This script contains functions to convert the *KnownColors Enum* into 3 ArrayLists with the names & (1-Based) ColorIndex values for the 174 *KnownColors*, further splitting the *KnownColors* into the subsets of 33 *SystemColors* and 141 *WebColors*. Functions are also provided for generating ArrayLists of the HSL and Argb values for the 3 color sets, and an Arraylist that contains PSObjects detailing all the information relating to a color set. The names in these ArrayLists may be used in a ForEach-Object Loop to get the corresponding [System.Drawing.Color] by use of commands like: $Color = [System.Drawing.Color]::FromName($_.Name). Using this ability a finaloption allows you to get an ArrayList of PSObjects that correspond to Powershell's native representation of a [System.Drawing.Color].

The *WebColor* values are static, while the SystemColors values will vary from PC to PC as their values reflect the active System Colors based on current color settings or Themes.

Parameter: *ColorSet* - Alias: CS

Select the Output Color Set: **Known**, **System**, or **Web**.

Parameter: *OutputType* - Alias: OT

Sets the data to be output in an ArrayList:

- **ColorSet:** Native Powershell Color Objects
- **ColorSetInfo:** Color Names & (1-Based) ColorIndex values
- **ColorInfoTable:** All Color Property Values
- **ArgbInHex:** Color Alpha, Red, Green, Blue channels as a single Hex value: AARRGGBB
- **HueList:** Color Names & HSL Hue values
- **SaturationList:** Color Names & HSL Saturation values
- **BrightnessList:** Color Names & HSL Lightness values

The following is a sample of the output from the **ColorInfoTable** option piped to a GridView: