# *The Python Imaging Library (PIL)*

**Python code for processing images**

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Create from scratch**

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Create from scratch**
  - **Read from an image file**

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Create from scratch**
  - **Read from an image file**
  - **Process the image's pixels**

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Create from scratch**
  - **Read from an image file**
  - **Process the image's pixels**
  - **Write an image file**

# *The Python Imaging Library (PIL)*

## Python modules and the PIL

# *The Python Imaging Library (PIL)*

## Python modules and the PIL

### A module is a file of Python code.

# *The Python Imaging Library (PIL)*

## Python modules and the PIL

- A module is a file of Python code.
- You "import" a module to use its code.

# *The Python Imaging Library (PIL)*

## Python modules and the PIL

- A module is a file of Python code.
- You "import" a module to use its code.
- The names in a module are preceded by the module name and a period character.

# *The Python Imaging Library (PIL)*

**Python modules and the PIL**

**A module is a file of Python code.**

**You "import" a module to use its code.**

**The names in a module are preceded by the module name and a period character.**

***For example, the "new" function***

# *The Python Imaging Library (PIL)*

**Python modules and the PIL**

**A module is a file of Python code.**

**You "import" a module to use its code.**

**The names in a module are preceded by the module name and a period character.**

***For example, the "new" function in the "Image" module***

# *The Python Imaging Library (PIL)*

## Python modules and the PIL

A module is a file of Python code.

You "import" a module to use its code.

The names in a module are preceded by the module name and a period character.

*For example, the "new" function in the "Image" module is called "Image.new"*

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Create from scratch**

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Create from scratch

```
import Image
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Create from scratch

```
import Image
width = 200
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Create from scratch

```
import Image
width = 200
height = 200
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Create from scratch

```
import Image
width = 200
height = 200
image = Image.new('RGB', [width, height])
```

# *The Python Imaging Library (PIL)*

**Python code for processing images**

**Create from scratch**

```
import Image
width = 200
height = 200
image = Image.new('RGB', [width, height])
```

***We're using the "new" function in the "Image" module.***

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Read from an image file**

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Read from an image file

```
import Image
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Read from an image file

```
import Image
input_image = Image.open("mantis.tif")
```

# *The Python Imaging Library (PIL)*

**Python code for processing images**

**Read from an image file**

```
import Image
input_image = Image.open("mantis.tif")
```

***We're using the "open" function in the "Image" module.***

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

### Making an instance of a class looks like a function call from a function in a module.

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

### Making an instance of a class looks like a function call from a function in a module.

### *A function in a module:*

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

**Making an instance of a class looks like a function call from a function in a module.**

***A function in a module:***

```
image = Image.new('RGB', [width, height])
```

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

**Making an instance of a class looks like a function call from a function in a module.**

***A function in a module:***

```
image = Image.new('RGB', [width, height])
```

***Making a class instance:***

# *The Python Imaging Library (PIL)*

## Classes in the Python Imaging Library

Making an instance of a class looks like a function call from a function in a module.

*A function in a module:*

```
image = Image.new('RGB', [width, height])
```

*Making a class instance:*

```
draw = ImageDraw.Draw(image)
```

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
  - **Process the image's pixels**

# *The Python Imaging Library (PIL)*

- **Python code for processing images**
- **Process the image's pixels**

```
import Image, ImageDraw
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
# "draw" is an instance of class "Draw"
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
# "draw" is an instance of class "Draw"
for y in range(height):
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
# "draw" is an instance of class "Draw"
for y in range(height):
  for x in range(width):
```

# *The Python Imaging Library (PIL)*

## Python code for processing images

### Process the image's pixels

```
import Image, ImageDraw
width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
# "draw" is an instance of class "Draw"
for y in range(height):
  for x in range(width):
    draw.point((x,y), 'red')
```

# *The Python Imaging Library (PIL)*

## Modules and classes

# *The Python Imaging Library (PIL)*

## Modules and classes

A module is a collection of Python code.

# *The Python Imaging Library (PIL)*

**Modules and classes**

**A module is a collection of Python code.**

**Names from a module are preceded by the module name:**

# *The Python Imaging Library (PIL)*

## Modules and classes

A module is a collection of Python code.

Names from a module are preceded by the module name:

```
input_image = Image.open("mantis.tif")
```

# *The Python Imaging Library (PIL)*

## Modules and classes

A module is a collection of Python code.

Names from a module are preceded by the module name:

```
input_image = Image.open("mantis.tif")
```

Names in a class are preceded by the class name:

# *The Python Imaging Library (PIL)*

## Modules and classes

A module is a collection of Python code.

Names from a module are preceded by the module name:

```
input_image = Image.open("mantis.tif")
```

Names in a class are preceded by the class name:

```
draw.point((x,y), 'red')
```

# *The Python Imaging Library (PIL)*

## Today's tutorials

# *The Python Imaging Library (PIL)*

**Today's tutorials**

**A set of Python scripts you can modify.**

# *The Python Imaging Library (PIL)*

**Today’s tutorials**

- **A set of Python scripts you can modify.**
- **The scripts use a module called "putil".**

# *The Python Imaging Library (PIL)*

**Today’s tutorials**

- **A set of Python scripts you can modify.**
- **The scripts use a module called "putil".**
- **Download and modify pictures from the Web.**

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

# *The Python Imaging Library (PIL)*

## A script that makes an image of a single color

```
import Image, ImageDraw, putil
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
color = putil.rgb(.4, .5, .6)
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
color = putil.rgb(.4, .5, .6)

for y in range(height):
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
color = putil.rgb(.4, .5, .6)

for y in range(height):
  for x in range(width):
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
color = putil.rgb(.4, .5, .6)

for y in range(height):
  for x in range(width):
    draw.point((x,y), color)
```

# *The Python Imaging Library (PIL)*

**A script that makes an image of a single color**

```
import Image, ImageDraw, putil

width = 200
height = 200
image = Image.new('RGB', [width, height])
draw = ImageDraw.Draw(image)
color = putil.rgb(.4, .5, .6)

for y in range(height):
  for x in range(width):
    draw.point((x,y), color)

image.save("color.tif")
```