Accessibility of Jupyter for Impaired Individuals

Asha Shekar

MICHIGAN STATE UNIVERSITY

CMSE

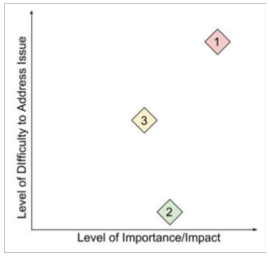
Introduction

Jupyter is an open-source programming environment that uses "literate programming" by combining documentation, coding and visualization into one file format. The Jupyter notebook format is also particularly well-suited for learning environments, where the instructional content (text, links, videos, etc.) can be combined with executable code and student output (graphs, charts, solutions, feedback) in a single, easy-to-use, file format. Unfortunately, Jupyter software is not fully accessible. This means individuals with impaired senses and abilities may not be able to interact with Jupyter effectively. MSU is leading an initiative working towards making Jupyter more accessible. In this project, I investigated the different accessibility issues with Jupyter and identified key problems that, if fixed, would provide an improved experience for the greatest number of our students. This project made progress in solving two key issues; 1) automatically identify and measuring color contrast across the entire graphical user interface and 2) identify structural edits to the source code that can be made to improve the overall readability for screen readers.

Methods

Three Accessibility Focus Issues:

1) Keyboard Traps
2) Visual Hindrances
3) Screen Reader Operational Requirements



Visual Hindrance – Color Contrast:

WCAG 2.0 Guideline 1.4.3: "The visual presentation of text and images of text has a contrast ratio of at least 4.5:1." [2]

Screen Reader Operational Requirement– Alt-Text, Language:

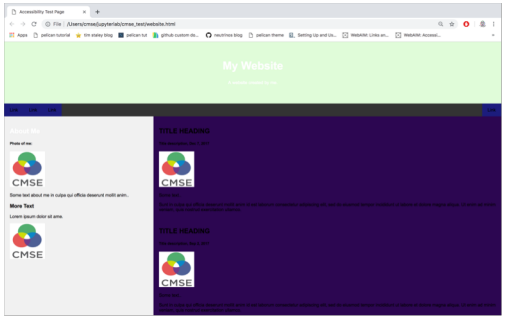
WCAG 2.0 Guideline 1.1: "Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language." [3]

Results

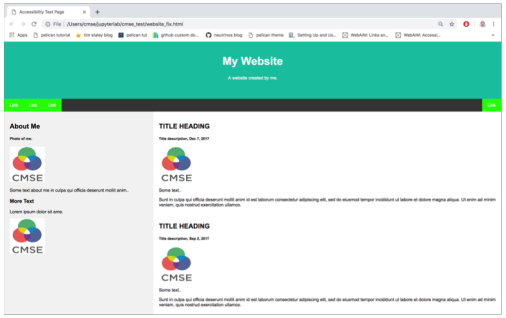
Python code iterates through the .html and .css flies in the Jupyter repository, pulling out all of the utilized colors. Using the following equation, the contrast ratio is calculated between two colors. In order to be AA compliant, the contrast ratio needs to be above 4.5.

```
def brightness(rgb):
    if rgb[0] <= 0.03928:
        r = rgb[0]/12.92
    else:
        r = ((rgb[0] + 0.055)/1.055)**2.4
    if rgb[1] <= 0.03928:
        g = rgb[1]/12.92
    else:
        g = ((rgb[1] + 0.055)/1.055)**2.4
    if rgb[2] <= 0.03928:
        b = rgb[2]/12.92
    else:
        b = ((rgb[2] + 0.055)/1.055)**2.4

    return 0.2126*r + 0.7152*g + 0.0722*b

def contrast(rgb_text, rgb_back):
    text_light = brightness(hexToRGB(rgb_text))
    back_light = brightness(hexToRGB(rgb_back))
    if text_light > back_light:
        return round((text_light + 0.05)/(back_light + 0.05), 3)
    else:
        return round((back_light + 0.05)/(text_light + 0.05), 3)
```

Code for contrast ratio calculation.



Example webpage with incorrect contrast ratio between text and background.



Same example webpage with compliant contrast ratio between text and background.

Conclusion

Jupyter is an ongoing project working towards a more accessible development environment for all programmers. A few major issues have been identified, namely keyboard traps, visual hindrance, and screen reader operation. The issue of visual hindrance was chosen because of its relative importance but fairly straightforward implementation. While other resources for calculating contrast ratio already exist, they do not allow for iterating through source code and pulling out contrast ratio issues. The developed contrast ratio code serves as a versatile solution that could theoretically be used on any application.

Future Development

It is evident in the GitHub project for Jupyter that there is a lot of work to be done before Jupyter can be considered truly accessible. Addressing some of the previously mentioned issues not covered in this project would be relevant next steps. Also, there a more preemptive measures that can be learnt from this process on how applications should be developed from the beginning with accessibility in mind.

Acknowledgements

Project completed with the help of my mentor, Dr. Dirk Colbry, and the GitHub community working on the Jupyter open source project.

References

[1] https://github.com/jupyterlab/jupyterlab

[2] https://www.w3.org/WAI/WCAG21/Understanding/contrast-minimum.html

[3] https://www.w3.org/TR/UNDERSTANDING-WCAG20/text-equiv.html