

Compsci 101

DeMorgan's Law, Short circuiting, Images, Tuples

Part 1 of 4

Susan Rodger
October 1, 2020



PFTD

- DeMorgan's Law
- Short Circuiting
- Images & Tuples
 - Start today, finish next class

Index without error?

```
lst = ["a", "b", "c", "a"]
```

```
dex = lst.index("b")
```

```
lst.index("b") is 1
```

```
lst.index("B") ERROR!
```

```
lst.index("B") ??? -1
```

- Use while loop to implement index.
- What is the while loop's Boolean condition?

```
dex = 0
```

```
while BOOL_CONDITION:
```

```
    dex += 1
```

Index without error?

```
lst = ["a", "b", "c", "a"]
```

```
dex = lst.index("b")
```

```
lst.index("b") is 1
```

```
lst.index("B") ERROR!
```

```
lst.index("B") ??? -1
```

- Use while loop to implement index.
- What is the while loop's Boolean condition?
 - Whether found value: `lst[dex] == elm`
 - Whether reach end of list: `dex >= len(lst)`

DeMorgan's Law

- While loop stopping conditions, stop with either:
 - `lst[dex] == elm`
 - `dex >= len(lst)`
- While loop needs negation: DeMorgan's Laws
 - `not (A and B)` equivalent to `(not A) or (not B)`
 - `not (A or B)` equivalent to `(not A) and (not B)`

`while not (lst[dex] == elm or dex >= len(lst)):`

`while lst[dex] != elm and dex < len(lst):`

Why did `>=`
become `<` ?

Think: DeMorgan's Law

A	B	not (A and B)	(not A) or (not B)
True	True		False
True	False	True	
False	True		True
False	False	True	

A	B	not (A or B)	(not A) and (not B)
True	True	False	
True	False		False
False	True	False	
False	False		True

Compsci 101

DeMorgan's Law, Short circuiting, Images, Tuples

Part 2 of 4

Susan Rodger
October 1, 2020



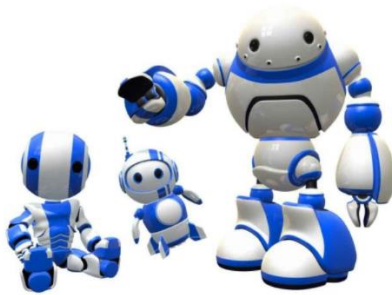
Think: DeMorgan's Law

A	B	not (A and B)	(not A) or (not B)
True	True	False	False
True	False	True	True
False	True	True	True
False	False	True	True

A	B	not (A or B)	(not A) and (not B)
True	True	False	False
True	False	False	False
False	True	False	False
False	False	True	True

Vince Conitzer

THE EVOLUTION OF AI: CAN MORALITY BE PROGRAMMED?



- Duke Professor of Philosophy, Economics, Computer Science
- Computational Social Choice
- Computers and Thought Award

To solve these problems, and to help figure out exactly how morality functions and can (hopefully) be programmed into an AI, the team is combining the methods from computer science, philosophy, and psychology “That’s, in a nutshell, what our project is about,” Conitzer asserts.



Compsci 101

DeMorgan's Law, Short circuiting, Images, Tuples

Part 3 of 4

Susan Rodger
October 1, 2020



Images



**What is
photoshop?**

Image Processing

- Convert image into format for manipulating the image
 - Visualization, Sharpening, Restoration, Recognition, Measurement, more
 - Resizing, Red-eye Removal, more
 - CrashCourse: Navigating Digital Info
 - <http://bit.ly/dukecs101-cc-images>

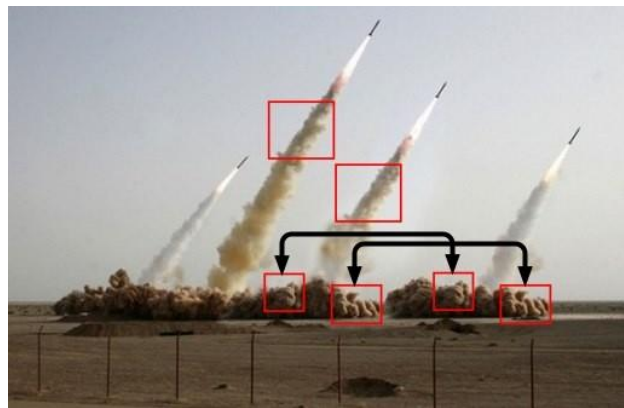
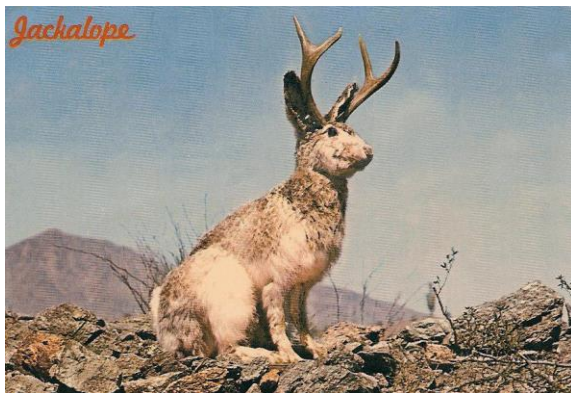


Image Library

- **PIL: Python Image Library -> Pillow**
 - To install run the command below in a terminal
 - Terminal in PyCharm, not “Python Console”
 - `pip install Pillow`
 - If that doesn’t work try:
 - `Python3 -m pip install Pillow`
- **Library has extensive API, far more than we need**
 - Concepts often apply to every image library
 - Realized in Python-specific code/functions

SimpleDisplay.py

- Access to PIL and Image module
 - What type is img?
 - <https://pillow.readthedocs.io/en/latest/>

```
6  from PIL import Image
7
8  ▶ if __name__ == '__main__':
9      img = Image.open("images/bluedevil.png")
10     img.show()
11     print("width %d, height %d" % (img.width, img.height))
```

Compsci 101

DeMorgan's Law, Short circuiting, Images, Tuples

Part 4 of 4

Susan Rodger
October 1, 2020



What is a class in Python?

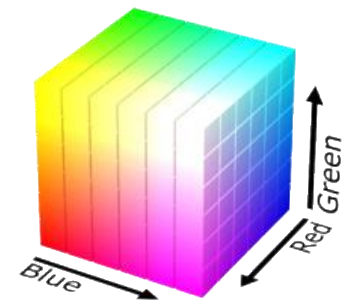
- Class \approx module \approx library (for this CS101)
- Class – Also blueprint/Factory for creating objects
 - We've used int, float, str
 - `<class 'int'>`, `<class 'list'>`
 - Everything is a class in Python3
- Use . dot notation to access object's innards
 - `"Hello".lower()` is a function aka method
 - `img.width` is an attribute aka field/property

Image Library Basics

- Library can create/open images in different formats, e.g., .png, .jpg, .gif, ...
- Images have properties: width, height, type, color-model, and more
 - Functions and fields access these properties, e.g., **`im.width`**, **`im.format`**, and more
- Pixels are formed as triples (255,255,255), (r,g,b)
 - In Python these are tuples: immutable sequence

Color Models

- Cameras, Displays, Phones, JumboTron: RGB
 - Additive Color Model: Red, Green, Blue
 - https://en.wikipedia.org/wiki/RGB_color_model
- Contrast Printers and Print which use CMYK
 - Subtractive: Cyan, Magenta, Yellow, Key/Black



Example: Convert Color to Gray



Process each pixel
Convert to gray



First View of Image for Grayscale

- Image is a collection of pixels
 - Organized in rows: # rows is image height
 - Each row has the same length: image width
- Pixels addressed by (x, y) coordinates
 - Upper-left (0,0), Lower-right (width-1,height-1)
 - Typically is a single (x, y) entity: tuple
- Tuple is immutable, indexed sequence (a, b, c)

Tuple: What and Why?

- Similar to a list in indexing starting at 0
 - Can store any type of element
 - Can iterate over
- Immutable - Cannot mutate/change its value(s)
 - Efficient because it can't be altered
- Consider `x = (5, 6)` and `y = ([1, 2], 3.14)`
 - What is `x[0]` = ? `y[0].append(5)` ?
 - <https://goo.gl/ooyHPQ>