### TODO:

* The topic of scripts should come from the OS.
* $OS(text that will be moved to the DESIGN cell in the OS draft)
* Improve the $new tag so it also passes information about lesson type. For this we should have designers’ input on what information they need.
* ~~A $design tag for the board column so the content goes into the design cell in the OS draft.~~
* A tag ($ke?) that can be used in sidebar comments in the lesson plan to indicate that those are the important comments to be copied into the lesson plan in the KE style, while the rest will go to sidebar comments in the script drafts.
* Row level tags: $new(par), $insa/b--$end, $branch(par)--$endbranch, $OS

TODO for post-scripting:

* [popup] tag copies a line in on-screen style.
* Generate ppt/pdf automatically.

| Board | Teacher | Student | Notebook |
| --- | --- | --- | --- |
| ($cut)  $new(par)  $OS  $screen  $insa--$end  $insb--$end  $branch(par)--$endbranch  ~~strikethrough~~ | $cut  $OS  ~~strikethrough~~ | $cut  $tutor  $weak  $average  $strong | ($cut)  $wtd : coming one day! |

* User interface overhaul: now you can run the program from any folder.
* Add tags in the lesson plan (saved in the .docx format) 🡪 Run script\_drafter 🡪 Edit OS draft 🡪 Run script\_drafter again 🡪 Edit script drafts.
* The tags are case sensitive.
* It’s probably a good idea to use an easy-to-spot style for the tags and put them at the end of the cell, although it generally does not matter where you put them.
* Striking-through a whole cell has the same effect as $cut on the Teacher and Student columns. By default, if the Teacher cell is cut, then the Student cell on the same row is also cut.
* You can change in the OS draft the names of the branches (such as, change from “Weak” to “If quiz problem 1 was solved correctly”), and the new names will be copied over to the script draft.
* You can add tables in the OS draft (copy from the pieces.docx file), which will generate empty script draft files.
* The characters’ names are now in a table in the OS draft, which are copied over to the script drafts.
* Tutor’s correct responses are now automatically generated based on EMW’s specification (response.csv).
* The $screen tag now supports tables and word math formulas, and replaces pictures with text [DRAWING] in the script draft.
* Pictures (that require linking a file) and MathType formulas are NOT supported. For technical reasons these might never be supported.
* The $OS tag copies the content of the cell (in either blackboard or teacher column) into the OS file to be edited by the KE. Multiple $OS tags are allowed.
* You can add tables (copy from the “pieces” file) in the OS draft to add scripts.
* Every stage automatically starts a new lesson item with an implicit $new (without branching) tag.
* Texts which are ~~struck through~~ will not be copied into the script and OS drafts. This could be used when you want to partially cut the content in a cell.
* Teachers’ comments, often in the form of “[something something]” in the lesson plan will now appear as sidebar comments. This applies to the Blackboard, Teacher, and Student columns in the lesson plan.
* Sidebar comments in the lesson plan are now copied into the script drafts, above the content the comments are about, in the KE style.
* About the $new(par) tag:
  + The parameter “par” must be one of the following: empty, -, w, a, s, wa, ws, as, was; and any of the letter could be capitalized to indicate “skip if behind”:
    - $new=$new() : lesson item for all students.
    - $new(-) : lesson item for all students and labeled skip if behind.
    - par =
      * w : weak
      * a : average
      * s : strong
      * wa : two branches: weak; average or strong
      * ws : two branches: weak or average; strong
      * as : two branches: average; strong
      * was : three branches: weak; average; strong
      * If any of the at most three letters is capitalized, then the corresponding branch is labeled skip if behind. For example, $new(wA) generates two branches: weak; average or strong, skip if behind. Similarly, $new(WAS) generates three branches: weak, skip if behind; average, skip if behind; strong, skip if behind.
* $insa--$end and $insb--$end might behave strangely if combined with other tags such as $new or $cut. You are advised to compare with the original lesson plan where you use them.
* About the $insa--$end tags:
  + $insa should be placed in a row right AFTER a submit. (So, right after a row where the student cell does not have $cut/$tutor/$weak/$average/$strong.)
  + The content between $insa and its paired $end will be in the no response branch of that submit.
  + $end should be placed in the last row of content to be in the no response branch of that submit.
  + Between $insa and $end you can still use $cut/$tutor/$weak/$average/$strong/$OS.
* About the $insb--$end tags:
  + $end paired with an $insb tag should be placed in a row BEFORE a submit.
  + The content between $insb and its paired $end will be in the no response branch of that submit. (So the order or content is changed!)
  + $insb should be on the first row of content to be in the no response branch of that submit.
  + Between $insb and $end you can still use $cut/$tutor/$weak/$average/$strong/$OS.
* About the $branch(par)--$endbranch tags:
  + $branch generates an animator’s note saying that in-script branching begins.
  + $endbranch generate an animator’s note saying that in-script branching ends.
  + The content between $branch and $endbranch will go under the weakest branch.
  + The parameter par works in the exact way as in $new(par). For example, $branch(wA) generates two in-script branches: weak; average or strong, skip if behind. The content between $branch and $endbranch goes under the weak branch, and the average or strong, skip if behind branch will be blank with just an animator’s note.
  + $branch(par)--$endbranch and $insa/b--$end may be used together. Also, within the branch you can still use $cut/$tutor/$weak/$average/$strong/$OS.