# ETC Test Parser

## Features as of V1.0.0

**Current Features**

* Reads all .htm/html files from the "./HTML\_Files" directory.
* Parses the students name from the file
* Tallies all missed questions, by objectives per Mod
* Outputs two files into the "./Output\_Files" directory based on students name; a text file and .csv file
* Calculates the students Grades

## Usage

1. Go to "S:\Inst\3-Programming Fundamentals\Instructors\Brion\TestParser"
2. Choose which Mod you wish to parse (3 or 9) (Note: Currently Programming/Scripting Module objectives are prefixed with '9'.)
3. Save each test you wish to parse in the "S:\Inst\3-Programming Fundamentals\Instructors\Brion\TestParser\Mod#\HTML\_Files" folder
4. Right Click on the "testParser\_mod#.ps1", and choose "Run with PowerShell" from the menu.
5. A screen will pop up as the program runs, but will not pause. This is normal behavior.
6. Check out the "S:\Inst\3-Programming Fundamentals\Instructors\Brion\TestParser\Mod#\Output\_Files" folder for the results.
7. Review the text files, and then copy the csv files into the remediation file.
8. DONE!

## Versions/To DO

**To Do for Verison v1.1.0**

* Read in and parse information from configuration file.
* Format a proper configurations file for Mod 3/9
* ~~Allow user to select which module – this will thus merge the two current scripts~~. (1.1.0.dev.1)🡪 Currently hard coded in the base code until a configuration file usage added.
* **Test paser v0.0.0** - Initial Working Code Block that works on the Unit Test.

**To Do for Verison v1.0.0**

* ~~Add documentation to code~~
* ~~General Code clean up/uniformity~~
* ~~Add saved output - report file from last run.~~
* ~~Add support for CSV saves~~
* ~~Add support for multiple html files in the same directory~~
* Add Auto detecting for the Mod# in the objective --> Push to next version due to need to track # questions on the test and the # of objectives to track.

**To Do for Future Versions** (not in any order)

* Add Auto login feature
* Create a single file text file as an output as well.
* Add support to download tests from ETC server by group
* Turn base script into a cmdlet
* Turn parsing into a function/cmdlet
* Have objective names loaded automatically from the configuration file
* Add support for any ETC testing based mod.
* Add support for THREE different parsed objective numbers: (Objective : 3.9.2)
  + Module Number (3 or 9)
  + Module Objective Number (10, 1, 02)
  + Daily Objective Number (10, 1, 02)
* Change the outputs to reflect the following requirements:
  + Success stream – normal outputs
  + Error stream – a halting error when the command cannot continue
  + Warning stream – an unexpected situation/error but command can continue
  + Verbose stream – messages to help users/dev to understand current processing of the command. Most of the debug stream current messages need to be changed to verbose stream.
  + Debug stream – a detailed message needed only by the dev to see current state and micro processing details of the command
  + Information stream – not used (may be used as a halfway point between Success and Verbose.)
  + Progress stream – not used