# IU Verse

* [When Life Gives You Tangerine](https://www.facebook.com/photo.php?fbid=1064828759020209&set=a.553989473437476&type=3)

IU took vocal lessons to produce lower notes.

* [Vocal lessons](https://x.com/erlinzhang/status/1897953410571550813)

# Gamma run-through of GTML

[0:00, Run—through gamma] This Plastic Tornado Distribution version 24 has a Gamma run-through for PepMan with GTML.

[0:20, Run—through gamma] The HTML is rendered using JEditorPane. This includes the tag rendering. This is extremely helpful because the tag allows you to peer into the tag structure.

[0:53, Run-through gamma] A context in the world of Dynamic Web Services of Tomcat means it is a developer’s naming scheme. The developer would create a name for the WAR file. The WAR gets extracted with a Catalina poller every one second and extracts the WAR file to create a daemon with the context information handled by the HttpServlet. Catalina, by the way, is Tomcat along with the static log and config file and the location of the webapps where the WAR file gets extracted.

For PepMan, the root location of ‘gtml’ is the root file of the context. There are three root file: Working directory, Bundled JAR, and SandBox. The options are given at the top of the ‘GTML’ tab. Eventually the jar file would be converted to a KAR file (KonaWare Archive) where the timestamp with a poller allows the location gtml content to be dynamic loaded and rendered into the ‘gtml’ Editor Pane.

[1:25, Run-through gamma] The original gtml packaged content is located in $PTD/doc/release/gtml directory. The developer would have to copy this content to ‘gtml’ of their working directory. If the developer starts the PepMan using Eclipses, it is located in eclipse-workspace/PepMan/bin. Most likely it is hard to get the source to compile in a portable manner because I have yet to make the Modules portable. Chances are: The developer would run PepMan using the java -jar command. The $PTD/doc/Installation.doc.

[1:50, Run-through gamma] The Packaged Directory of History has a Page on Tim-Bernes Lee. On ‘Marketing’ package, it talks about Menlo Ventures and its need to hire more people for its AI success.

[2:48, Run-through Gamma] More work will be done on the Public Key Authentication (Shama Algorithm aka Cube Control). The Cube Control will have a new component of Spreadsheet App. The entry of the Spreadsheet App will house the private and public keys.

[3:35, Run-through Gamma] The three load options are at the top of the ‘GTML’ Tab’s Combo Box option. They are: **Bundled Jar**, **Sandbox**, and **Working Directory**.

[3:41, Run-through Gamma] The rendering of the Directory Structure allows the content to be coded and debugged. In the world of PC, the office application has a set of menus and user interface to create content. IN the world of Plastic Tornado, it is that “packets” of Entry for Seafloor and KW-HashMap for Skyline to be the end-all of content. For example, three files: Books, Chapter, and Pages are needed to create a book at the minimum. For a Spreadsheet, a file with Content Entry like $CELL.A=”$6:00”. That’s it for the Spreadsheet document. Of course, there is a manifest that controls the Cube Control. As another example, a Calendar app would have a document as follows: February 14: 2025, tag=”Valentine’s Day”, meet at McDonald’s, Cube Control = $spreadhshet/entry.public key”. That is all there is for the document file to be sent to your destination.

[4:17, Run-through Gamma] A Entry is shown to render the Directory Package called ‘history’ and ‘marketing’ to have the Pages loaded for **Book Rendering**. The Entry is an important milestone. It offers “completes” for the documents and the viewer to be coded and shipped as part of **Plastic Tornado Distribution**.

[Run-through Gamma](https://www.youtube.com/watch?v=QvVCh8d5zLQ)

# Social Media and Grok

# [Run-through gamma 'gtml'](https://www.youtube.com/watch?v=QvVCh8d5zLQ)

# Homework for Dynamic Web Services

Create a ‘Hello World’ web apps to be deployed on Tomcat. You will see why Tomcat is very scalable using the **Dynamic Web Service** architecture. In building the next-generation development environment for flying cars, Tomcat is the de facto standard platform.