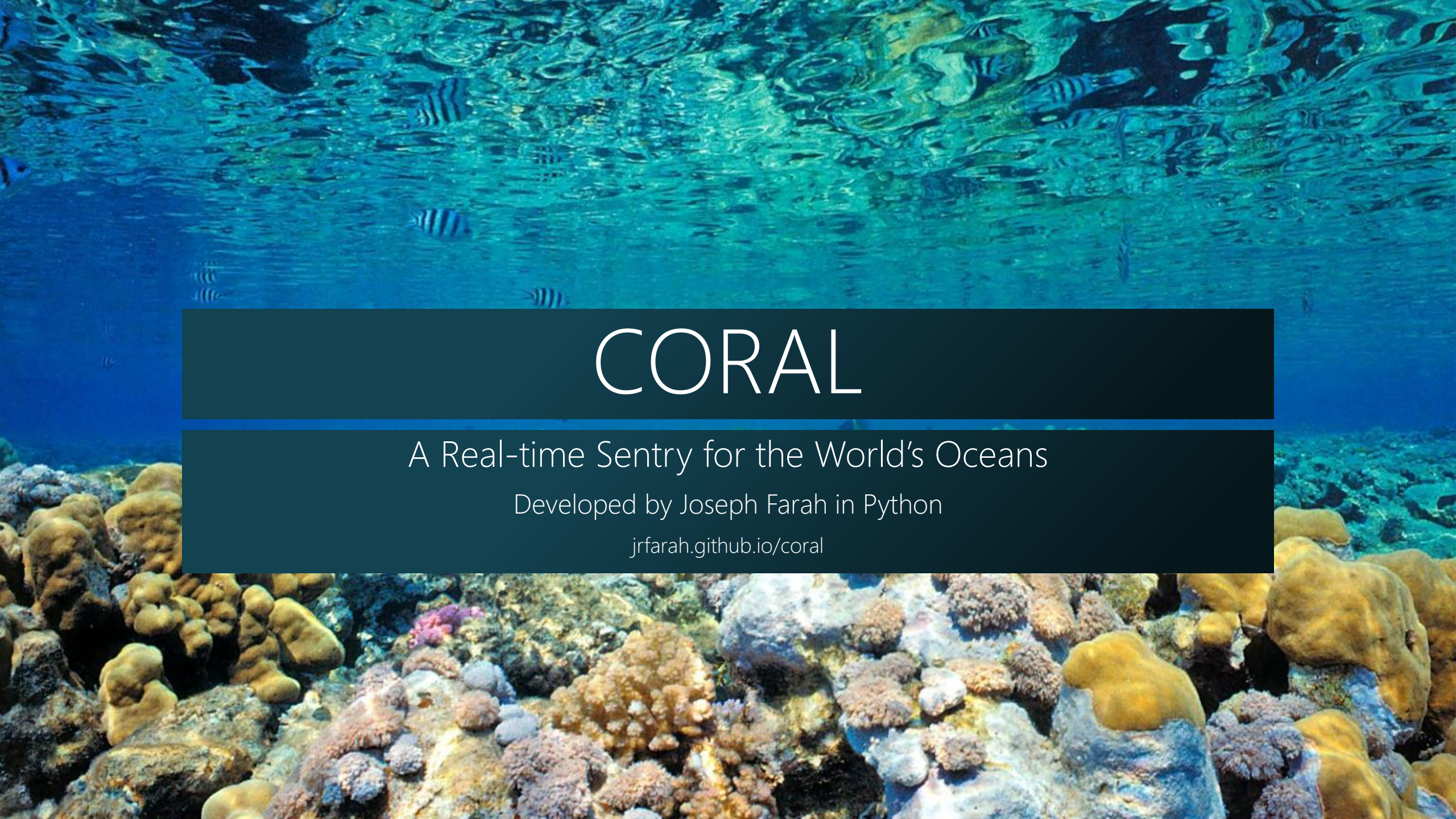


CORAL

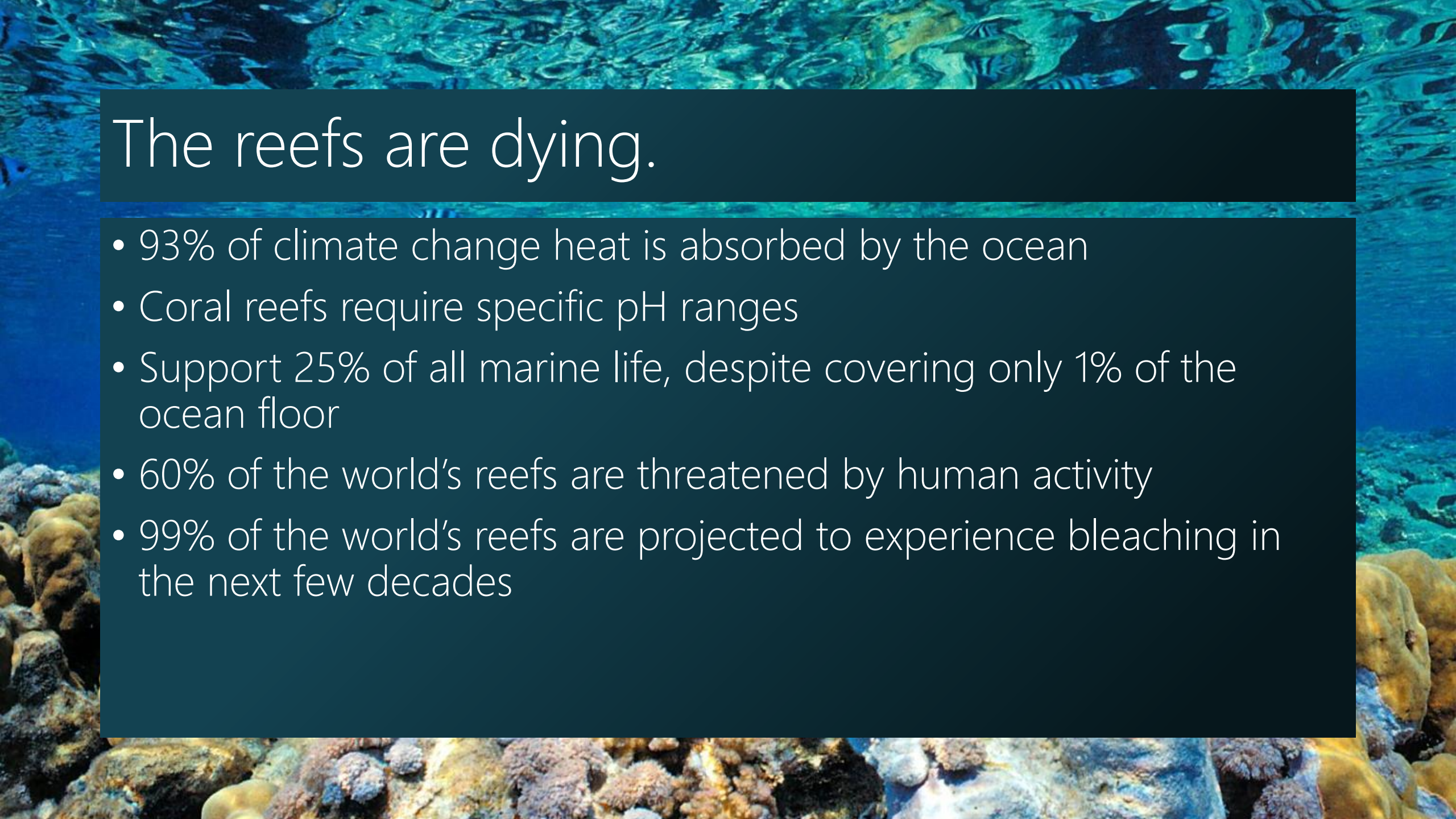
A Real-time Sentry for the World's Oceans

Developed by Joseph Farah in Python

jrfarah.github.io/coral

What are coral?

- A living creature—thousands of colonies of tiny coral polyps calcifying limestone to form hard outer shells, closely related to jellyfish
- Home to 25% of the world's marine life
- Helps direct the course of evolution in the ocean
- Coral reef plants and animals are important sources of new medicines being developed to treat cancer, arthritis, human bacterial infections, Alzheimer's disease, heart disease, viruses, and other diseases. (Source: NOAA)



The reefs are dying.

- 93% of climate change heat is absorbed by the ocean
- Coral reefs require specific pH ranges
- Support 25% of all marine life, despite covering only 1% of the ocean floor
- 60% of the world's reefs are threatened by human activity
- 99% of the world's reefs are projected to experience bleaching in the next few decades

What is CORAL?

- A central hub for real-time data from the National Oceanic and Atmospheric Administration
- A tool for scientists to monitor reef stress around the world and compare world conditions to world bleaching percentages
- An interactive and engaging way to encourage interest in protecting one of Earth's most important and beautiful resources
- Continuous monitoring of sea surface temperature at global scales provides researchers and stakeholders with tools to understand and better manage the complex interactions leading to coral bleaching. When bleaching conditions occur, these tools can be used to trigger bleaching response plans and support appropriate management decisions.

Built from the ground up with Python2.7

- Coded 100% in Python
- Modules used:
  - PIL (Python Image Library) for analysis of NOAA maps
  - MATPLOTLIB and PyRoot for graphing
  - Tkinter for GUI interactions
  - Urllib for image scraping
- Why Python?
  - Extremely versatile
  - Easy GUI framework
  - Easier to allow newcomers to create additional functions for the software (due to readability of existing code)

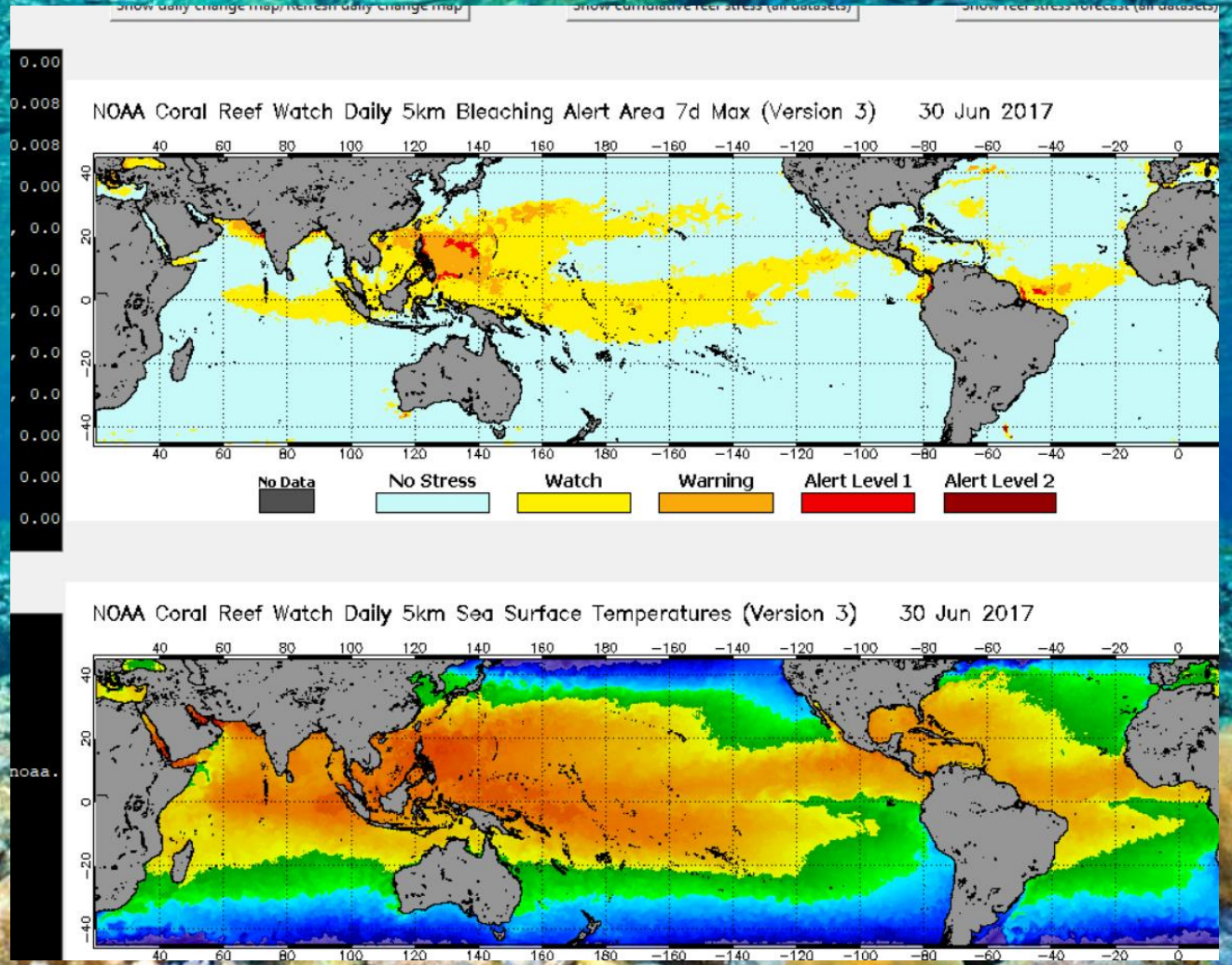
```
6  ##########################################
7  #imports
8  ##########################################
9
10 # consider adding try/except for pillow
11 import PIL.Image
12 from PIL import ImageDraw
13 from PIL import ImageTk
14 import os
15 import random
16 import datetime
17 import time
18 import matplotlib.pyplot as plt
19 from Tkinter import *
20 from tkFileDialog import askopenfilename as selectFILE
21 import tkMessageBox as tkmb
22 # not necessary for the historical program because only this one downloads crap from the
23 # scrap that its necessary for both, im an idiot
24 import shutil
25 import urllib
26 import sys
27
28 ##########################################
29 # count vars, global
30 ##########################################
31
32 # color and temperature ranges
33 no_stress_color_range = '#ffffff'
34 watch_color_range = '#fff000'
35 warning_color_range = "#faaa0a"
36 alert_1_color_range = "#f00000"
37 alert_2_color_range = "#960000"
38 black_color_range = "#000000"
39 land_range = "#c8c8c8"
40
41 # temp colors
```

A hub, a tool, an engager

CORAL is many things

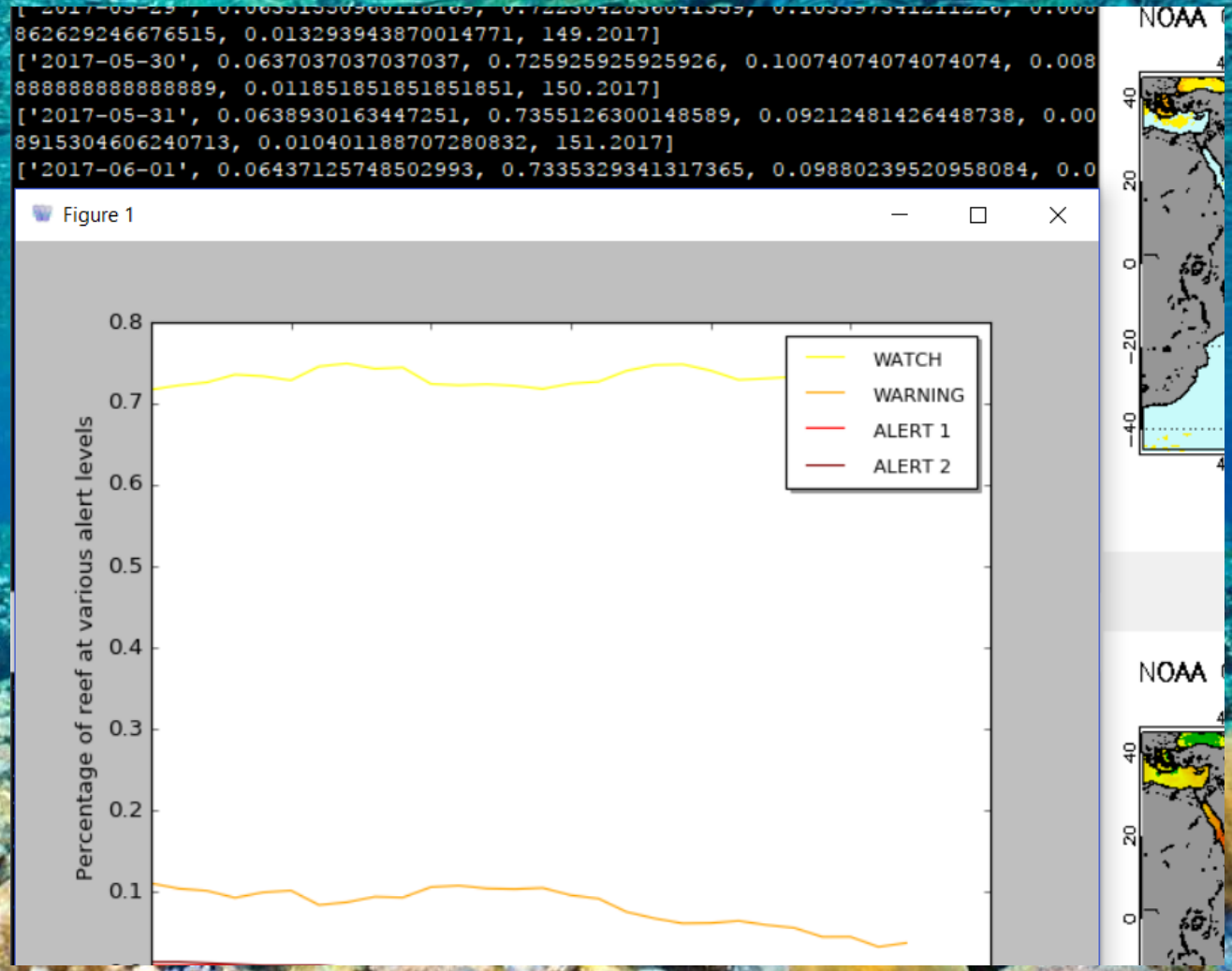
Central Hub for NOAA

- Every day the NOAA puts out datasets and maps
  - Gathered from satellites, boats, and ground monitoring stations
  - Also in collaboration with NASA's GFS and EarthWatch systems
- CORAL groups all these maps in one location
  - Provides easy accessibility and interpretation
  - Colorful and easy-to-read maps are useful for interested parties and scientists alike



A Tool for Scientists

- Keeps track of EVERYTHING
  - Percentage of coral reef bleaching worldwide
  - Percentage of coral reefs worldwide on bleaching alert status
  - Estimations of pH in 50kmx50km strips of ocean
  - Ocean temperatures
- Graphs all over time
- Provides integrated tools to compare any variable to the other and manipulate the datasets in anyway (with the exception of changing the contents)
  - This feature will be constantly under development



Dataset formatting

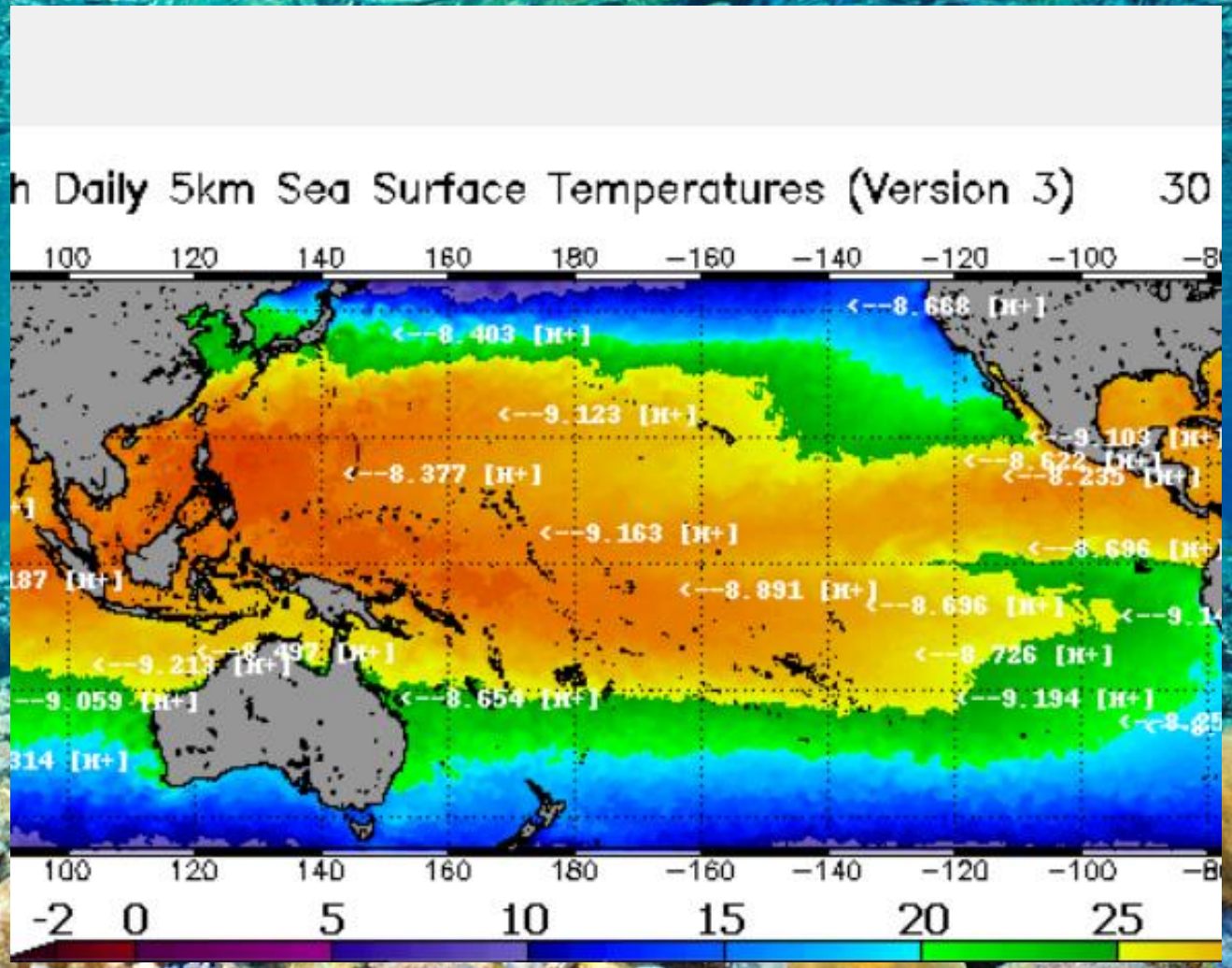
- Datasets are organized like thus:
  - [individual, (data, bunches), identifier]
- Individual
  - The timestamp that the program analyzed the data, this gives each element in the dataset a unique attribute and helps protect against confusing Python list methods
- Data bunches
  - The data is stored in the middle section, usually as decimals representing some sort of percentage
- Identifier
  - The program's estimation of the date the NOAA gathered the data, in the form of (day_of_the_year.year) ex: 167.2017

realtime.db

```
['2017-05-28', 0.06304985337243402, 0.717008797653959, 0.10997067448680352, 0.0087
['2017-05-29', 0.06351550960118169, 0.7223042836041359, 0.103397341211226, 0.00886
['2017-05-30', 0.0637037037037037, 0.725925925925926, 0.10074074074074074, 0.00888
['2017-05-31', 0.0638930163447251, 0.7355126300148589, 0.09212481426448738, 0.0089
['2017-06-01', 0.06437125748502993, 0.7335329341317365, 0.09880239520958084, 0.004
['2017-06-02', 0.06485671191553545, 0.7285067873303167, 0.10105580693815988, 0.006
['2017-06-03', 0.06515151515151515, 0.7454545454545455, 0.08333333333333333, 0.006
['2017-06-04', 0.06534954407294832, 0.7492401215805471, 0.08662613981762918, 0.001
['2017-06-05', 0.06584992343032159, 0.7427258805513017, 0.09341500765696784, 0.001
['2017-06-06', 0.0662557781201849, 0.7442218798151001, 0.09244992295839753, 0.0015
['2017-06-07', 0.0685805422647528, 0.7240829346092504, 0.10526315789473684, 0.0031
['2017-06-08', 0.0698051948051948, 0.7224025974025974, 0.10714285714285714, 0.0032
['2017-06-09', 0.07072368421052631, 0.7236842105263158, 0.10361842105263158, 0.003
['2017-06-10', 0.07251264755480608, 0.7217537942664418, 0.10286677908937605, 0.001
['2017-06-11', 0.0735042735042735, 0.717948717948718, 0.10427350427350428, 0.00170
['2017-06-12', 0.07452339688041594, 0.7244367417677643, 0.09532062391681109, 0.001
['2017-06-13', 0.07530647985989491, 0.7267950963222417, 0.09106830122591944, 0.001
['2017-06-14', 0.07651245551601424, 0.7402135231316725, 0.07473309608540925, 0.001
['2017-06-15', 0.0776173285198556, 0.7472924187725631, 0.06678700361010831, 0.0, 0
['2017-06-16', 0.07904411764705882, 0.7481617647058824, 0.06066176470588235, 0.001
['2017-06-17', 0.08206106870229007, 0.7404580152671756, 0.061068702290076333, 0.00
['2017-06-18', 0.08565737051792828, 0.7290836653386454, 0.06374501992031872, 0.001
['2017-06-19', 0.08704453441295547, 0.7307692307692307, 0.058704453441295545, 0.00
['2017-06-20', 0.08775510204081632, 0.7326530612244898, 0.05510204081632653, 0.002
['2017-06-21', 0.09014675052410902, 0.7379454926624738, 0.0440251572327044, 0.0020
['2017-06-22', 0.09492273730684327, 0.7262693156732892, 0.04415011037527594, 0.002
['2017-06-23', 0.09772727272727273, 0.7340909090909091, 0.031818181818181815, 0.0,
['2017-06-24', 0.09839816933638444, 0.7276887871853547, 0.036613272311212815, 0.0,
```

An engager for laymen

- Also provides less-scientifically-important data and statistics
  - Not for studies but to engage people who download the program to learn more about reef bleaching
  - Non-scientists will also be kept update on global ocean CO2 ppm, pH flux, bleaching percentages, etc, but in a less data raw format
- Provides opportunity for interested people to contribute
  - In the future, CORAL will be able to sync datasets from thousands of computers to provide millions of data points for scientists regarding temperature/pH changes vs bleaching all around the world, something no single computer can accomplish



Where to go from here?

- Easier interface to graph multiple variables side-by-side
- Integration of thousands of pH data points into one master dataset
- Better integration with pyROOT
- Provide on-tap explanations for maps
- Use machine–learning and TensorFlow to improve predictions of coral reef bleaching
- Improve UI
  - UI sometimes realigns when new maps are loaded

And now, for a live demo