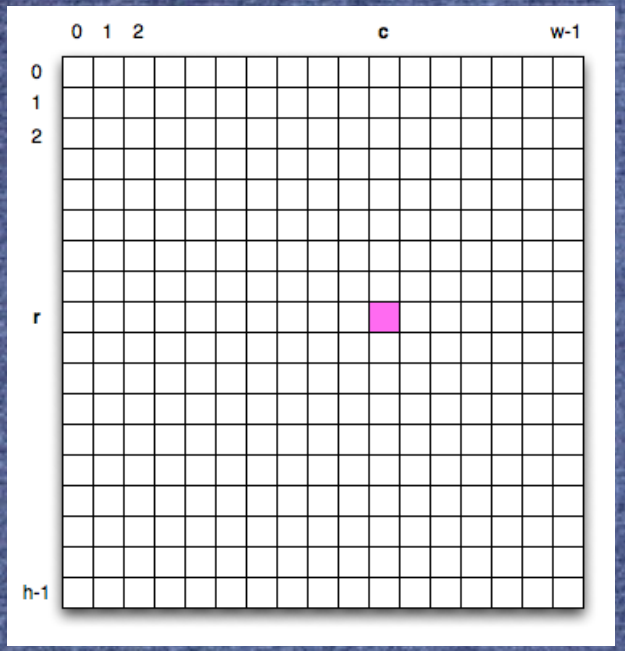
2-Dimensional Iteration (Image Processing)



Salish Kootenai College

RGB Color Model

- Images made of pixels
- Pixels are a specific color
- Colors determined by RGB Color Model

Color	Red	Green	Blue
Red	255	0	0
Green	0	255	0
Blue	0	0	255
White	255	255	255
Black	0	0	0
Yellow	255	255	0
Magenta	255	0	255

Image Module

- Two Classes

Classes, Objects, Instances

Classes, Objects, Instances

- Class – template for Object
- Object – instance of Class

Classes, Objects, Instances

- Class – template for Object
- Object – instance of Class

```
1   import turtle
2
3   wn = turtle.Screen()              # Set up the window and its attributes
4   wn.bgcolor("lightgreen")
5
6
7   tess = turtle.Turtle()            # create tess and set some attributes
8   tess.color("hotpink")
9   tess.pensize(5)
10
11  alex = turtle.Turtle()            # create alex
12
13  tess.forward(80)                  # draw with tess
14  tess.left(120)
15
16  alex.right(90)                    # draw with alex
17  alex.forward(90)
```

Classes, Objects, Instances

- Class – template for Object
- Object – instance of Class

- Methods – actions of classes
- Attributes – data of Objects

Classes, Objects, Instances Methods, Attributes

```
1   import turtle
2
3   wn = turtle.Screen()              # Set up the window and its attributes
4   wn.bgcolor("lightgreen")
5
6
7   tess = turtle.Turtle()            # create tess and set some attributes
8   tess.color("hotpink")
9   tess.pensize(5)
10
11  alex = turtle.Turtle()            # create alex
12
13  tess.forward(80)                  # draw with tess
14  tess.left(120)
15
16  alex.right(90)                    # draw with alex
17  alex.forward(90)
```

Class (Blueprint):

- Defines the structure and behavior of objects.
- Specifies attributes (data/variables) and methods (functions/actions) that objects of that class will have.
- Think of a class as a recipe or instructions for creating or building something
- Example: A class named "Car" might define attributes like "color", "make", and "model", and methods like "start_engine()" and "accelerate()".

Class (Blueprint):

- Defines the structure and behavior of objects.
- Specifies attributes (data/variables) and methods (functions/actions) that objects of that class will have.
- Think of a class as a recipe or instructions for creating or building something
- Example: A class named "Car" might define attributes like "color", "make", and "model", and methods like "start_engine()" and "accelerate()".

Object (Instance):

- A specific, concrete realization of a class.
- Each object has its own set of attributes and can perform the methods defined by its class.
- Think of an object as a specific instance of the "recipe" or the "built" product.
- Example: A "Ford Mustang" would be an object (instance) of the "Car" class. It would have specific values for its attributes (e.g., "color" = "red", "make" = "Ford", "model" = "Mustang") and would be able to use the "start_engine()" and "accelerate()" methods.

Class (Blueprint):

- Defines the structure and behavior of objects.
- Specifies attributes (data/variables) and methods (functions/actions) that objects of that class will have.
- Think of a class as a recipe or instructions for creating or building something
- Example: A class named "Car" might define attributes like "color", "make", and "model", and methods like "start_engine()" and "accelerate()".

Object (Instance):

- A specific, concrete realization of a class.
- Each object has its own set of attributes and can perform the methods defined by its class.
- Think of an object as a specific instance of the "recipe" or the "built" product.
- Example: A "Ford Mustang" would be an object (instance) of the "Car" class. It would have specific values for its attributes (e.g., "color" = "red", "make" = "Ford", "model" = "Mustang") and would be able to use the "start_engine()" and "accelerate()" methods.

Instance:

- A synonym for object.
- Emphasizes that the object is a unique creation based on the class blueprint.
- Example: You could say "The Ford Mustang is an instance of the Car class" or "The Ford Mustang is an object of the Car class".

Classes,
Objects,
Instances

Methods,
Attributes

Image Module

- Two Classes
  - image
  - pixel

Image Object

- Attributes
  - Width & Height
  - Pixels
- Methods
  - Image(*filename*)
  - EmptyImage(*w,h*)
  - getWidth(), get Height()
  - getPixel(*col, row*)
  - setPixel(*col,row, pixel*)

Pixel Object

- Attributes
  - Red, Blue, Green
- Methods
  - getRed(), getBlue(), getGreen()
  - setRed(), setBlue(), setGreen()

Working with Images

```
1 import image
2 img = image.Image("luther.jpg")
3
4 print(img.getWidth())
5 print(img.getHeight())
6
7 p = img.getPixel(45, 55)
8 print(p.getRed(), p.getGreen(), p.getBlue())
9
```

Create Instance of Image Object using Image(*filename)* method

Working with Images

```
1 import image
2 img = image.Image("luther.jpg")
3
4 print(img.getWidth())
5 print(img.getHeight())
6
7 p = img.getPixel(45, 55)
8 print(p.getRed(), p.getGreen(), p.getBlue())
9
```

Create Instance of Image Object using Image(*filename)* method

Print Width of Image Object using getWidth() method

Working with Images

```
1 import image
2 img = image.Image("luther.jpg")
3
4 print(img.getWidth())
5 print(img.getHeight())
6
7 p = img.getPixel(45, 55)
8 print(p.getRed(), p.getGreen(), p.getBlue())
9
```

Create Instance of Image Object using Image(*filename)* method

Print Width of Image Object using getWidth() method

Create Pixel Object using getPixel() method of Image Object

Working with Images

```
1 import image
2 img = image.Image("luther.jpg")
3
4 print(img.getWidth())
5 print(img.getHeight())
6
7 p = img.getPixel(45, 55)
8 print(p.getRed(), p.getGreen(), p.getBlue())
9
```

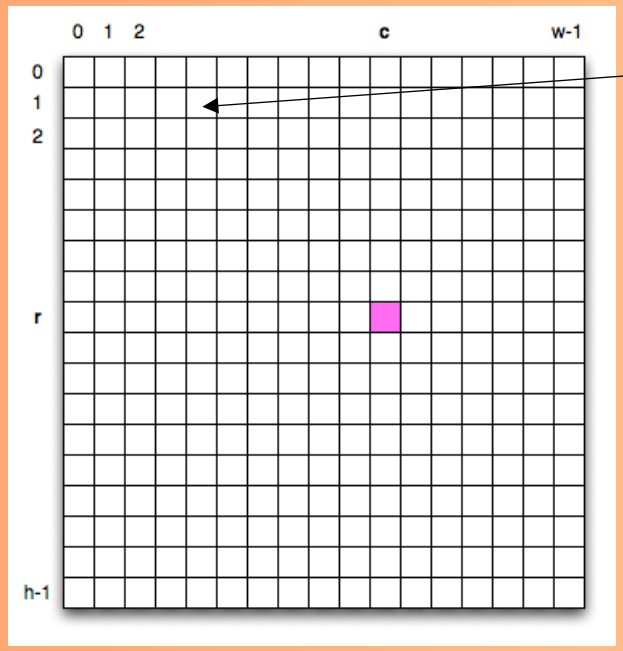
Create Instance of Image Object using Image(*filename)* method

Print Width of Image Object using getWidth() method

Create Pixel Object using getPixel() method of Image Object

Print Red value of Pixel Object using getRed() method of Pixel Object

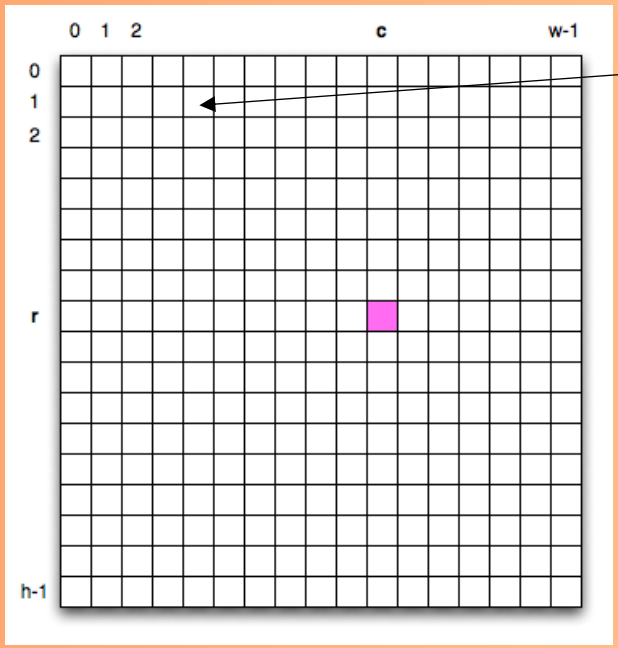
Working with Images



Each square is a pixel with three attributes
  Red, Green, Blue

Working with Images

- Goal in Image Processing is to *visit* each Pixel



Each square is a pixel with three attributes
Red, Green, Blue

How to Access Each Pixel

- Nested Iteration

Nested Iteration

- Outer Iteration

```
for i in range(5):
    print(i)
```

Nested Iteration

- Outer Iteration
- Inner Iteration

```
for i in range(5):
    for j in range(3):
        print(i, j)
```

You Try It
Nested Iteration

```
for i in range(5):
    for j in range(3):
        print(i, j)
```

Create Negative of Image

- Negative or Opposite of a color in RGB model
  - 255 – redAttribute

Create Negative of Image

- Algorithm
  - For each Pixel
    - Get Pixel's RGB colors
    - Calculate Negative RGB values
    - Set Pixel's new RGB colors

Create Negative of Image

```
import image

img = image.Image("SKCbison.gif")
win = image.ImageWin(img.getWidth(),
img.getHeight())
img.draw(win)

for row in range(img.getHeight()):
    for col in range(img.getWidth()):
        p = img.getPixel(col, row)

        newred = 255 - p.getRed()
        newgreen = 255 - p.getGreen()
        newblue = 255 - p.getBlue()

        newpixel = image.Pixel(newred, newgreen,
newblue)

        img.setPixel(col, row, newpixel)

    img.draw(win)

win.exitonclick()
```

Create Negative of Image

```
import image

img = image.Image("SKCbison.gif")
win = image.ImageWin(img.getWidth(),
img.getHeight())
img.draw(win)

for row in range(img.getHeight()):
    for col in range(img.getWidth()):
        p = img.getPixel(col, row)

        newred = 255 - p.getRed()
        newgreen = 255 - p.getGreen()
        newblue = 255 - p.getBlue()

        newpixel = image.Pixel(newred, newgreen,
newblue)

        img.setPixel(col, row, newpixel)

    img.draw(win)

win.exitonclick()
```

- Algorithm
  - For each Pixel
    - Get Pixel's RGB colors
    - Calculate Negative RGB values
    - Set Pixel's new RGB colors

You Try It

```
import image

img = image.Image("SKCbison.gif")
win = image.ImageWin(img.getWidth(), img.getHeight())
img.draw(win)

for row in range(img.getHeight()):
    for col in range(img.getWidth()):
        p = img.getPixel(col, row)

        newred = 255 - p.getRed()
        newgreen = 255 - p.getGreen()
        newblue = 255 - p.getBlue()

        newpixel = image.Pixel(newred, newgreen, newblue)

        img.setPixel(col, row, newpixel)

    img.draw(win)

win.exitonclick()
```