Images

Let’s make em

Housekeeping

- We are going very fast!

- Our first review session is Wednesday Jul 5 in SkillAud - please attend!

- Happy day off!

Today

- **Variables recap**
  - **Also a first look at numbers in Python**

- Introducing: Images
  - Images as a variable
  - for loop and range() function
  - Editing images

A note

- We are going very fast!

- It is okay if you don't understand everything, that's why we have section, review sessions, office hours and LAIR

- Don't struggle through lecture/homework questions alone! Come to LAIR and OH - we love explaining things!

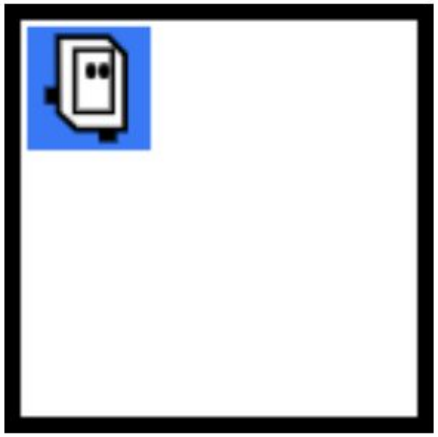
- We go this fast through week 4, then you know enough Python syntax that everything calms down quite a Bit

Recap: Variables

- Variables are something that hold a value, and you can update that value as much as you want
- Here is the syntax for making a variable:

```
var_name = #something
```

```
def variable_example(filename):
    bit = Bit(filename)
    color = bit.get_color()
```

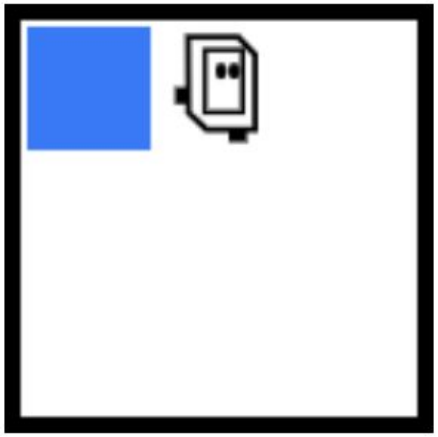


On this map, `color` has value `'blue'`

Recap: Variables

- We can change the value of variables

```
def variable_example(filename):
    bit = Bit(filename)
    color = bit.get_color()
    bit.move()
    color = 'red'
    bit.paint(color)
```

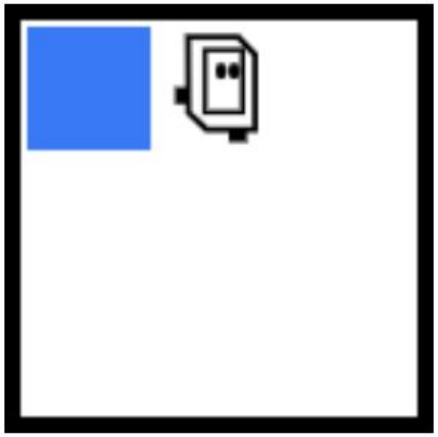


`color` still has value `'blue'`

Recap: Variables

- We can change the value of variables

```
def variable_example(filename):
    bit = Bit(filename)
    color = bit.get_color()
    bit.move()
    color = 'red'
    bit.paint(color)
```

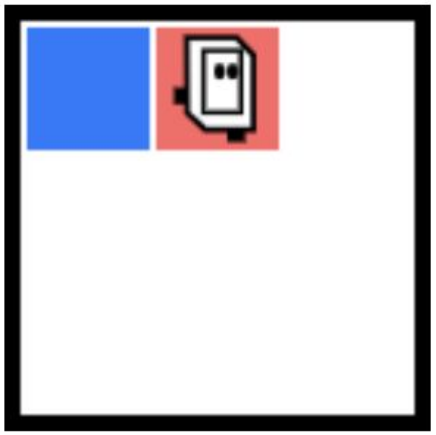


`color` now has value `'red'`

Recap: Variables

- We can change the value of variables

```
def variable_example(filename):
    bit = Bit(filename)
    color = bit.get_color()
    bit.move()
    color = 'red'
    bit.paint(color)
```



`color` now has value `'red'`

So bit paints red!

Variables as numbers

- We haven't used numbers much yet, but Python is *very good* with numbers

```
def variable_num_example(filename):
    lecture_days = 3
    weeks_in_q = 8
    if go_to_wednesday_review():
        lecture_days = 4
    total_lectures = lecture_days * weeks_in_q
    # 24 if you don't go to wednesday review
    # 32 if you do go to wednesday review
```

Aside: Math is as expected in Python

```
def variable_num_example(filename):
    x = 3
    y = 8
    sum = x + y # sum is 11
    prod = x * y # prod is 24
    diff = x - y # diff is -5
```

Aside: Math is as expected in Python

```
def variable_num_example(filename):
    x = 3
    y = 8
    sum = x + y # sum is 11
    prod = x * y # prod is 24
    diff = x - y # diff is -5

    # regular division
    quotient = y / x # quotient is 2.6666

    # int division
    int_quotioent = y // x
    # int_quotient is 2, // truncates decimal
```

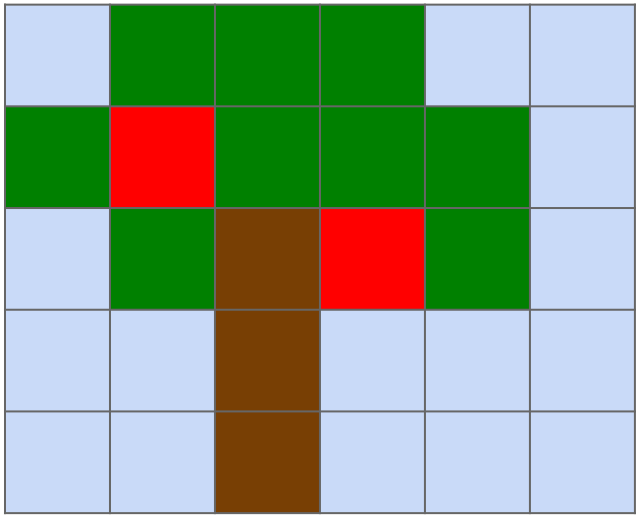
Today

- ~~Variables recap~~
  - ~~Also a first look at numbers in Python~~

- **Introducing: Images**
  - **Images as a variable**
  - **for loop and range() function**
  - **Editing images**

Introducing Images

- On computers, images are represented as grids of "pixels"

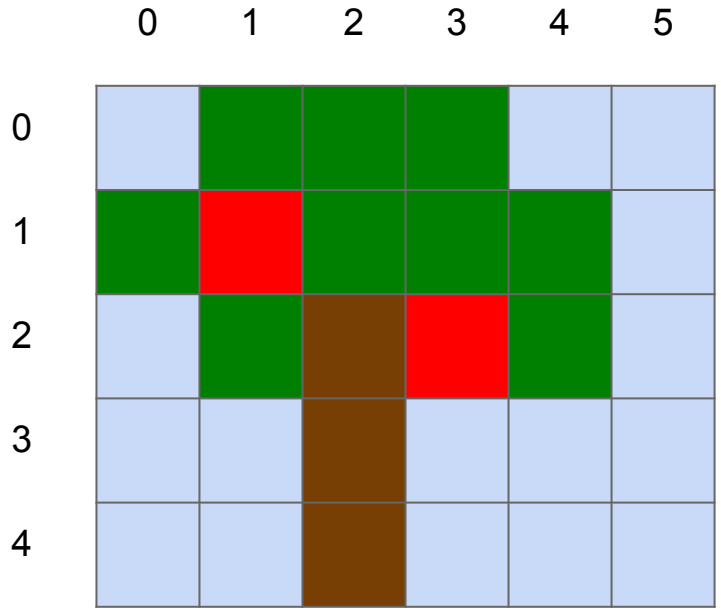


- More pixels = higher resolution

- A "pixel" is usually represented as a tuple of 3 numbers: the red, green and blue values (RGB), which make up the color of that pixel. RGB values are 0-255

Pixel details

- We can locate pixels by their (x, y) coordinates
- An RGB value of (0, 0, 0) is black, (255, 255, 255) is white
- Every other color is in between. Lower = darker



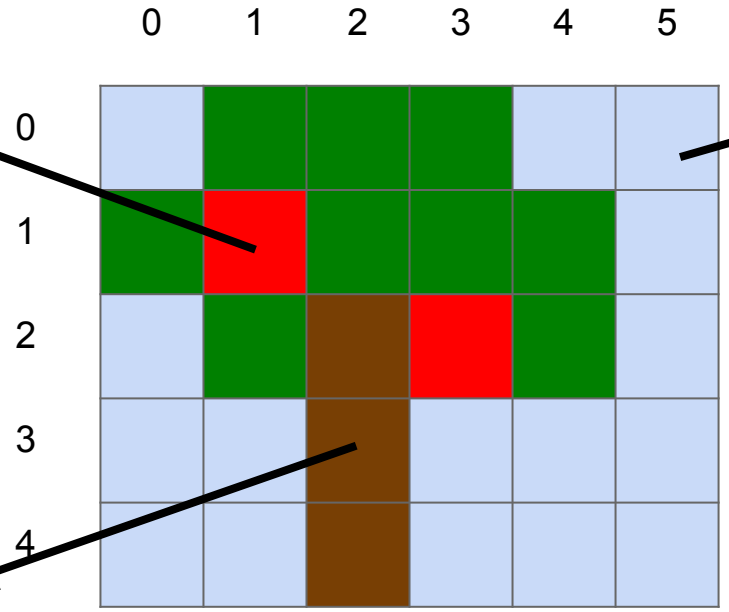
Pixel details

- We can locate pixels by their (x, y) coordinates
- An RGB value of (0, 0, 0) is black, (255, 255, 255) is white
- Higher value = more of that color mixed in, lighter

Pixel **(1, 1)** has value:

(200, 0, 0)

Intense red, no blue or green



Pixel **(5, 0)** has value:

(0, 160, 250)

No red, mix green and lots of blue for light blue!

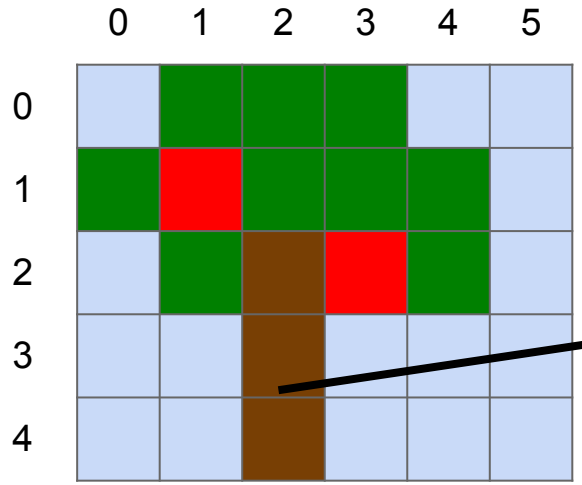
Pixel **(2, 3)** has value:

(115, 55, 18)

Reddish, darker mix!

Images in code

```
def my_first_image_func():
    image = SimpleImage('apple_tree.jpg')
    # image stores a bunch of pixel tuples!
    some_pixel = image.get_pixel(2, 3)
    # some pixel = (115, 55, 18)
    red_val = some_pixel.red # red_val is 115
```



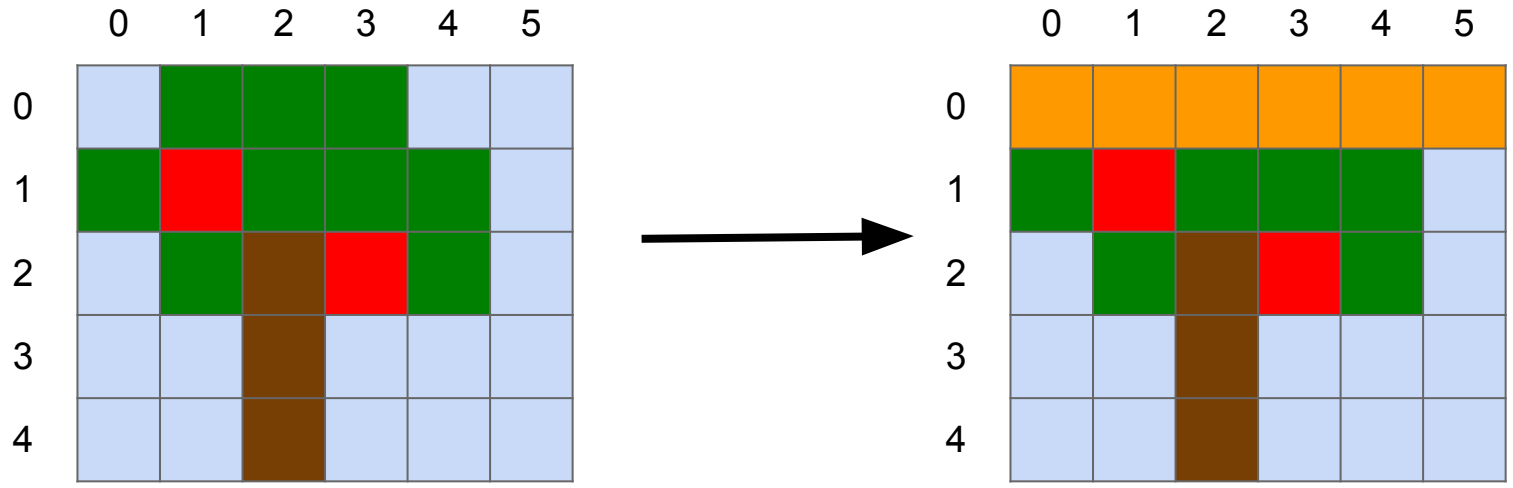
some_pixel at **(2, 3)** has value:

(115, 55, 18)

Reddish, darker mix!

Looping through pixels

Let's write a function **`sun()`** that turns every pixel in the first row of **`apple_tree.jpg`** to be a nice sunny orange, or RGB value (255, 165, 0)



Looping through pixels

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # change pixels one at a time?
    pixel = image.get_pixel(0, 0)
    pixel.red = 255
    pixel.green = 165
    pixel.blue = 0

    pixel = image.get_pixel(1, 0)
    . . .
    pixel = image.get_pixel(2, 0) . . .
```

Stanford | ENGINEERING
Computer Science

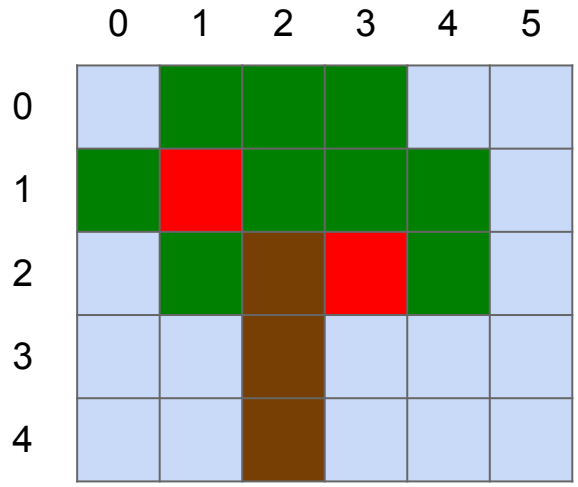
Introducing the for loop!

- It seems like we need a loop here, but we don't exactly have a condition like bit.front_clear() to test
- Instead, we have a number of pixels we want to loop through
- This is where the for loop comes in!

```
for var_name in range(start_num, end_num):
    # body of loop
    # var_name will be equal to start_num, then
    # start_num + 1 on the next loop and so on
    # until we hit end_num (don't run end_num)
```

Introducing the for loop!

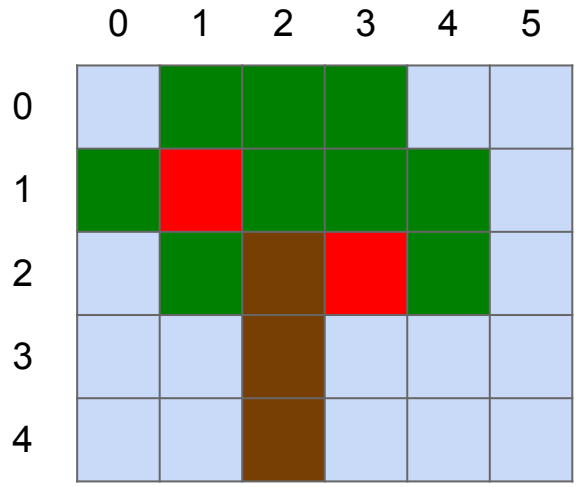
```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```

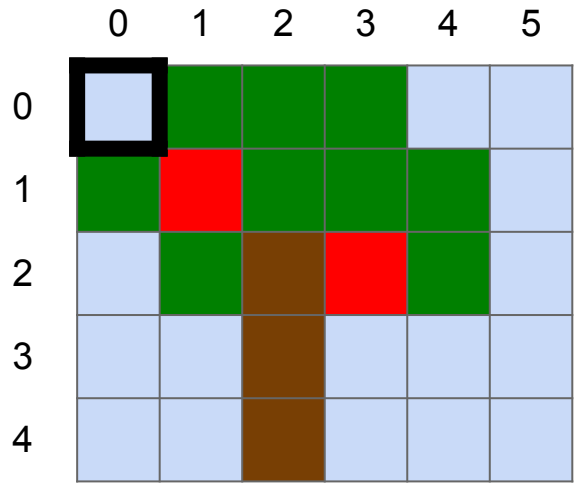


We now have a variable named **x** and its value is **0**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



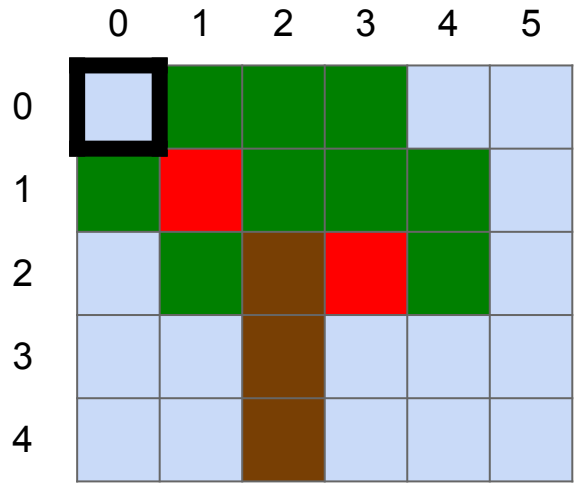
x is still **0**

We now have a variable named **pixel** its position is **(0, 0)** and its RGB value is **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



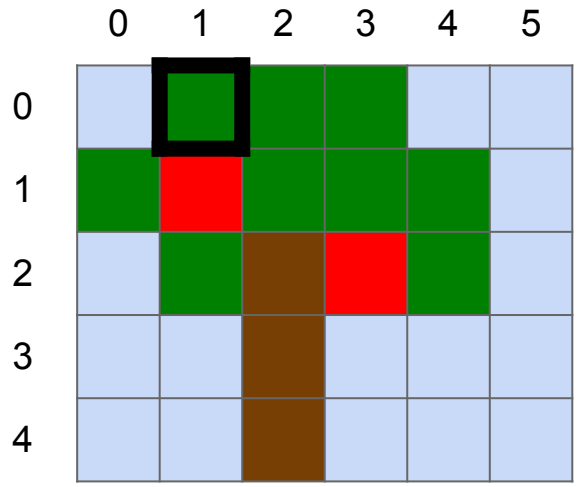
x is now **1**

pixel is still at **(0, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



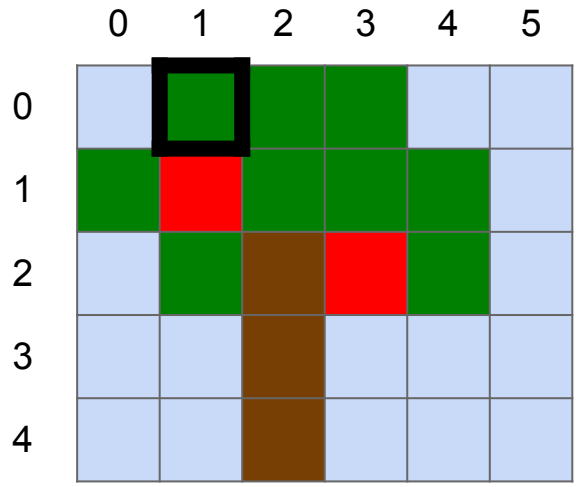
x is still **1**

pixel is now at **(1, 0)** with RGB **(0, 200, 0)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



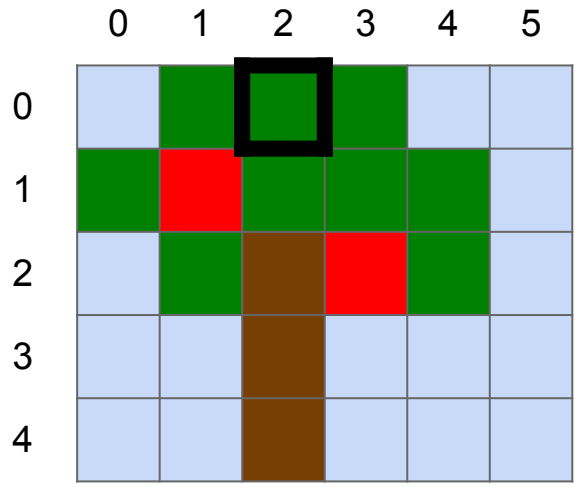
x is now 2

pixel is still at **(2, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```

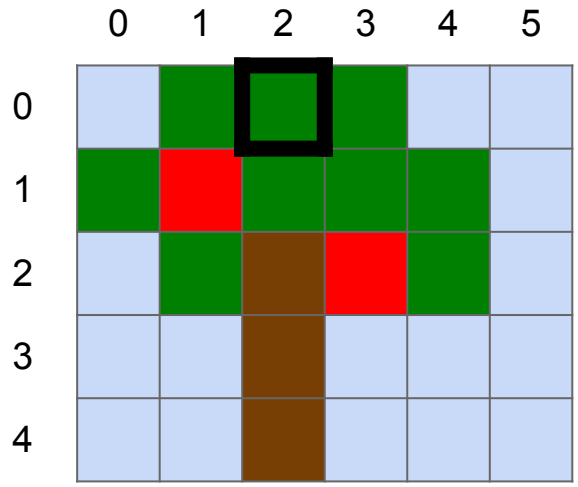


x is still **2**

pixel is now at **(2, 0)** with RGB **(0, 200, 0)**

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



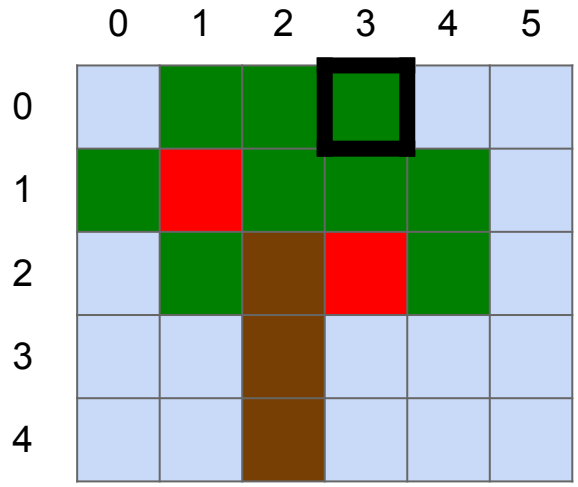
x is now 3

pixel is still at **(2, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



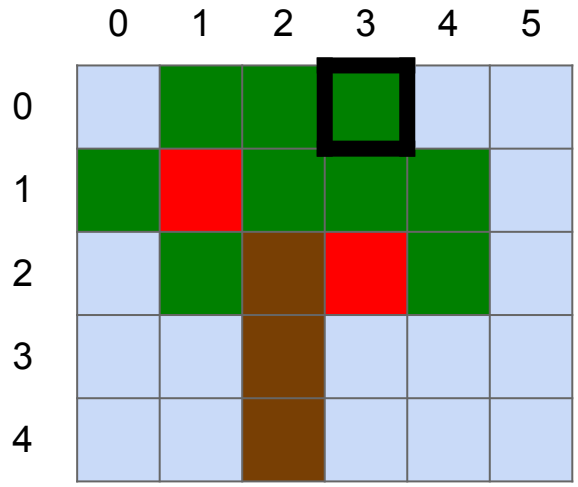
x is still **3**

pixel is now at **(3, 0)** with RGB **(0, 200, 0)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



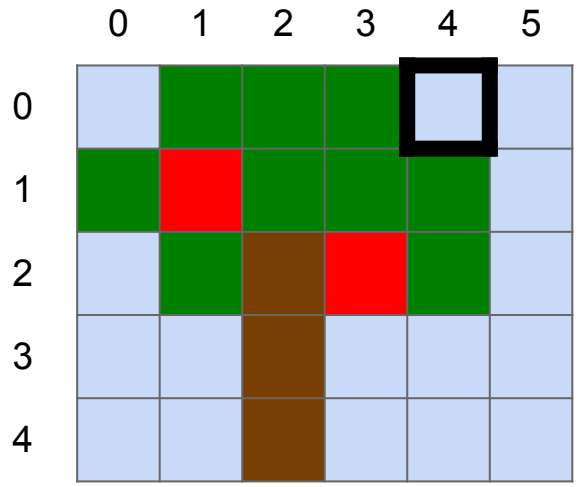
x is now **4**

pixel is still at **(3, 0)** with RGB **(0, 200, 0)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



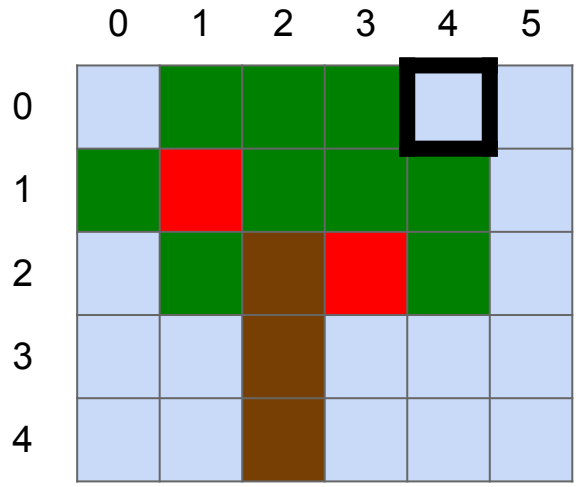
x is still **4**

pixel is now at **(4, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



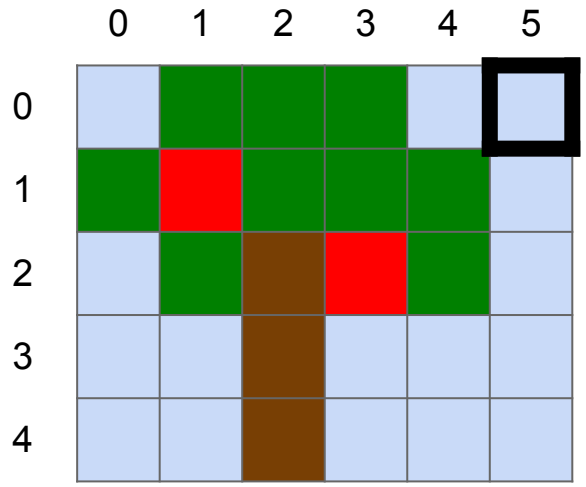
x is now 5

pixel is still at **(4, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```



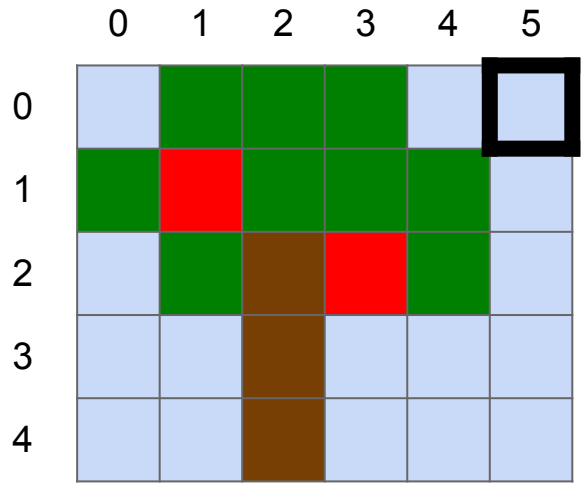
x is still **5**

pixel is now at **(5, 0)** with RGB **(0, 160, 250)**

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
```

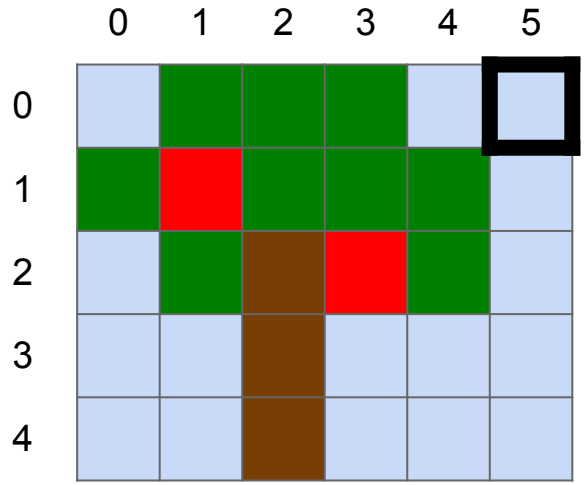


x was **5**, and our end number is **6**, so we stop here (x does not become 6)

Stanford | ENGINEERING
Computer Science

Introducing the for loop!

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # try just grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
    # post loop code
```



x was **5**, and our end number is **6**, so we stop here (x does not become 6)

Finish `sun()`

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
        # now change pixel to be (255, 165, 0)
```

Finish `sun()`

```
def sun():
    # load image file into variable ✅
    image = SimpleImage('apple_tree.jpg')
    # grabbing each pixel
    for x in range(0, 6):
        pixel = image.get_pixel(x, 0)
        # now change pixel to be (255, 165, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
```

Two tweaks

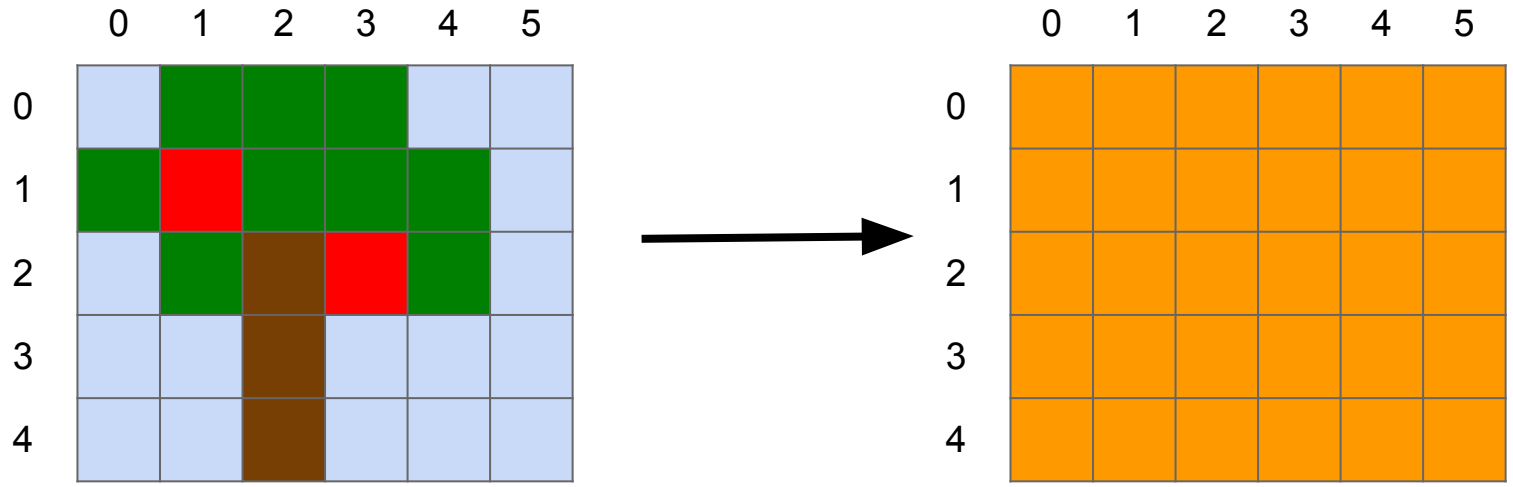
```
def sun():
    image = SimpleImage('apple_tree.jpg')
    width =  image.width # 6 for our picture
    # now, this loop works for any size picture!
    for x in range(0, width):
        pixel = image.get_pixel(x, 0)
        # now change pixel to be (255, 165, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
```

Two tweaks

```
def sun():
    image = SimpleImage('apple_tree.jpg')
    width =  image.width # 6 for our picture
    # now, this loop works for any size picture!
    for x in range(0, width):
        pixel = image.get_pixel(x, 0)
        # now change pixel to be (255, 165, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
    return image # return changed image to caller
    # without this, our work is lost!
```

Looping through ALL pixels

Let's now write a function **`all_sun(filename)`** that turns every pixel in the first row of **any image** to be a nice sunny orange, or RGB value (255, 165, 0)



Stanford | ENGINEERING
Computer Science

Start with `sun()` code

```
def all_sun(filename):
    image = SimpleImage(filename)
    width =  image.width
    # change first row
    for x in range(0, width):
        pixel = image.get_pixel(x, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
    . . .
    return image
```

Use filename

```
def all_sun(filename):
    # works for any image!
    image = SimpleImage(filename)
    width =  image.width
    # change first row
    for x in range(0, width):
        pixel = image.get_pixel(x, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
    . . .
    return image
```

How to change multiple rows?

```
def all_sun(filename):
    image = SimpleImage(filename)
    width =  image.width
    # change first row
    for x in range(0, width):
        pixel = image.get_pixel(x, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
    # change second row?
    for x in range(0, width):
        pixel = image.get_pixel(x, 1)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0 . . .
```

Our loop needs to loop!

```
def all_sun(filename):
    image = SimpleImage(filename)
    width =  image.width
    # can we repeat this loop for every row?
    for x in range(0, width):
        # the 0 coordinate is the row, need to get rows
        # 1, 2, 3. . .
        pixel = image.get_pixel(x, 0)
        pixel.red = 255
        pixel.green = 165
        pixel.blue = 0
```

Introducing the double for-loop

```
def all_sun(filename):
    image = SimpleImage(filename)
    width =  image.width
    # how many times do we need to run the inner loop?
    for y in range(0, ???):
        for x in range(0, width):
            # y is now another variable that
            # changes with each loop
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
```

Introducing the double for-loop

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
```

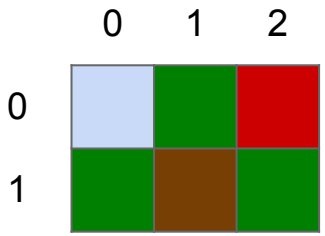
Introducing the double for-loop

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```

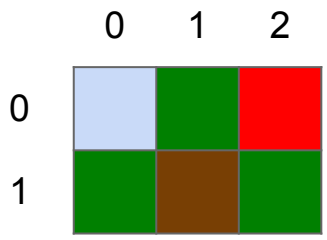


Run `all_sun('apple_bush.jpg')`

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```

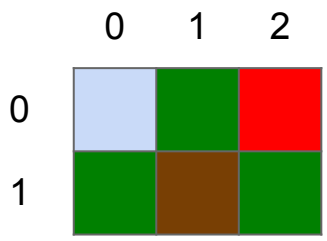


Variables:

width = 3

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



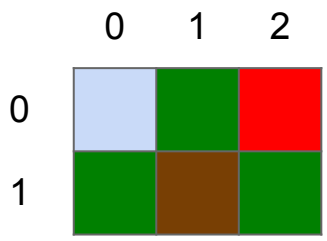
Variables:

width = 3

height = 2

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

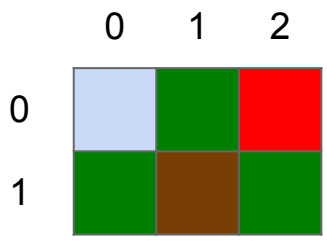
width = 3

height = 2

y = 0

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



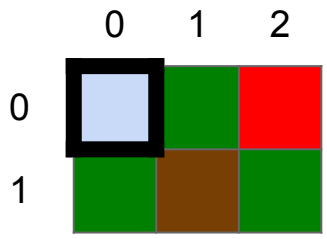
Variables:

width = 3 **x = 0**

height = 2 **y = 0**

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 0**

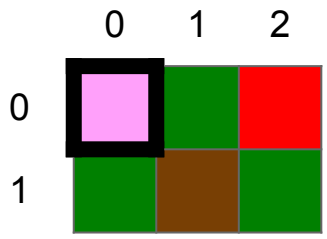
height = 2 **y = 0**

pixel = RGB(0, 160, 250) at (0, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

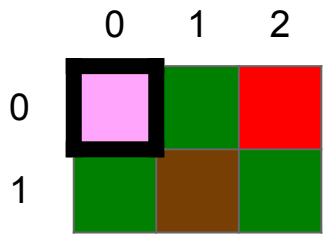
width = 3 **x = 0**

height = 2 **y = 0**

pixel = RGB(255, 160, 250) at (0, 0)

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 0**

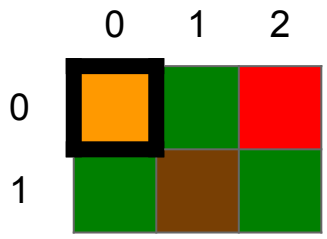
height = 2 **y = 0**

pixel = RGB(255, 165, 250) at (0, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 0**

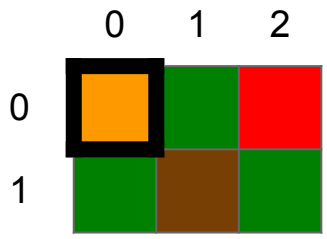
height = 2 **y = 0**

pixel = RGB(255, 165, 0) at (0, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

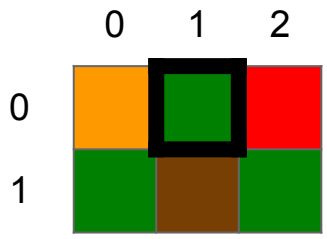
height = 2 **y = 0**

pixel = RGB(255, 165, 0) at (0, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

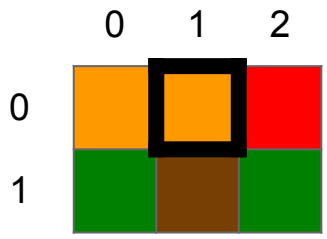
height = 2 **y = 0**

pixel = RGB(0, 200, 0) at (1, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

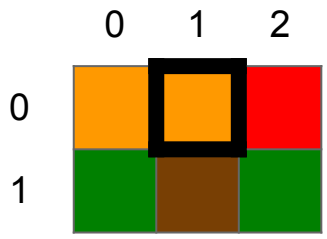
height = 2 **y = 0**

pixel = RGB(0, 200, 0) at (1, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

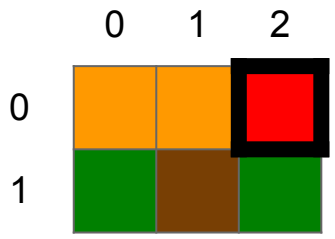
height = 2 **y = 0**

pixel = RGB(255, 165, 0) at (0, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

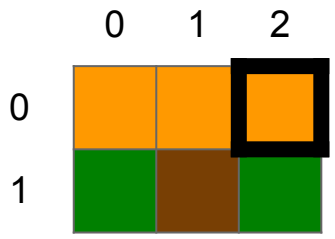
height = 2 **y = 0**

pixel = RGB(200, 200, 0) at (2, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

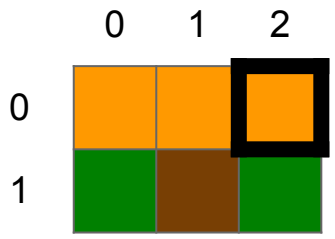
height = 2 **y = 0**

pixel = RGB(200, 0, 0) at (2, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2 (end x-loop)**

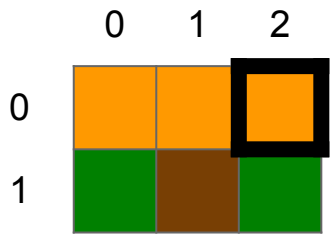
height = 2 **y = 0**

pixel = RGB(200, 0, 0) at (2, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

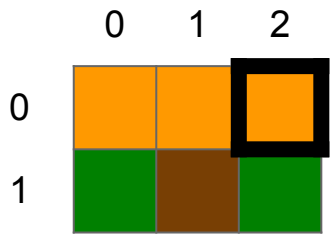
height = 2 **y = 1**

pixel = RGB(200, 0, 0) at (2, 0)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3

height = 2

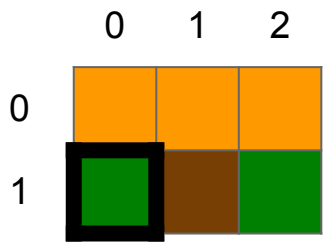
pixel = RGB(200, 0, 0) at (2, 0)

x = 0 (new x-loop)

y = 1

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 0**

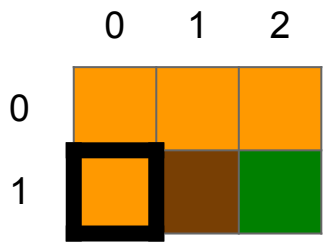
height = 2 **y = 1**

pixel = RGB(0, 200, 0) at (0, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 0**

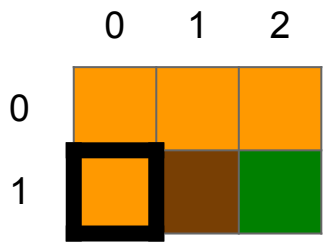
height = 2 **y = 1**

pixel = RGB(255, 165, 0) at (0, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

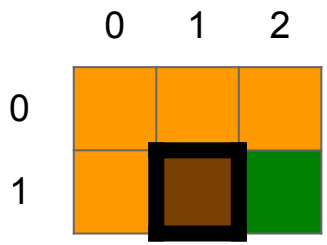
height = 2 **y = 1**

pixel = RGB(255, 165, 0) at (0, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

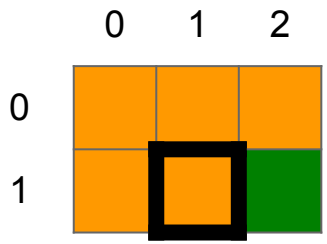
height = 2 **y = 1**

pixel = RGB(115,55,18) at (1, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 1**

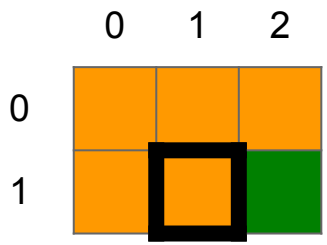
height = 2 **y = 1**

pixel = RGB(255,165,0) at (1, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

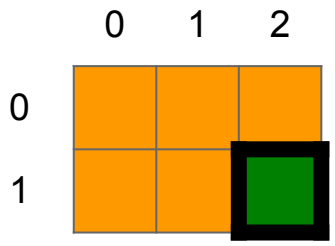
height = 2 **y = 1**

pixel = RGB(255,165,0) at (1, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

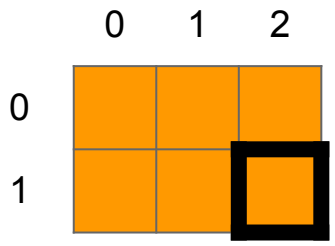
height = 2 **y = 1**

pixel = RGB(0,200,0) at (2, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2**

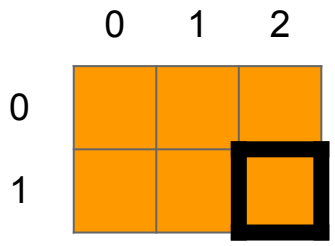
height = 2 **y = 1**

pixel = RGB(255,165,0) at (2, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2 (end x-loop)**

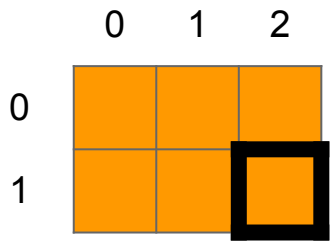
height = 2 **y = 1**

pixel = RGB(255,165,0) at (2, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2 (end x-loop)**

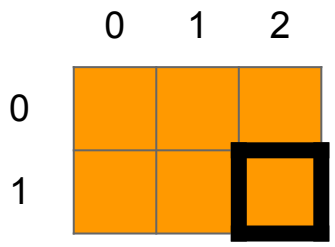
height = 2 **y = 1 (end y-loop)**

pixel = RGB(255,165,0) at (2, 1)

Stanford | ENGINEERING
Computer Science

```
def all_sun(filename):
    image = SimpleImage(filename)
    width = image.width
    height = image.height # this is the number of rows!

    for y in range(0, height):
        for x in range(0, width):
            # y is the the row index
            #x is the column index
            pixel = image.get_pixel(x, y)
            pixel.red = 255
            pixel.green = 165
            pixel.blue = 0
    return image
```



Variables:

width = 3 **x = 2 (end x-loop)**

height = 2 **y = 1 (end y-loop)**

pixel = RGB(255,165,0) at (2, 1)

Stanford | ENGINEERING
Computer Science

Recap: Double for-loops

- To loop through every pixel in an image:

```
for y in range(0, image.height):
    for x in range(0, image.width):
```

- This will run the "**`x-loop`**" **`image.height`** times, traversing row-by-row

- To traverse col-by-col:

```
for x in range(0, image.width):
    for y in range(0, image.height):
```

If time: darker

Recap

- Images can be stored in variables
- Images are made up of RGB pixels
- We can change the pixels
- We can loop through every pixel in an image with a double for loop over width and height