<https://robertheaton.com/2018/06/12/programming-projects-for-advanced-beginners-ascii-art/>

* In cmd wrote:
  + C:\Users\saren\Desktop>py -m pip install pillow
* To get the image, wrote in code
  + from PIL import Image
  + jpgfile = Image.open("pineapple.jpg")
  + print(jpgfile.bits, jpgfile.size, jpgfile.format)
* I imported numpy through the cmd bar
  + py -m pip install numpy
* I changed jpgfile to picture and also stored it in array (tuple) of all the pixels
  + from PIL import Image
  + import numpy as np
  + picture = Image.open("pineapple.jpg")
  + print(picture.bits, picture.size, picture.format)
  + iar = np.asarray(picture)
  + print(iar)
* I iterated through the array and printed my results which on for many thousands of lines
  + width, height = picture.size
  + for x in range(width):
  + for y in range(height):
  + r,g,b = picture.getpixel((x,y))
  + print (r,g,b)
* I then created a new array that would hold the brightness idex of each pixel. It does this by finding the average of each pixel’s r g and b. My code now looks like this:
  + from PIL import Image
  + import numpy as np
  + picture = Image.open("kosel-1.jpg")
  + #print(picture.bits, picture.size, picture.format)
  + pixlArray = np.asarray(picture)
  + size = pixlArray.shape
  + #small array
  + print ("The array is",size)
  + print(pixlArray)
  + width, height = picture.size
  + # big array
  + brightArray = []
  + i=0
  + for x in range(width):
  + for y in range(height):
  + r,g,b = picture.getpixel((x,y))
  + bright=(r + g + b) / 3
  + brightArray.append(bright)
  + i+=1
  + #print(len(brightArray), "is the shape of bright array")
  + #print (r,g,b)
* I made an array of all the symbols from thinnest to boldest
  + symblArray=["`","^","\\",'"',",",":",";","I","l","!","i","~","+","\_","-","?","]",
  + "[","}","{","1",")","(","|","\\\\","/","t","f",
  + "j","r","x","n","u","v","c","z","X","Y","U",
  + "J","C","L","Q","0","O","Z","m","w","q","p",
  + "d","b","k","h","a","o","\*","#","M","W","&","8","%","B","@","$"]
  + print (symblArray)
* To go from the 255 brightness levels to the 70 symbols I used integer division, dividing the brightness level by 3.642857142857143. I then found the corresponding symbol to that number and entered into the display array. Here is the code
  + displayArray=[]
  + j=0
  + conversion=0
  + print (len(brightArray))
  + while j<len(brightArray):
  + conversion=(brightArray[j])//(3.642857142857143)
  + conversion=int(conversion)
  + displayArray.append(symblArray[conversion])
  + print(displayArray[j])
  + j+=1
* I changed the code so that it finds the conversion based on size and then I had it output each symbol 3 times so that the picture would not be squashed. Also, I told it to enter each new line, not after each character.
  + from PIL import Image
  + import numpy as np
  + picture = Image.open("kosel-1.jpg")
  + #print(picture.bits, picture.size, picture.format)
  + pixlArray = np.asarray(picture)
  + size = pixlArray.shape
  + #small array
  + print ("The array is",size)
  + print(pixlArray)
  + width, height = picture.size
  + # big array
  + brightArray = []
  + i=0
  + for x in range(width):
  + for y in range(height):
  + r,g,b = picture.getpixel((x,y))
  + bright=(r + g + b) / 3
  + brightArray.append(bright)
  + i+=1
  + symblArray=["`","^","\\",'"',",",":",";","I","l","!","i","~","+","\_","-","?","]",
  + "[","}","{","1",")","(","|","\\\\","/","t","f",
  + "j","r","x","n","u","v","c","z","X","Y","U",
  + "J","C","L","Q","0","O","Z","m","w","q","p",
  + "d","b","k","h","a","o","\*","#","M","W","&","8","%","B","@","$"]
  + print (symblArray)
  + displayArray=[]
  + j=0
  + f=0
  + conversion=float((max(brightArray)/len(symblArray)-1))
  + #print (conversion)
  + while j<len(brightArray):
  + symbolIndex=(brightArray[j])//(conversion)
  + symbolIndex=int(symbolIndex)
  + if (symbolIndex<0):
  + symbolIndex=0
  + elif (symbolIndex>len(symblArray)):
  + symbolIndex=len(symblArray)-1
  + displayArray.append(symblArray[symbolIndex-1])
  + if (f<width):
  + print((displayArray[j])\*3, end="")
  + f+=1
  + else:
  + print ()
  + print((displayArray[j])\*3, end="")
  + f=0
  + j+=1
* In command prompt wrote
  + py -m pip install python-resize-image
* This is my final code so that the picture does not end up sideways (switched x and y when reading in pixels), And so that if it has to many character that it will go onto the next line, that it will stop running and let you know that.
  + from PIL import Image
  + import numpy as np
  + #import Exif#Tags as exif
  + from resizeimage import resizeimage
  + picture = Image.open("castle.jpg")
  + pixlArray = np.asarray(picture)
  + size = pixlArray.shape
  + width, height = picture.size
  + picture = resizeimage.resize\_cover(picture, [width\*2/3, height\*2/3])
  + width, height = picture.size
  + brightArray = []
  + i=0
  + for x in range(height):
  + for y in range(width):
  + r,g,b = picture.getpixel((y,x))
  + bright=(r + g + b) / 3
  + brightArray.append(bright)
  + i+=1
  + symblArray=["`","^","\\",'"',",",":",";","I","l","!","i","~","+","\_","-","?","]",
  + "[","}","{","1",")","(","|","\\\\","/","t","f",
  + "j","r","x","n","u","v","c","z","X","Y","U",
  + "J","C","L","Q","0","O","Z","m","w","q","p",
  + "d","b","k","h","a","o","\*","#","M","W","&","8","%","B","@","$"]
  + displayArray=[]
  + j=0
  + f=0
  + conversion=float((max(brightArray)/len(symblArray)-1))
  + while j<len(brightArray):
  + symbolIndex=(brightArray[j])//(conversion)
  + symbolIndex=int(symbolIndex)
  + if (symbolIndex<0):
  + symbolIndex=0
  + elif (symbolIndex>len(symblArray)-1):
  + symbolIndex=len(symblArray)-1
  + displayArray.append(symblArray[symbolIndex])
  + if (f>225): # this is so it fits my screen
  + print("\nYour picture is to big")
  + break
  + if (f<width-1):
  + print((displayArray[j]\*2), end="")
  + f+=1
  + else:
  + print ()
  + f=0
  + j+1
* I then moved my code to thonny where I could make the text print out in smaller, first I had to import lthe packages in a thonny compatible way though, here is the code for that.
  + from PIL import Image
  + import numpy as np
  + import image
  + #import Exif#Tags as exif
  + from resizeimage import resizeimage
  + picture = Image.open("llchaim.jpg")
  + pixlArray = np.asarray(picture)
  + size = pixlArray.shape
  + width, height = picture.size
  + picture = resizeimage.resize\_cover(picture, [width\*3/8, height\*3/8])
  + width, height = picture.size
  + brightArray = []
  + i=0
  + for x in range(height):
  + for y in range(width):
  + r,g,b = picture.getpixel((y,x))
  + bright=(r + g + b) / 3
  + brightArray.append(bright)
  + i+=1
  + symblArray=["`","^","\\",'"',",",":",";","I","l","!","i","~","+","\_","-","?","]",
  + "[","}","{","1",")","(","|","\\\\","/","t","f",
  + "j","r","x","n","u","v","c","z","X","Y","U",
  + "J","C","L","Q","0","O","Z","m","w","q","p",
  + "d","b","k","h","a","o","\*","#","M","W","&","8","%","B","@","$"]
  + displayArray=[]
  + j=0
  + f=0
  + conversion=float((max(brightArray)/len(symblArray)-1))
  + while j<len(brightArray):
  + symbolIndex=(brightArray[j])//(conversion)
  + symbolIndex=int(symbolIndex)
  + if (symbolIndex<0):
  + symbolIndex=0
  + elif (symbolIndex>len(symblArray)-1):
  + symbolIndex=len(symblArray)-1
  + displayArray.append(symblArray[symbolIndex])
  + # if (f>225): # this is so it fits my screen
  + # print("\nYour picture is to big")
  + # break
  + if (f<width-1):
  + print((displayArray[j]\*2), end="")
  + f+=1
  + else:
  + print ()
  + f=0
  + j+=1
* I then changed the code so that any size picture would automatically be resized to fit the screen. I also changed to output to green text
  + from PIL import Image
  + import numpy as np
  + import image
  + #import Exif#Tags as exif
  + from resizeimage import resizeimage
  + from colorama import Fore, Back, Style
  + picture = Image.open("room.jpg")
  + pixlArray = np.asarray(picture)
  + size = pixlArray.shape
  + width, height = picture.size
  + screenChars=215 #how many charecters fit on my screen
  + w=1 #what you will enlarge the width of the image by
  + h=1 #what you will enlarge the length of the image by
  + while ((width>screenChars) and (w\*width>screenChars)):
  + w=w/1.2
  + h=h/1.2
  + picture = resizeimage.resize\_cover(picture, [width\*w, height\*h])
  + width, height = picture.size
  + brightArray = []
  + i=0
  + for x in range(height):
  + for y in range(width):
  + r,g,b = picture.getpixel((y,x))
  + bright=(r + g + b) / 3
  + brightArray.append(bright)
  + i+=1
  + symblArray=["`","^","\\",'"',",",":",";","I","l","!","i","~","+","\_","-","?","]",
  + "[","}","{","1",")","(","|","\\\\","/","t","f",
  + "j","r","x","n","u","v","c","z","X","Y","U",
  + "J","C","L","Q","0","O","Z","m","w","q","p",
  + "d","b","k","h","a","o","\*","#","M","W","&","8","%","B","@","$"]
  + displayArray=[]
  + j=0
  + f=0
  + conversion=float((max(brightArray)/len(symblArray)-1))
  + while j<len(brightArray):
  + symbolIndex=(brightArray[j])//(conversion)
  + symbolIndex=int(symbolIndex)
  + if (symbolIndex<0):
  + symbolIndex=0
  + elif (symbolIndex>len(symblArray)-1):
  + symbolIndex=len(symblArray)-1
  + displayArray.append(symblArray[symbolIndex])
  + # if (f>450): # this is so it fits my screen
  + # print("\nYour picture is to big")
  + # print ("F", f, "width", width, "height",height )
  + # break
  + if (f<width-1):
  + printy=(displayArray[j]\*2)
  + print((Fore.GREEN + printy), end="")
  + f+=1
  + else:
  + print ()
  + f=0
  + j+=1