# **Advanced Code Editor (ACE) Notebook Extension**

## **Background**

This project introduces an extension to the Advanced Code Editor (ACE) that allows the creation of rich media notebooks, similar to Jupyter notebooks. ACE notebooks can be hosted directly on the web because the ACE editor is written in JavaScript and runs in most modern web browsers.

ACE does not itself load and store content - this is provided by an external service. Refer to the new technology report describing the Project Cloud Service (*PCS*) as an example. *PCS* can load data from a cloud file and store it back in the cloud. *PCS* with ACE can also run as a desktop application.

The addition of notebook capability to ACE requires minor modifications to the source code of the ACE editor, as well as a new module. This software is collectively referred to as ACE/NB.

**About ACE**

ACE is an open source code editor written in JavaScript. It matches the features and performance of most desktop editors. It can be easily embedded in any web page or JavaScript application. ACE is maintained as the primary editor for the Cloud9 IDE from Amazon Web Services. Current integrators of ACE also include Khan Academy, R studio, and Wikipedia.

ACE provides many features, including syntax highlighting for over 110 computer languages, over 20 visual themes, and good performance even for very large files. Code checking is also built-in for JavaScript.

The ACE source code is hosted on GitHub and released under the BSD license and is hosted on the major content delivery networks such as jsdelivr.net and CloudFlare.

Microsoft Visual Code is similar, but more complex, and is mainly useful for the desktop only. ACE has been proven workable and useful on mobile devices such as smart phones and tablets.

Therefore, ACE was selected for this notebook extension project.

**What is a Notebook?**

A “Notebook” is an interactive computing platform that combines live code, equations, narrative text, visualizations, interactive dashboards and other media. The idea is a powerful extension of an earlier concept called “Literate Programming,” which combines a programming language source code with its a documentation in a simultaneously editable and viewable platform. Many notebook systems are now web based.

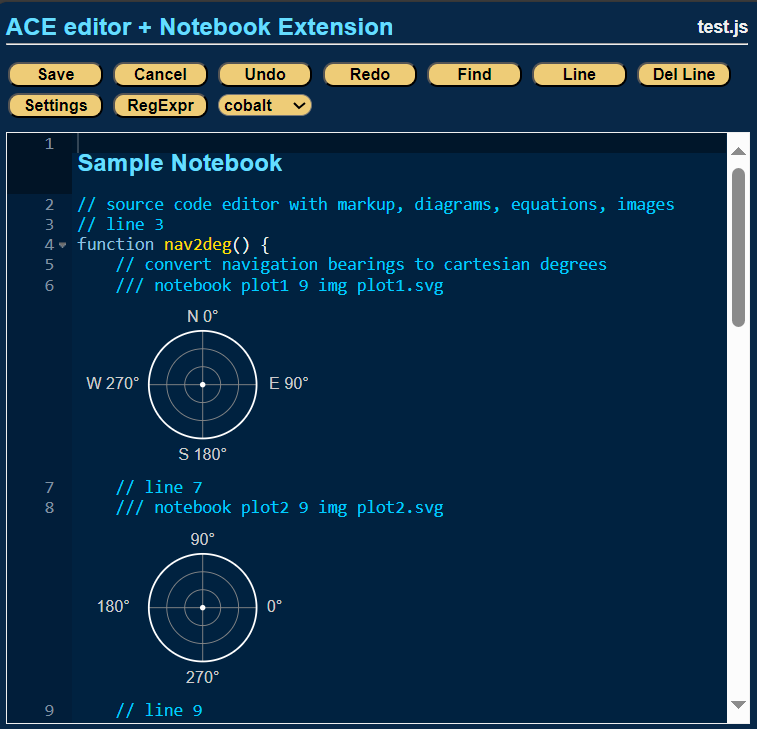
A notebook can also be viewed as an extension of Computer Aided Learning, in which a pupil is guided through a sequence of steps to solve a problem, and each step is presented one piece at a time. In some systems, the student can enter their own solution and compare it to a prepared solution, for each step. The notebook can include graphs of equations, equations, source code and pseudo code. Figure 1 below shows an example of a math problem to convert navigational bearings to cartesian degrees where diagrams are included to elucidate the problem. Directives are processed by the ACE editor with Notebook Extension, that cause inclusion of rich media content. The directives begin with three slashes /// which are treated as comments and ignored by most computer languages. The # comment introducer can also be used for Python programs.

## **Description of the Software**

The notebook extension to the ACE editor processes rich media directives in real time as they are entered by the user, or if they are loaded from persistent storage. The rich media content is fetched as directed from external files.

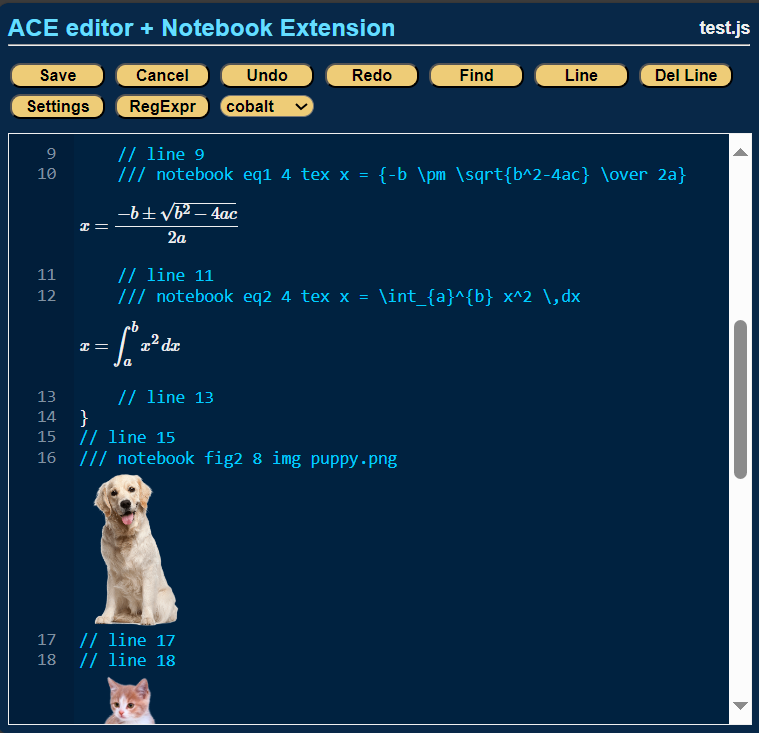
There is also a facility for programmatic insertion of rich media content or markup. In Figure 1, the title “Sample Notebook” was inserted by a wrapper script, that loaded the editor at startup.

An example of a wrapper script would be the Project Cloud Service (*PCS*), which was disclosed in a separate new technology report. The wrapper script configures the user interface, loads the content to be edited, starts the ACE editor, and can also inserts notebook content as shown below.



**Figure 1. Source code combined with markup and graphs in ACE/NB.**

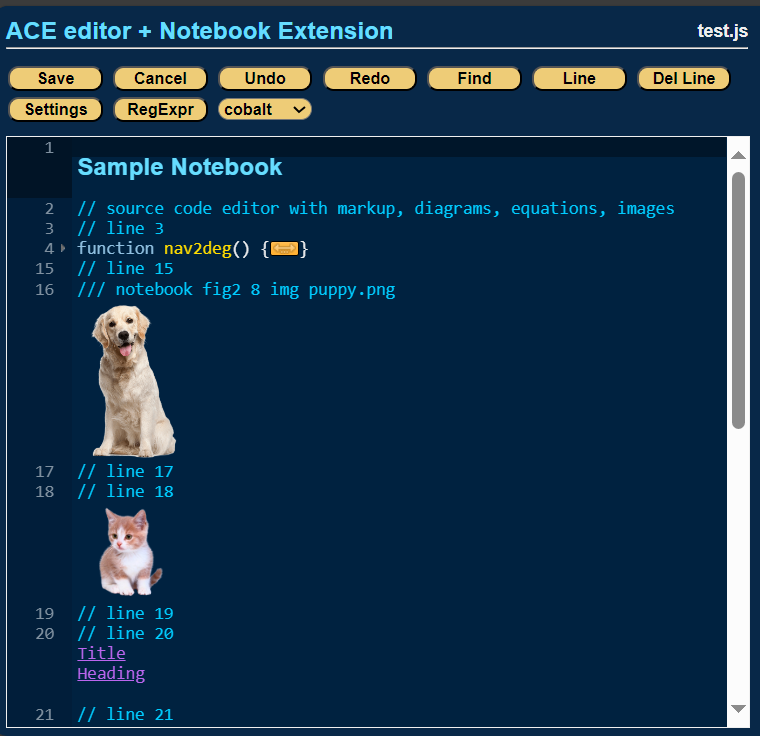
The two graphs are in scalable vector graphics (SVG) format, and these are static graphs that are fetched by ACE/NB from persistent storage. In this case the contents are fetched from a website managed by *PCS*.



**Figure 2. Tex equation rendering and rich media is also directly integrated with ACE/NB.**

The user can directly enter equations in TeX code which is rendered in real time below. Changes are immediately updated. When the file is saved it can be recalled and the equation will be re rendered as last updated. Text rendering is performed by the MathJax system which is integrated into ACE/NB.

Another supported notebook element is the image element. In this case stock images are fetched from the *PCS* website.



**Figure 3. Source code folding is supported. HTML functionality is supported.**

A useful feature of the ACE editor is source code folding, which can contract functions and hierarchical data into single lines, and these can be re-expanded as needed by the user. In figure 3 the function nav2deg has been folded into a single line. ACE/NB suppresses rendering of notebook elements that are contained within the folded function and re-render them when the function is expanded.

Recall that ACE itself is running in the user’s web browser, and that underlying technology is HTML. This means that HTML functionality is natively supported. As a simple example, above and figure 3, in line 20 the “Title” link has been added, and when the user clicks on it, the cursor moves to the line 1 of the file “Sample Notebook”.

Thus, complex documents or notebooks containing a table of contents with links to within the document, or to external websites can easily be implemented. Generation of HTML dynamically to perform specific functions, such as a table of contents, or a dynamic user controllable plotting facility, is outside the scope of this software project, however the function ace\_nb.insert() can be called by the wrapper script either when the file is loaded, or at any time in the life cycle to provide these kinds of features.

**Supported Notebook Directives**

|  |  |  |
| --- | --- | --- |
| **Type** | **Code** | **Description** |
| Image | img | Inserts an image, SVG, or can be animated GIF also |
| Tex | tex | Parses a TeX equation and renders it immediately |
| HTML | html | Inserts HTML which can be links, buttons, etc. |

**Software Components**

Most of the core functionality of the ACE editor is contained in a single file called ace.js, consisting of about 20,000 source lines of existing open sourced code. Approximately 15 patches of new code were applied, which is part of this NTR, and these consist of the addition of one to two lines per patch. These patches hook into a new module called ace\_nb.js, which provides the notebook extension functionality to the ACE editor.

The file ace\_nb.js contains new code that implements the functionality disclosed in this report. It contains about 20 functions which deal with the creation, updating, and life cycle of notebook rich media elements.

Text

Description automatically generated

**Figure 4. File contents as stored. Rich media directives are seen as comments by JavaScript in this example.**

Files that contain notebook elements are compatible with any standard text editor, which will simply not display the rich media elements.

**Open Source Dependencies, Licenses, and Credits**

**ACE** editor is Copyright (c) 2010, Ajax.org B.V.   
Licensed under the BSD open source license.

**jshint** is Copyright (c) 2012, Anton Kovalyov.  
Licensed under the MIT open source license.

**MathJax** is Copyright (c) 2021, The MathJax Consortium.  
Licensed under the Apache 2.0 open source license.

**polyfill service** is Copyright (c) 2024, Jake Champion.  
Licensed under the MIT open source license.

**References**

ACE editor, <https://ace.c9.io/>

MathJax, <https://www.mathjax.org/>

JShint, <https://jshint.com/docs/>

[end]