

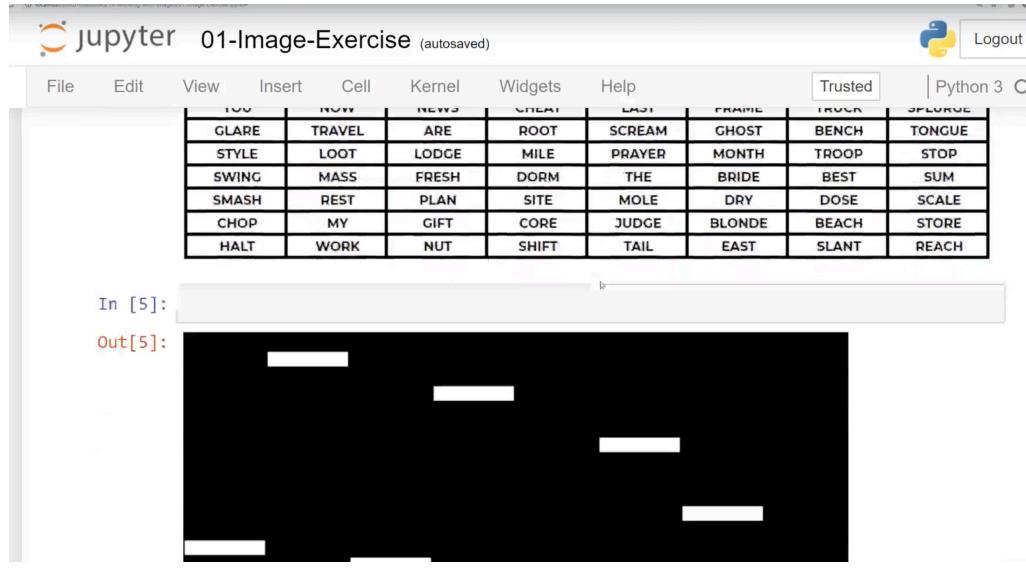
# SECTION 16: WORKING WITH IMAGES WITH PYTHON - 24 minutes, 4 parts

## 3/4 Python Image Exercises -

### Overview

#### • exercise

- -> this is a puzzle
- -> there is a solutions notebook
- -> this is an image exercise
- -> there are two images
- -> the first is a word matrix and the second is a mask
- -> the first is a png which is a spreadsheet and the second is a mask
- -> we are revealing a hidden message in the spreadsheet by stacking one of the images on top of the other





- In this lecture we will explore how to work with Images with Python.
- We will use the Pillow library for this, which is a fork of the PIL (Python Imaging Library) with easy to use function calls.
- We will need to install this additional library.



- Install it at your command line with:
  - **pip install pillow**
- You can find the official documentation for it at:
  - **pillow.readthedocs.io**

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

2x 1:40 / 18:05

This screenshot shows a Jupyter Notebook interface titled "126. Working with Images with Python". The notebook is titled "jupyter 00-Overview-of-Working-with-Images (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The main content area contains text explaining the use of PILLOW for image manipulation and instructions on how to install the Pillow library using pip. It also includes a note about potential warnings when working with images in the notebook. At the bottom, there are standard Jupyter notebook controls like zoom level, time, and a toolbar.

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [ ]:
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [8]: mac.filename  
Out[8]: 'example.jpg'  
In [9]: mac.format_description  
Out[9]: 'JPEG (ISO 10918)'  
In [ ]:
```

126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

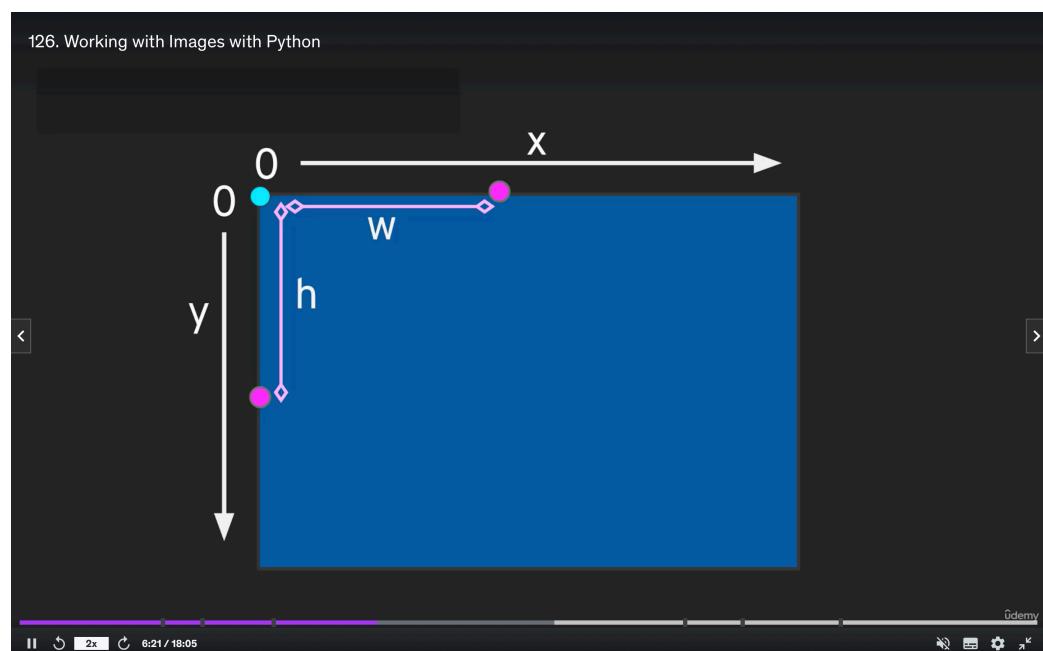
In [9]: `mac.filename`  
Out[9]: 'example.jpg'

In [10]: `mac.format_description`  
Out[10]: 'JPEG (ISO 10918)'

## Cropping Images

In [ ]: `mac.crop()`

In [ ]:



126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

## Cropping Images

In [12]: `mac.crop((0,0,100,100))`  
Out[12]:

In [13]: `pencils= Image.open('pencils.jpg')`  
In [14]: `pencils`  
Out[14]:

126. Working with Images with Python

```
In [18]: # BOTTOM PENCILS
x = 0
y = 1100

w = 1950 / 3
h = 1300
```

```
In [19]: pencils.crop((x,y,w,h))
```

```
Out[19]:
```



In [ ]:

126. Working with Images with Python

```
In [21]: mac.size
```

```
Out[21]: (1993, 1257)
```

```
In [22]: halfway = 1993/2
```

```
In [23]: x = halfway - 200
w = halfway + 200
```

```
In [24]: y = 800
h = 1257
```

```
In [25]: mac.crop((x,y,w,h))
```

```
Out[25]:
```



In [ ]:

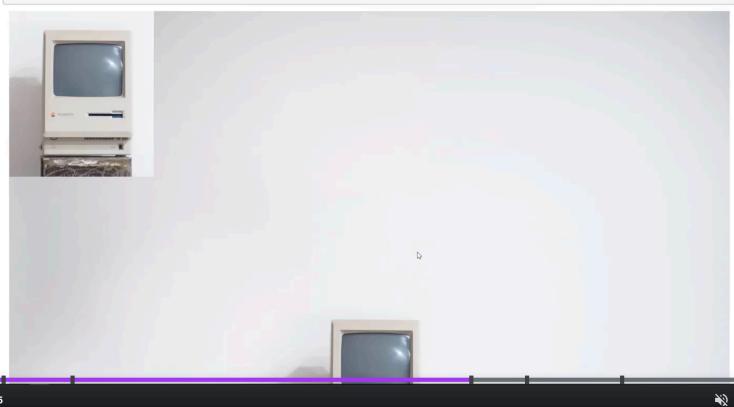
126. Working with Images with Python

```
In [26]: computer = mac.crop((x,y,w,h))
```

```
In [27]: mac.paste(im=computer,box=(0,0))
```

```
In [28]: mac
```

```
Out[28]:
```



In [ ]:

126. Working with Images with Python

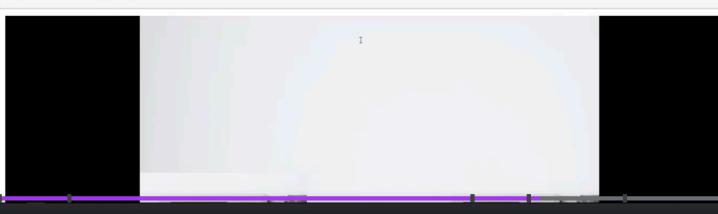
```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [ ]: mac
```

126. Working with Images with Python

```
In [31]: mac.size  
Out[31]: (1993, 1257)  
In [33]: mac.resize((3000,500))  
Out[33]:
```



```
In [34]: mac.rotate(90)  
Out[34]:
```



126. Working with Images with Python

```
File Edit View Insert Cell Kernel Widgets Help Trusted | Python 3
```

## Color Transparency

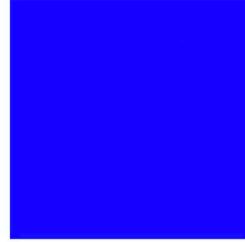
RGBA - Red , Green, Blue, Alpha

```
In [41]: red = Image.open('red_color.jpg')  
In [42]: red  
Out[42]:
```



126. Working with Images with Python

```
In [43]: blue = Image.open('blue_color.png')
In [47]: blue.putalpha(128)
In [48]: blue
Out[48]:
```



```
In [ ]:
```

126. Working with Images with Python

```
In [15]: red.putalpha(128)
In [18]: red
Out[18]:
```



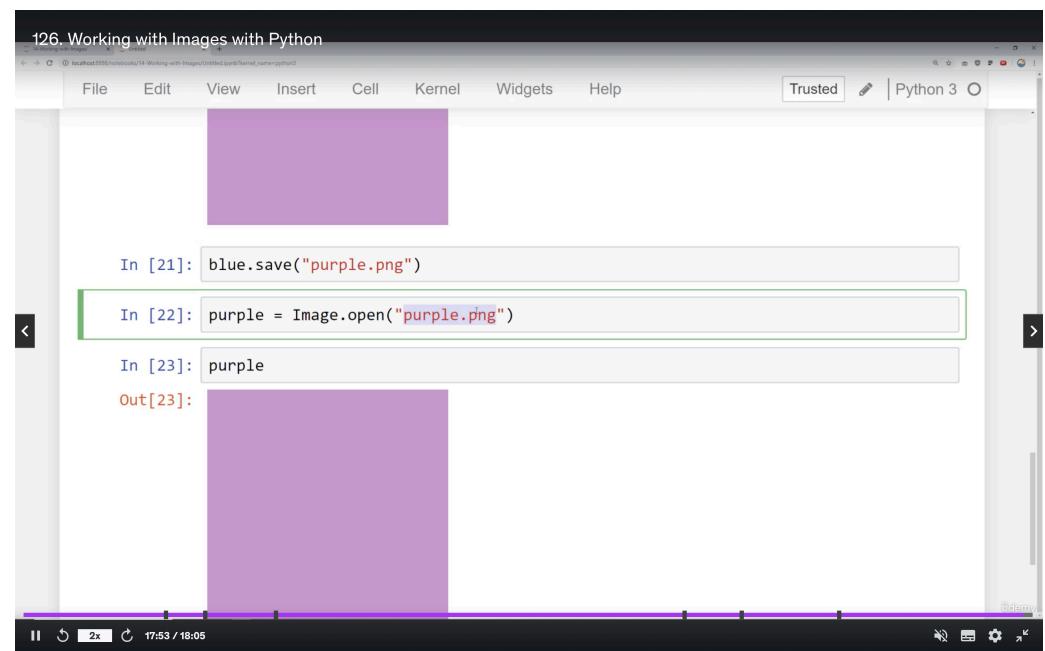
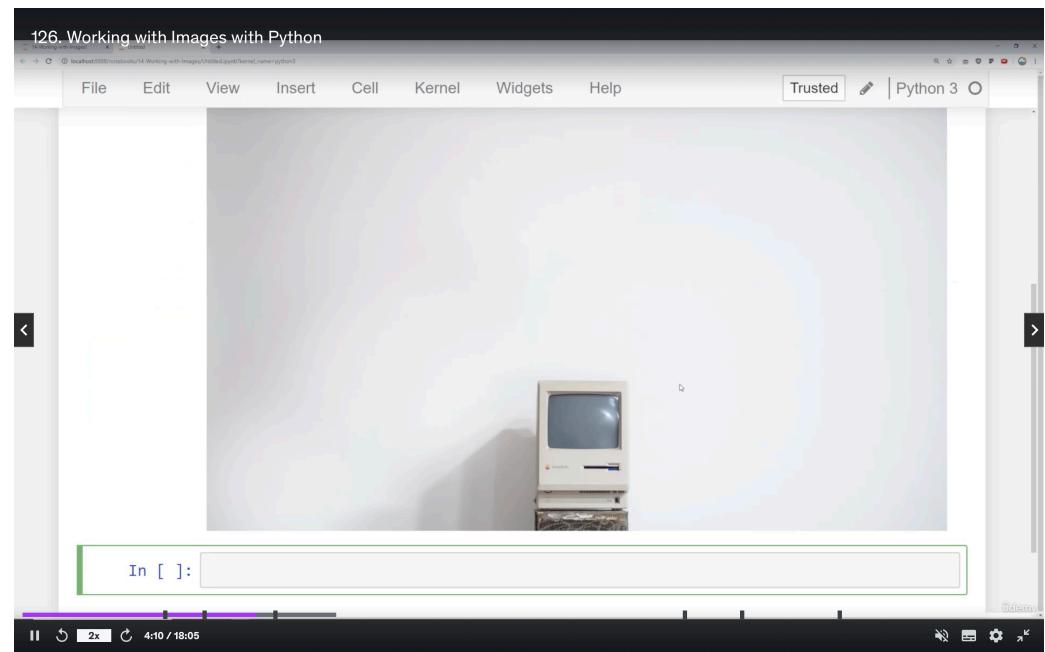
```
In [ ]:
```

126. Working with Images with Python

```
In [19]: blue.paste(im=red,box=(0,0),mask=red)
In [20]: blue
Out[20]:
```



```
In [ ]:
```



TOO	NOW	NEW	CHEAT	END	FRAME	TRUCK	SPURGEON
GLARE	TRAVEL	ARE	ROOT	SCREAM	GHOST	BENCH	TONGUE
STYLE	LOOT	LODGE	MILE	PRAYER	MONTH	TROOP	STOP
SWING	MASS	FRESH	DORM	THE	BRIDE	BEST	SUM
SMASH	REST	PLAN	SITE	MOLE	DRY	DOSE	SCALE
CHOP	MY	GIFT	CORE	JUDGE	BLONDE	BEACH	STORE
HALT	WORK	NUT	SHIFT	TAIL	EAST	SLANT	REACH

In [5]:

out[5]:



In [1]: from PIL import Image

In [2]: words = Image.open('word\_matrix.png')

In [3]: mask = Image.open('mask.png')

In [4]: words

Out[4]:

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT

In [5]: mask

Out[5]:



In [6]: words.paste(mask,(0,0),mask)

In [7]: words

Out[7]:

WOLF	SNATCH	PIER	DAWN
FRONT	TEAM	CLOSED	PITCH
DESK	FADE	NAME	LIST
SPHERE	CHORD	COAST	BOLD
TRAY	PLUCK	DARE	GRIND
TIGHT	FUR	STREAM	SIN
TENSE	WITH	PUMP	BAT
DRAWER	HOUSE	TOUCH	SUN
LOUNGE	SAVE	FOX	WAGE
BRINK	SOW	MAP	OH
DRINK	NOTE	WAVE	IMAGES
TIRE	SOLVE	MAPS	TRACE
CLOSE	POP	MAP	OH
HEAT	EAST	WAVE	IMAGES
SHOT	SOLVE	MAP	TRACE



```
In [6]: words.paste(mask,(0,0),mask)
```

```
In [7]: words
```

```
Out[7]:
```

				WOLF	SNATCH	PIER	DAWN
				FRONT	TEAM	CLOSED	PITCH
				DESK	FADE	NAME	LIST
				SPHERE	CHORD	COAST	BOLD
				TRAY	PLUCK	DARE	GRIND
				TIGHT	FUR	STREAM	SIN
				TENSE	WITH	PUMP	BAT
				DRAWER	HOUSE	TOUCH	SUN
LOUNGE	THE	CLOSE	DUKE	HIKE	SAVE	FOX	WAGE
BRINK	STORE	HEAT	POP	EAST	SOW	MAP	OH

```
In [13]: words.size
```

```
Out[13]: (1015, 559)
```

```
In [14]: mask = mask.resize((1015,559))
```

```
In [15]: mask.size
```

```
Out[15]: (1015, 559)
```

```
In [ ]: mask.putalpha(200)
```

```
In [17]: mask
```

```
Out[17]:
```



```
In [18]: words.paste(mask,(0,0),mask)
```

```
In [19]: words
```

```
Out[19]:
```

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT
FLU	WENT	SOW	QUOTE	DRAWER	HOUSE	TOUCH	SUN





- In this lecture we will explore how to work with Images with Python.
- We will use the Pillow library for this, which is a fork of the PIL (Python Imaging Library) with easy to use function calls.
- We will need to install this additional library.



- Install it at your command line with:
  - **pip install pillow**
- You can find the official documentation for it at:
  - **pillow.readthedocs.io**

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

2x 1:40 / 18:05

This screenshot shows a Jupyter Notebook interface titled "126. Working with Images with Python". The notebook is titled "jupyter 00-Overview-of-Working-with-Images (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The main content area contains text explaining the use of PILLOW for image manipulation and provides instructions for installing the Pillow library via pip. It also includes a note about potential warnings when working with images in the notebook. At the bottom, there are standard Jupyter notebook controls for zooming (2x), time (1:40 / 18:05), and other settings.

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [ ]:
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [8]: mac.filename  
Out[8]: 'example.jpg'  
In [9]: mac.format_description  
Out[9]: 'JPEG (ISO 10918)'  
In [ ]:
```

126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

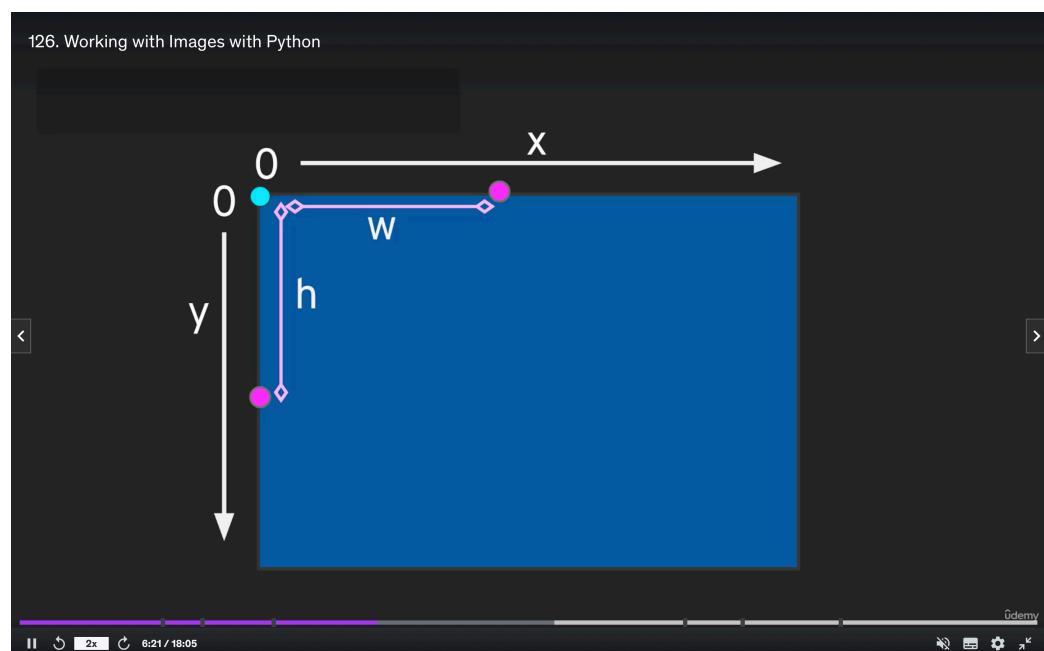
In [9]: `mac.filename`  
Out[9]: 'example.jpg'

In [10]: `mac.format_description`  
Out[10]: 'JPEG (ISO 10918)'

## Cropping Images

In [ ]: `mac.crop()`

In [ ]:



126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

## Cropping Images

In [12]: `mac.crop((0,0,100,100))`  
Out[12]:

In [13]: `pencils= Image.open('pencils.jpg')`  
In [14]: `pencils`  
Out[14]:

126. Working with Images with Python

```
In [18]: # BOTTOM PENCILS
x = 0
y = 1100

w = 1950 / 3
h = 1300
```

```
In [19]: pencils.crop((x,y,w,h))
```

```
Out[19]:
```



In [ ]:

126. Working with Images with Python

```
In [21]: mac.size
```

```
Out[21]: (1993, 1257)
```

```
In [22]: halfway = 1993/2
```

```
In [23]: x = halfway - 200
w = halfway + 200
```

```
In [24]: y = 800
h = 1257
```

```
In [25]: mac.crop((x,y,w,h))
```

```
Out[25]:
```



In [ ]:

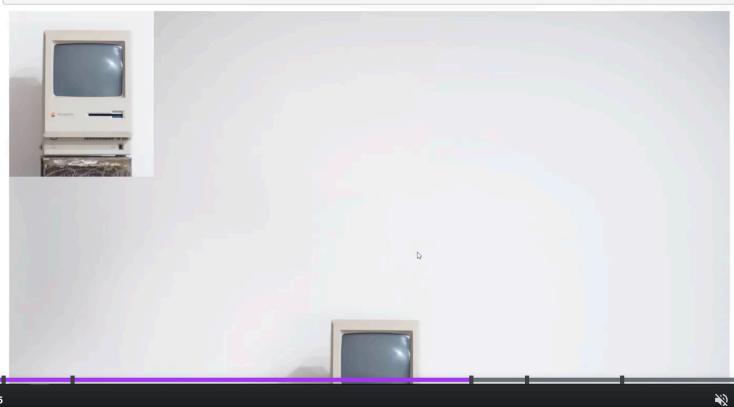
126. Working with Images with Python

```
In [26]: computer = mac.crop((x,y,w,h))
```

```
In [27]: mac.paste(im=computer,box=(0,0))
```

```
In [28]: mac
```

```
Out[28]:
```



In [ ]:

126. Working with Images with Python

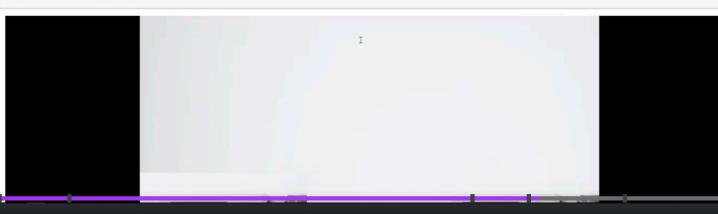
```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [ ]: mac
```

126. Working with Images with Python

```
In [31]: mac.size  
Out[31]: (1993, 1257)  
In [33]: mac.resize((3000,500))  
Out[33]:
```



```
In [34]: mac.rotate(90)  
Out[34]:
```



126. Working with Images with Python

Color Transparency

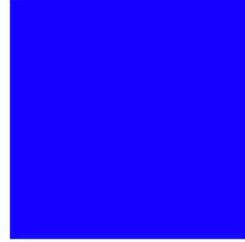
RGBA - Red , Green, Blue, Alpha

```
In [41]: red = Image.open('red_color.jpg')  
In [42]: red  
Out[42]:
```



126. Working with Images with Python

```
In [43]: blue = Image.open('blue_color.png')
In [47]: blue.putalpha(128)
In [48]: blue
Out[48]:
```



```
In [ ]:
```

126. Working with Images with Python

```
In [15]: red.putalpha(128)
In [18]: red
Out[18]:
```



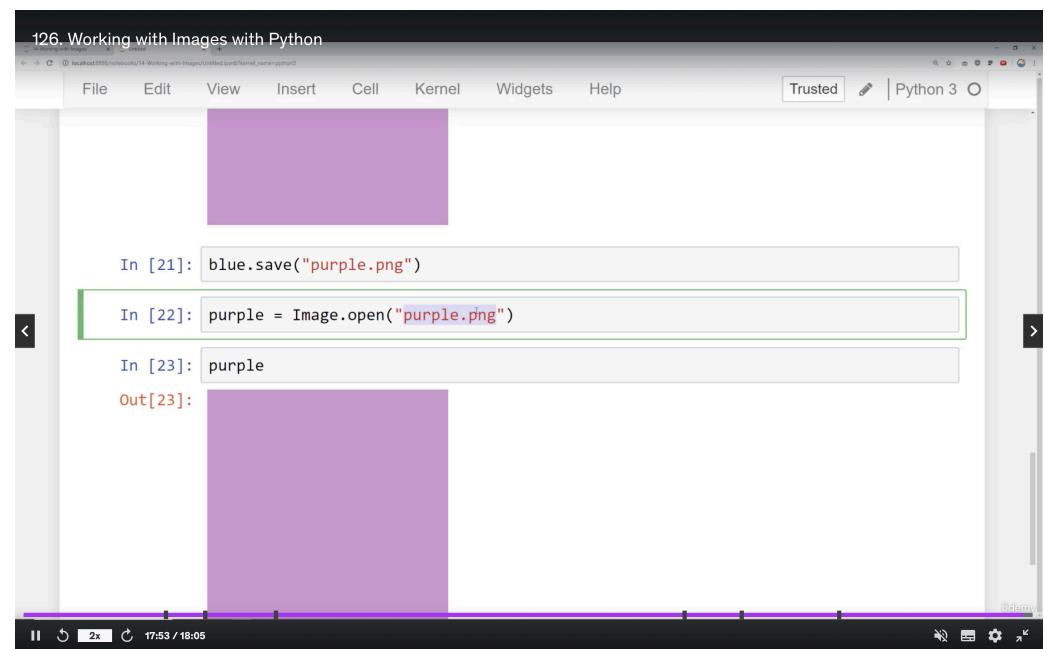
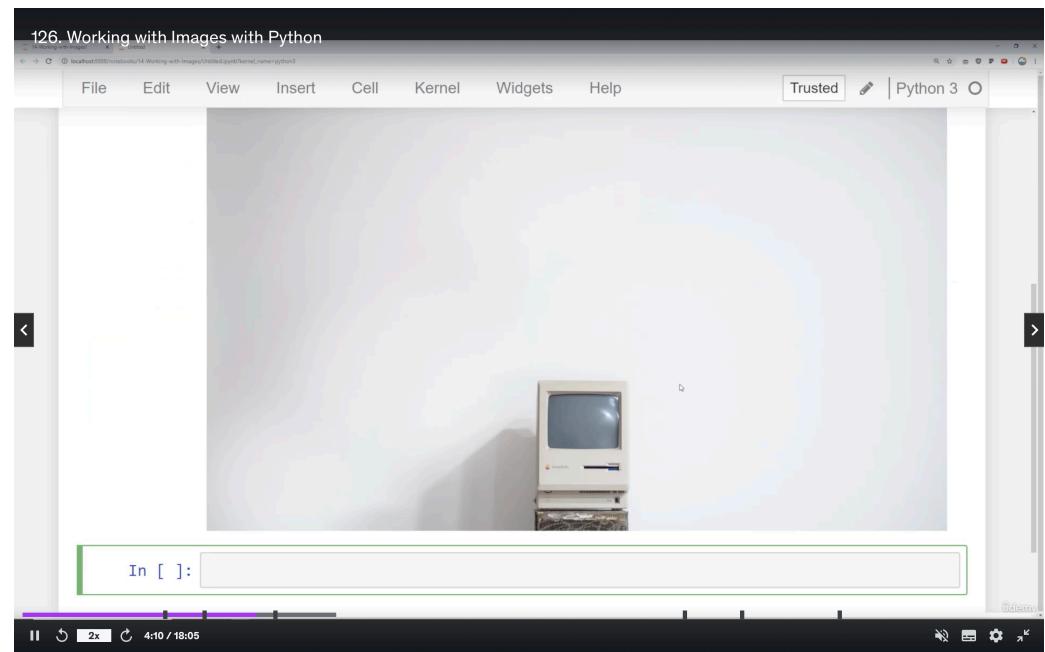
```
In [ ]:
```

126. Working with Images with Python

```
In [19]: blue.paste(im=red,box=(0,0),mask=red)
In [20]: blue
Out[20]:
```



```
In [ ]:
```



TOO	NOW	NEW	CHEAT	END	FRAME	TRUCK	SPURGEON
GLARE	TRAVEL	ARE	ROOT	SCREAM	GHOST	BENCH	TONGUE
STYLE	LOOT	LODGE	MILE	PRAYER	MONTH	TROOP	STOP
SWING	MASS	FRESH	DORM	THE	BRIDE	BEST	SUM
SMASH	REST	PLAN	SITE	MOLE	DRY	DOSE	SCALE
CHOP	MY	GIFT	CORE	JUDGE	BLONDE	BEACH	STORE
HALT	WORK	NUT	SHIFT	TAIL	EAST	SLANT	REACH

In [5]:

out[5]:



In [1]: from PIL import Image

In [2]: words = Image.open('word\_matrix.png')

In [3]: mask = Image.open('mask.png')

In [4]: words

Out[4]:

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT

In [5]: mask

Out[5]:



In [6]: words.paste(mask,(0,0),mask)

In [7]: words

Out[7]:

WOLF	SNATCH	PIER	DAWN
FRONT	TEAM	CLOSED	PITCH
DESK	FADE	NAME	LIST
SPHERE	CHORD	COAST	BOLD
TRAY	PLUCK	DARE	GRIND
TIGHT	FUR	STREAM	SIN
TENSE	WITH	PUMP	BAT
DRAWER	HOUSE	TOUCH	SUN
LOUNGE	SAVE	FOX	WAGE
BRINK	SOW	MAP	OH
DRINK	NOTE	WAVE	IMAGES
TIRE	SOLVE	MAPS	TRACE
CLOSE	SHOT	MAP	OH
DUKE	SOLVE	MAPS	TRACE



```
In [6]: words.paste(mask,(0,0),mask)
```

```
In [7]: words
```

```
Out[7]:
```

				WOLF	SNATCH	PIER	DAWN
				FRONT	TEAM	CLOSED	PITCH
				DESK	FADE	NAME	LIST
				SPHERE	CHORD	COAST	BOLD
				TRAY	PLUCK	DARE	GRIND
				TIGHT	FUR	STREAM	SIN
				TENSE	WITH	PUMP	BAT
				DRAWER	HOUSE	TOUCH	SUN
LOUNGE	THE	CLOSE	DUKE	HIKE	SAVE	FOX	WAGE
BRINK	STORE	HEAT	POP	EAST	SOW	MAP	OH

```
In [13]: words.size
```

```
Out[13]: (1015, 559)
```

```
In [14]: mask = mask.resize((1015,559))
```

```
In [15]: mask.size
```

```
Out[15]: (1015, 559)
```

```
In [ ]: mask.putalpha(200)
```

```
In [17]: mask
```

```
Out[17]:
```



```
In [18]: words.paste(mask,(0,0),mask)
```

```
In [19]: words
```

```
Out[19]:
```

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	Brave	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT
FLU	WENT	SOW	QUOTE	DRAWER	HOUSE	TOUCH	SUN





- In this lecture we will explore how to work with Images with Python.
- We will use the Pillow library for this, which is a fork of the PIL (Python Imaging Library) with easy to use function calls.
- We will need to install this additional library.



- Install it at your command line with:
  - **pip install pillow**
- You can find the official documentation for it at:
  - **pillow.readthedocs.io**

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

2x 1:40 / 18:05

This screenshot shows a Jupyter Notebook interface titled "126. Working with Images with Python". The notebook is titled "jupyter 00-Overview-of-Working-with-Images (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The main content area contains text explaining the use of PILLOW for image manipulation and instructions on how to install the Pillow library using pip. It also includes a note about potential warnings when working with images in the notebook. At the bottom, there are standard Jupyter notebook controls for zooming (2x), time (1:40 / 18:05), and other settings.

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [ ]:
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [8]: mac.filename  
Out[8]: 'example.jpg'  
In [9]: mac.format_description  
Out[9]: 'JPEG (ISO 10918)'  
In [ ]:
```

126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

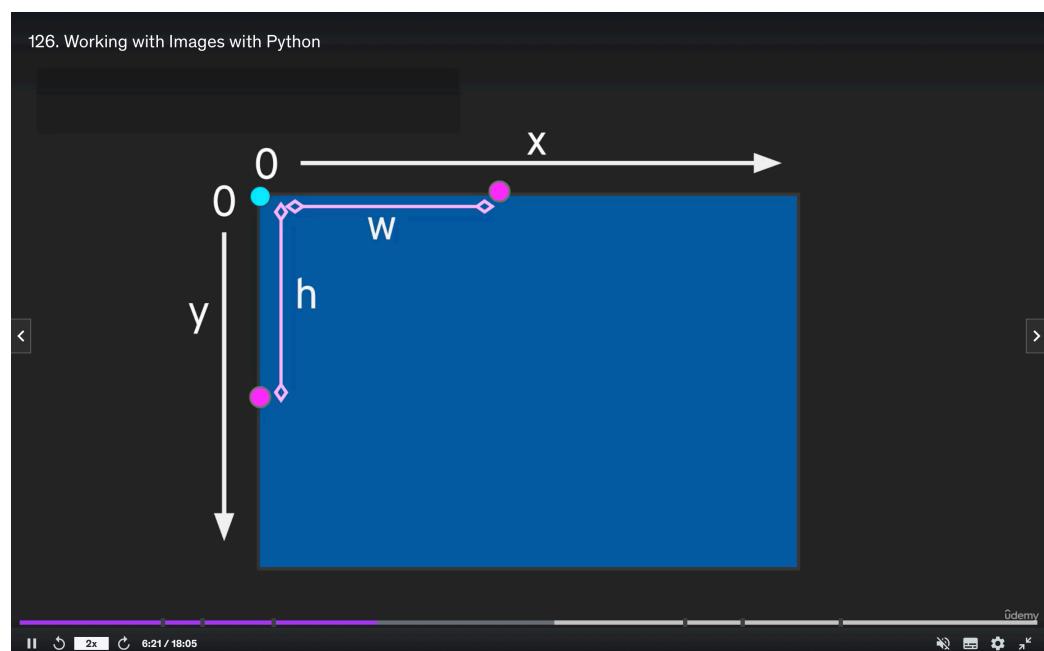
In [9]: `mac.filename`  
Out[9]: 'example.jpg'

In [10]: `mac.format_description`  
Out[10]: 'JPEG (ISO 10918)'

## Cropping Images

In [ ]: `mac.crop()`

In [ ]:



126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

## Cropping Images

In [12]: `mac.crop((0,0,100,100))`  
Out[12]:

In [13]: `pencils= Image.open('pencils.jpg')`  
In [14]: `pencils`  
Out[14]:

126. Working with Images with Python

```
In [18]: # BOTTOM PENCILS
x = 0
y = 1100

w = 1950 / 3
h = 1300
```

```
In [19]: pencils.crop((x,y,w,h))
```

```
Out[19]:
```



In [ ]:

126. Working with Images with Python

```
In [21]: mac.size
```

```
Out[21]: (1993, 1257)
```

```
In [22]: halfway = 1993/2
```

```
In [23]: x = halfway - 200
w = halfway + 200
```

```
In [24]: y = 800
h = 1257
```

```
In [25]: mac.crop((x,y,w,h))
```

```
Out[25]:
```



In [ ]:

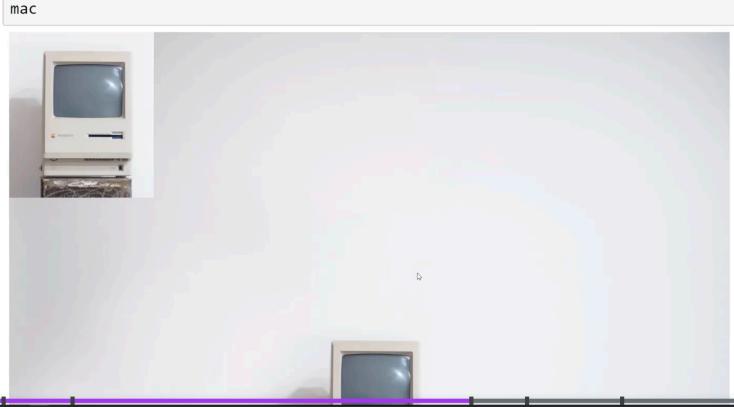
126. Working with Images with Python

```
In [26]: computer = mac.crop((x,y,w,h))
```

```
In [27]: mac.paste(im=computer,box=(0,0))
```

```
In [28]: mac
```

```
Out[28]:
```



In [ ]:

126. Working with Images with Python

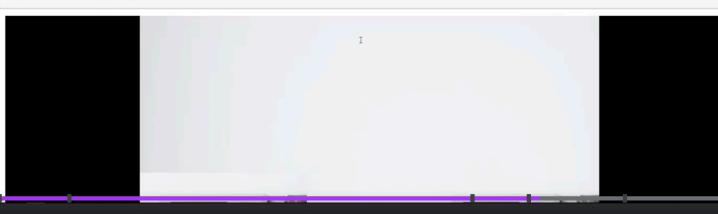
```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [ ]: mac
```

126. Working with Images with Python

```
In [31]: mac.size  
Out[31]: (1993, 1257)  
In [33]: mac.resize((3000,500))  
Out[33]:
```



```
In [34]: mac.rotate(90)  
Out[34]:
```



126. Working with Images with Python

Color Transparency

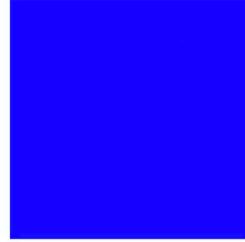
RGBA - Red , Green, Blue, Alpha

```
In [41]: red = Image.open('red_color.jpg')  
In [42]: red  
Out[42]:
```



126. Working with Images with Python

```
In [43]: blue = Image.open('blue_color.png')
In [47]: blue.putalpha(128)
In [48]: blue
Out[48]:
```



```
In [ ]:
```

126. Working with Images with Python

```
In [15]: red.putalpha(128)
In [18]: red
Out[18]:
```



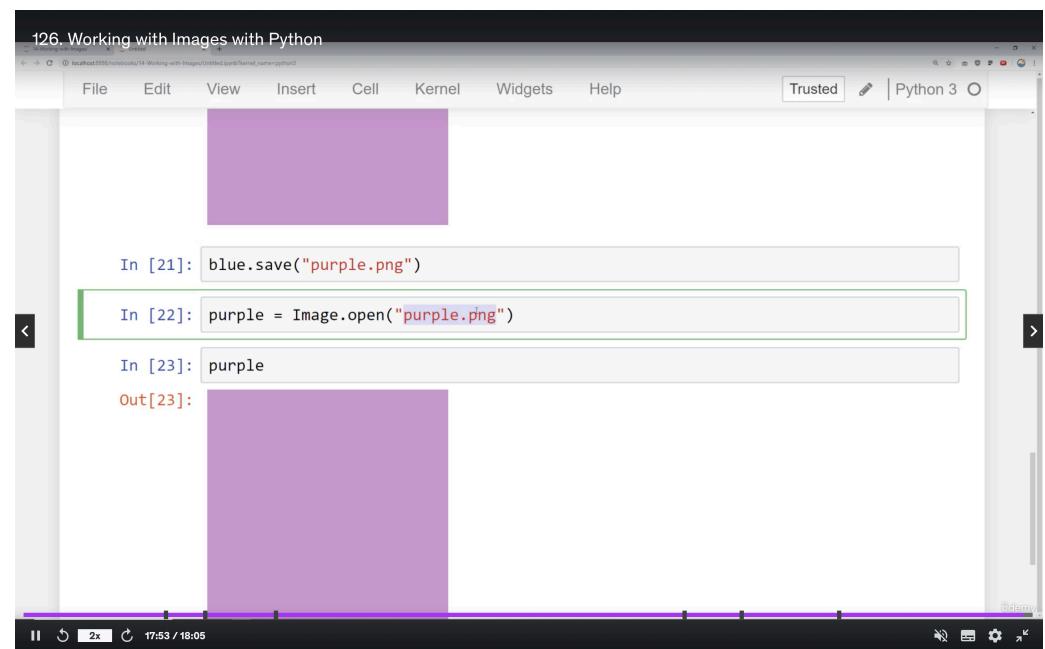
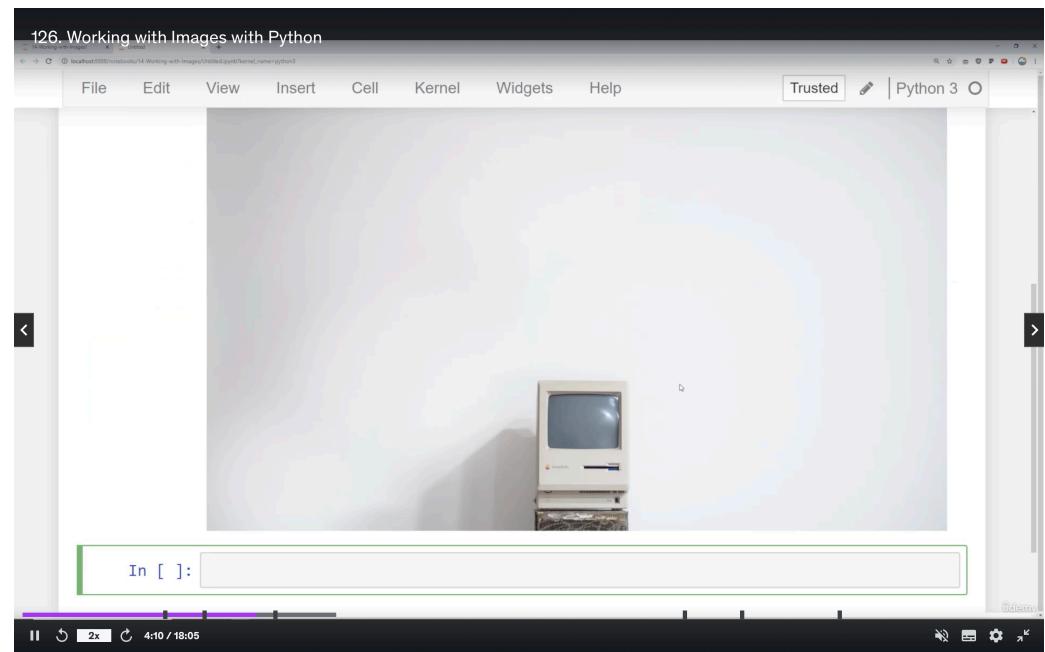
```
In [ ]:
```

126. Working with Images with Python

```
In [19]: blue.paste(im=red,box=(0,0),mask=red)
In [20]: blue
Out[20]:
```



```
In [ ]:
```



TOO	NOW	NEW	CHEAT	END	FRAME	TRUCK	SPURGEON
GLARE	TRAVEL	ARE	ROOT	SCREAM	GHOST	BENCH	TONGUE
STYLE	LOOT	LODGE	MILE	PRAYER	MONTH	TROOP	STOP
SWING	MASS	FRESH	DORM	THE	BRIDE	BEST	SUM
SMASH	REST	PLAN	SITE	MOLE	DRY	DOSE	SCALE
CHOP	MY	GIFT	CORE	JUDGE	BLONDE	BEACH	STORE
HALT	WORK	NUT	SHIFT	TAIL	EAST	SLANT	REACH

In [5]:

out[5]:



In [1]: from PIL import Image

In [2]: words = Image.open('word\_matrix.png')

In [3]: mask = Image.open('mask.png')

In [4]: words

Out[4]:

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT

In [5]: mask

Out[5]:



In [6]: words.paste(mask,(0,0),mask)

In [7]: words

Out[7]:

WOLF	SNATCH	PIER	DAWN
FRONT	TEAM	CLOSED	PITCH
DESK	FADE	NAME	LIST
SPHERE	CHORD	COAST	BOLD
TRAY	PLUCK	DARE	GRIND
TIGHT	FUR	STREAM	SIN
TENSE	WITH	PUMP	BAT
DRAWER	HOUSE	TOUCH	SUN
LOUNGE	SAVE	FOX	WAGE
BRINK	SOW	MAP	OH
DRINK	NOTE	WAVE	IMAGES
TIRE	SOLVE	MAPS	TRACE
CLOSE	POP	MAP	OH
HEAT	EAST	WAVE	IMAGES
SHOT	SOLVE	MAP	TRACE



```
In [6]: words.paste(mask,(0,0),mask)
```

```
In [7]: words
```

```
Out[7]:
```

				WOLF	SNATCH	PIER	DAWN
				FRONT	TEAM	CLOSED	PITCH
				DESK	FADE	NAME	LIST
				SPHERE	CHORD	COAST	BOLD
				TRAY	PLUCK	DARE	GRIND
				TIGHT	FUR	STREAM	SIN
				TENSE	WITH	PUMP	BAT
				DRAWER	HOUSE	TOUCH	SUN
LOUNGE	THE	CLOSE	DUKE	HIKE	SAVE	FOX	WAGE
BRINK	STORE	HEAT	POP	EAST	SOW	MAP	OH

```
In [13]: words.size
```

```
Out[13]: (1015, 559)
```

```
In [14]: mask = mask.resize((1015,559))
```

```
In [15]: mask.size
```

```
Out[15]: (1015, 559)
```

```
In [ ]: mask.putalpha(200)
```

```
In [17]: mask
```

```
Out[17]:
```



```
In [18]: words.paste(mask,(0,0),mask)
```

```
In [19]: words
```

```
Out[19]:
```

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT
FLU	WENT	SOW	QUOTE	DRAWER	HOUSE	TOUCH	SUN





- In this lecture we will explore how to work with Images with Python.
- We will use the Pillow library for this, which is a fork of the PIL (Python Imaging Library) with easy to use function calls.
- We will need to install this additional library.



- Install it at your command line with:
  - **pip install pillow**
- You can find the official documentation for it at:
  - **pillow.readthedocs.io**

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

2x 1:40 / 18:05

This screenshot shows a Jupyter Notebook interface titled "126. Working with Images with Python". The notebook is titled "jupyter 00-Overview-of-Working-with-Images (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, Help, Trusted, and Python 3. The main content area contains text explaining the use of PILLOW for image manipulation and provides instructions for installing the Pillow library via pip. It also includes a note about potential warnings when working with images in the notebook. At the bottom, there are standard Jupyter notebook controls for zooming (2x), time (1:40 / 18:05), and other settings.

126. Working with Images with Python

jupyter 00-Overview-of-Working-with-Images (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

By leveraging the power of some common libraries that you can install, such as PILLOW, Python gains the ability to work with and manipulate images for simple tasks. You can install Pillow by running:

```
pip install pillow
```

In case of any issues, you can refer to their [official documentation](#) on installation. But for most computers, the simple pip install method should work.

Note: When working with images in the jupyter notebook, you may get the following warning:

```
IOPub data rate exceeded.  
The notebook server will temporarily stop sending output  
to the client in order to avoid crashing it.  
To change this limit, set the config variable  
`--NotebookApp.iopub_data_rate_limit`.
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [ ]:
```

126. Working with Images with Python

localhost:8888/notebooks/14%20-Working%20with%20Images/Untitled.ipynb?kernel\_name=python3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [7]: mac.size  
Out[7]: (1993, 1257)  
In [8]: mac.filename  
Out[8]: 'example.jpg'  
In [9]: mac.format_description  
Out[9]: 'JPEG (ISO 10918)'  
In [ ]:
```

126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

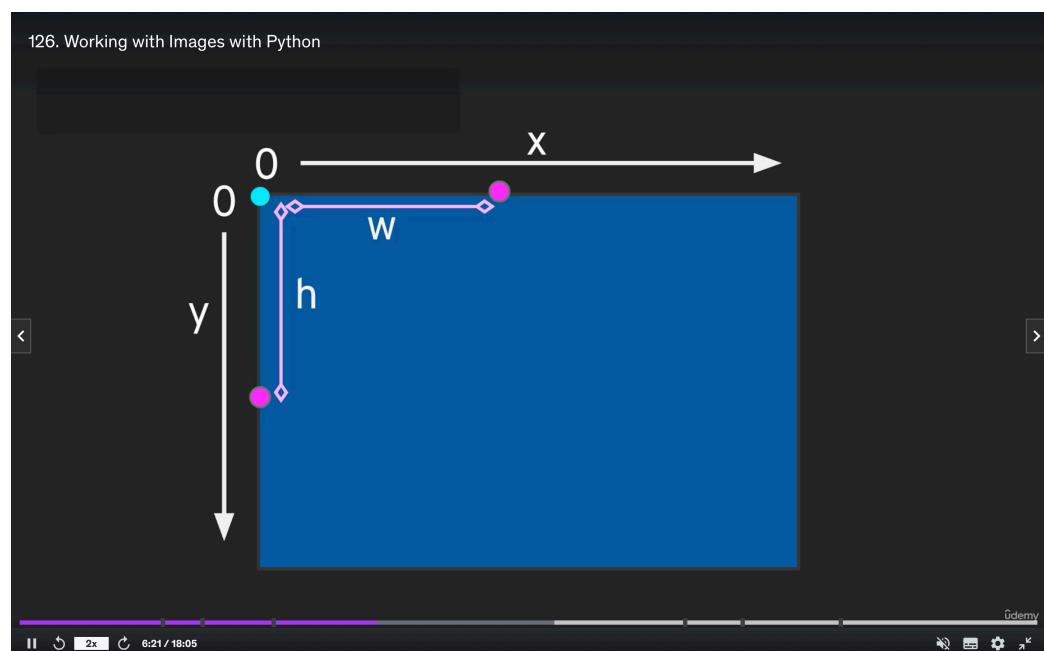
In [9]: `mac.filename`  
Out[9]: 'example.jpg'

In [10]: `mac.format_description`  
Out[10]: 'JPEG (ISO 10918)'

## Cropping Images

In [ ]: `mac.crop()`

In [ ]:



126. Working with Images with Python

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

## Cropping Images

In [12]: `mac.crop((0,0,100,100))`  
Out[12]:

In [13]: `pencils= Image.open('pencils.jpg')`  
In [14]: `pencils`  
Out[14]:

126. Working with Images with Python

```
In [18]: # BOTTOM PENCILS
x = 0
y = 1100

w = 1950 / 3
h = 1300
```

```
In [19]: pencils.crop((x,y,w,h))
```

```
Out[19]:
```



In [ ]:

126. Working with Images with Python

```
In [21]: mac.size
```

```
Out[21]: (1993, 1257)
```

```
In [22]: halfway = 1993/2
```

```
In [23]: x = halfway - 200
w = halfway + 200
```

```
In [24]: y = 800
h = 1257
```

```
In [25]: mac.crop((x,y,w,h))
```

```
Out[25]:
```



In [ ]:

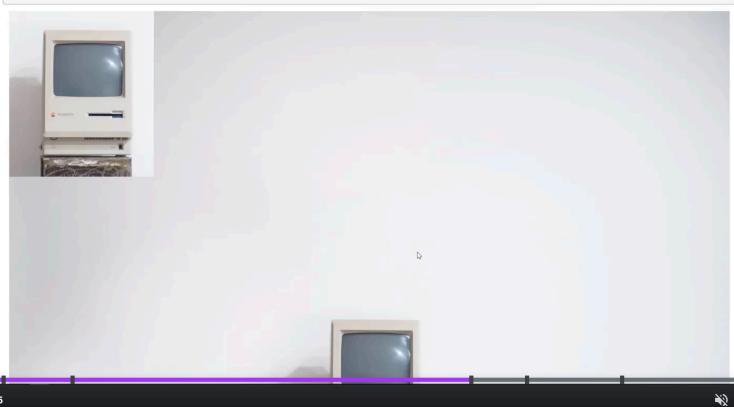
126. Working with Images with Python

```
In [26]: computer = mac.crop((x,y,w,h))
```

```
In [27]: mac.paste(im=computer,box=(0,0))
```

```
In [28]: mac
```

```
Out[28]:
```



In [ ]:

126. Working with Images with Python

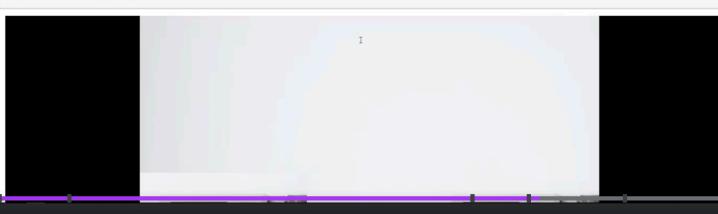
```
In [3]: from PIL import Image  
In [4]: mac = Image.open('example.jpg')  
In [5]: type(mac)  
Out[5]: PIL.JpegImagePlugin.JpegImageFile  
In [ ]: mac
```

126. Working with Images with Python

```
In [31]: mac.size  
Out[31]: (1993, 1257)  
In [33]: mac.resize((3000,500))  
Out[33]:
```



```
In [34]: mac.rotate(90)  
Out[34]:
```



126. Working with Images with Python

Color Transparency

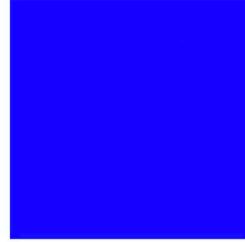
RGBA - Red , Green, Blue, Alpha

```
In [41]: red = Image.open('red_color.jpg')  
In [42]: red  
Out[42]:
```



126. Working with Images with Python

```
In [43]: blue = Image.open('blue_color.png')
In [47]: blue.putalpha(128)
In [48]: blue
Out[48]:
```



```
In [ ]:
```

126. Working with Images with Python

```
In [15]: red.putalpha(128)
In [18]: red
Out[18]:
```



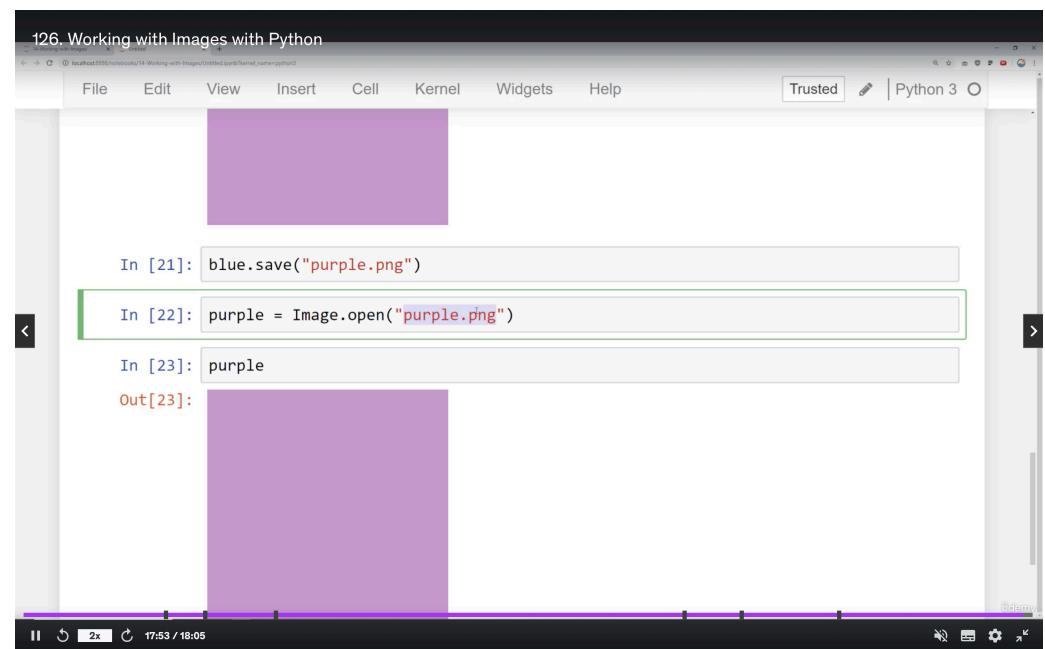
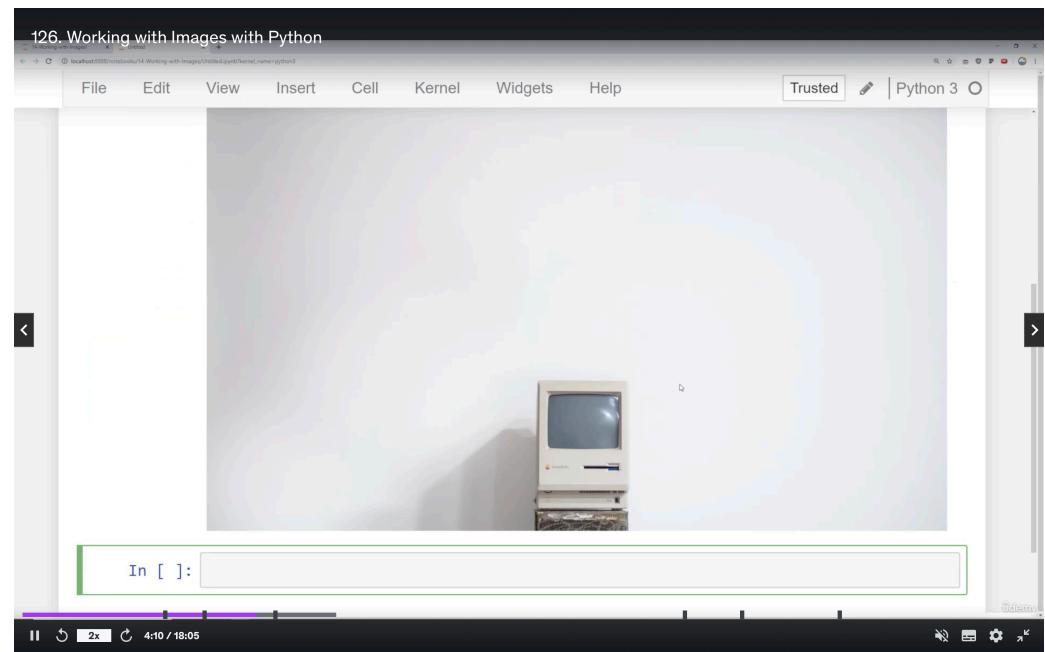
```
In [ ]:
```

126. Working with Images with Python

```
In [19]: blue.paste(im=red,box=(0,0),mask=red)
In [20]: blue
Out[20]:
```



```
In [ ]:
```



TOO	NOW	NEW	CHEAT	END	FRAME	TRUCK	SPURGEON
GLARE	TRAVEL	ARE	ROOT	SCREAM	GHOST	BENCH	TONGUE
STYLE	LOOT	LODGE	MILE	PRAYER	MONTH	TROOP	STOP
SWING	MASS	FRESH	DORM	THE	BRIDE	BEST	SUM
SMASH	REST	PLAN	SITE	MOLE	DRY	DOSE	SCALE
CHOP	MY	GIFT	CORE	JUDGE	BLONDE	BEACH	STORE
HALT	WORK	NUT	SHIFT	TAIL	EAST	SLANT	REACH

In [5]:

out[5]:



In [1]: from PIL import Image

In [2]: words = Image.open('word\_matrix.png')

In [3]: mask = Image.open('mask.png')

In [4]: words

Out[4]:

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT

In [5]: mask

Out[5]:



In [6]: words.paste(mask,(0,0),mask)

In [7]: words

Out[7]:

WOLF	SNATCH	PIER	DAWN
FRONT	TEAM	CLOSED	PITCH
DESK	FADE	NAME	LIST
SPHERE	CHORD	COAST	BOLD
TRAY	PLUCK	DARE	GRIND
TIGHT	FUR	STREAM	SIN
TENSE	WITH	PUMP	BAT
DRAWER	HOUSE	TOUCH	SUN
LOUNGE	SAVE	FOX	WAGE
BRINK	SOW	MAP	OH
DRINK	NOTE	WAVE	IMAGES
TIRE	SOLVE	MAPS	TRACE
CLOSE	POP	MAP	OH
HEAT	EAST	WAVE	IMAGES
SHOT	SOLVE	MAP	TRACE



```
In [6]: words.paste(mask,(0,0),mask)
```

```
In [7]: words
```

```
Out[7]:
```

				WOLF	SNATCH	PIER	DAWN
				FRONT	TEAM	CLOSED	PITCH
				DESK	FADE	NAME	LIST
				SPHERE	CHORD	COAST	BOLD
				TRAY	PLUCK	DARE	GRIND
				TIGHT	FUR	STREAM	SIN
				TENSE	WITH	PUMP	BAT
				DRAWER	HOUSE	TOUCH	SUN
LOUNGE	THE	CLOSE	DUKE	HIKE	SAVE	FOX	WAGE
BRINK	STORE	HEAT	POP	EAST	SOW	MAP	OH

```
In [13]: words.size
```

```
Out[13]: (1015, 559)
```

```
In [14]: mask = mask.resize((1015,559))
```

```
In [15]: mask.size
```

```
Out[15]: (1015, 559)
```

```
In [ ]: mask.putalpha(200)
```

```
In [17]: mask
```

```
Out[17]:
```



```
In [18]: words.paste(mask,(0,0),mask)
```

```
In [19]: words
```

```
Out[19]:
```

STOP	FAIR	SCORE	FIELD	WOLF	SNATCH	PIER	DAWN
WOOD	GREAT	MAID	STRONG	FRONT	TEAM	CLOSED	PITCH
HOLD	CURL	BRAVE	SPITE	DESK	FADE	NAME	LIST
FALL	HILL	TREE	WORK	SPHERE	CHORD	COAST	BOLD
YARD	LAND	CHURCH	LATE	TRAY	PLUCK	DARE	GRIND
FIGHT	MY	PAUSE	DOG	TIGHT	FUR	STREAM	SIN
CALF	HIKE	DASH	FLOOD	TENSE	WITH	PUMP	BAT
FLU	WENT	SOW	QUOTE	DRAWER	HOUSE	TOUCH	SUN