/2, 0, h/2)
  swap(h, w, swapPic, pic, w/2, h/2, w, h, -(w/2), 0)
  swap(h, w, swapPic, pic, 0, h/2, w/2, h, 0, -(h/2))
  third = swapPic
  forth = colorSwap(pic)
  fifth = waffle(pic)
  ran = random.sample(xrange(5),5)
  turtle1 = randomTurtle(ran[0], first, second, third, forth, fifth)
  turtle2 = randomTurtle(ran[1], first, second, third, forth, fifth)
  turtle3 = randomTurtle(ran[2], first, second, third, forth, fifth)
  turtle4 = randomTurtle(ran[3], first, second, third, forth, fifth)
  turtle5 = randomTurtle(ran[4], first, second, third, forth, fifth)
  copy(scale(turtle1, 2), canvas, 490, 332)
  copy(turtle2, canvas, 411, 391)
  copy(turtle3, canvas, 333, 380)
  copy(turtle4, canvas, 240, 385)
  copy(turtle5, canvas, 150, 372)
```

```
signature = scale(sig, 0.25)
copy(signature, canvas, 0,0)
chromakey(bkgr, canvas)
explore(canvas)
```