```
with this///Â  Â Â # FUNCTIONs
 Â Â Â # with the list, calculate the length of the longest function name
 Â Â Â # Â Â add 2 for "()" to get true max displayed length
 Â Â Â _sh_fn_max_len=$(( $(get_longest $_sh_fn_list) + 2 ))
 Â Â Â d_message "_sh_fn_max_len: $_sh_fn_max_len" 4
 Â Â Â # determine the offset to the function description column - this is where the curs
or
 Â Â Â # Â Â will be when echo_n_long is called to smart-wrap the description
 Â Â Â # Â Â Math: _sh_fn_max_len + 2c.Lpad + 2c.dots(min) + 2c.Rpad
 Â Â Â _sh_fn_col=$((_sh_fn_max_len + 6))
 Â Â Â d_message "_sh_fn_col: $_sh_fn_col" 4
 Â Â Â # assign the number of the column to which to indent any subseqent lines of the de
scription
 Â Â Â _sh_fn_indent="${_sh_fn_col}" Â Â # for now, wrap/indent to align directly under s
tart point
 Â Â Â # assign the internal separator to appear between function name and padding
 Â Â Â _sh_fn_sep=" "
 Â Â Â # calculate length of the column (between the gutters) to be passed to pad()-->ech
o_n_long()
 Â Â Â _sh_fn_col_width=$(( _sh_fn_max_len + ${#_sh_fn_sep} - 1 ))

 Â Â Â # METADATA
 Â Â Â # calculate the length of the longest metadata lable to determine the prinf column
 width
 Â Â Â # Â Â for metadata (these are known a priori, but change) Â Â
 Â Â Â _sh_tag_list="usage arguments variables dependencies requirements"
 Â Â Â _sh_tag_list="${_sh_tag_list} rules notes warning unknown"
 Â Â Â _sh_tag_max_len=$(get_longest $_sh_tag_list)
 Â Â Â d_message "_sh_tag_list: $_sh_tag_list" 4
 Â Â Â d_message "_sh_tag_max_len: $_sh_tag_max_len" 4
 Â Â Â # determine the offset to the metadata payload column - this is where the cursor
 Â Â Â # Â Â will be when echo_n_long is called to smart-wrap the metadata payload
 Â Â Â # Â Â Math: _sh_tag_max_len + 4c.Lpad + 2c.dots(min) + 2c.Rpad
 Â Â Â _sh_tag_col=$((_sh_tag_max_len + 8))
 Â Â Â d_message "_sh_tag_col: $_sh_tag_col" 4
 Â Â Â # assign the number of the column to which to indent any subsequent lines of metad
ata payload
 Â Â Â _sh_tag_indent="${_sh_tag_col}" Â Â # for now, wrap/indent to align directly under
 start point
 Â Â Â # assign the internal separator to appear between metadata label and padding
 Â Â Â _sh_tag_sep=": "
 Â Â Â # calculate the length of the column (between the gutters) to be passed to pad()
 Â Â Â _sh_tag_col_width=$(( _sh_tag_max_len + ${#_sh_tag_sep} - 2 ))
///i get this///Â summarize_header
 * looking for headers in /home/joe/myUtilities/dev-util
/script_header_joetoo/testing/script_header_joetoo*

---[ bash-0.0.0 (19620207) (content summary) ]----------
 * This script header defines common variables, ANSI
 Â Â sequences, and the following functions/utilities as
 Â Â described below:

---[ bash-0.0.0 (19620207) (script_header_joetoo_testdum
my) ]
 Â toc() ............... Â table of contents (this fn
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â just calls
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â summarize_header
 Â Â Â notes: ....... Â this is a test line intended to wrap
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â describes the usage syntax, options,
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â and arguments for a
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â function
 Â Â Â notes: ....... Â lines below starting '#' followed by
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â '
 Â Â Â usage: ....... Â <empty>
 Â Â Â arguments: ... Â <empty>
 Â Â Â variables: ... Â <empty>
 Â Â Â dependencies: Â Â <empty>
 Â Â Â requirements: Â Â <empty>
 Â Â Â rules: ....... Â <empty>
 Â Â Â notes: ....... Â <empty>
 Â Â Â warning: ..... Â <empty>
 Â SCP() ............... Â (ANSI) save the current cursor
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â position
 Â RCP() ............... Â (ANSI) restore the cursor to
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â the saved position
 Â HCU() ............... Â (ANSI) Hide the cursor (Note:
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â the trailing character is
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â lowercase L)
 Â isnumber() .......... Â (POSIX) deprecated - use
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â isint(); tests if $1 is an
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â integer
 Â W_message() ......... Â (POSIX) display text warning
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@
 Â E_message() ......... Â (POSIX) display text error
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@
 Â E_message_n() ....... Â (POSIX) display text error
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@ (no CR)
 Â _d_build_and_print() . Â Internal helper: Not for
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â direct use.

 * To run functional tests, use: demonstrate_me
---[ gmki91 (complete) ]-------------------------------
joe@gmki91 ~/myUtilities/dev-util/script_header_joetoo $
 VERBOSE=$TRUE verbosity=5 summarize_header
 * looking for headers in /home/joe/myUtilities/dev-util
/script_header_joetoo/testing/script_header_joetoo*

---[ bash-0.0.0 (19620207) (content summary) ]----------
 * This script header defines common variables, ANSI
 Â Â sequences, and the following functions/utilities as
 Â Â described below:

---[ bash-0.0.0 (19620207) (script_header_joetoo_testdum
my) ]
 * _sh_fn_max_len: 20
 * _sh_fn_col: 26
 * _sh_tag_list: usage arguments variables dependencies
requirements rules notes warning unknown
 * _sh_tag_max_len: 12
 * _sh_tag_col: 20
 * _sh_line: toc() # table of contents (this fn just cal
ls summarize_header
 * _sh_fn_name: [toc()]
 Â toc() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 5
............... Â table of contents (this fn
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â just calls
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â summarize_header
 * _sh_line: # @note this is a test line intended to wra
p describes the usage syntax, options, and arguments for
 a function
 Â Â Â notes: ....... Â this is a test line intended to wrap
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â describes the usage syntax, options,
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â and arguments for a
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â function
 * _sh_line: # @note lines below starting '#' followed b
y ' @xxx' are interpreted by summarize_header as metadat
a
 Â Â Â notes: ....... Â lines below starting '#' followed by
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â '
 * _sh_line: # @usage @usage describes the usage syntax,
 options, and arguments for a function
 Â Â Â usage: ....... Â <empty>
 * _sh_line: # @args @args defines positional parameters
 ($1, $2, etc) and their roles
 Â Â Â arguments: ... Â <empty>
 * _sh_line: # @vars @vars identifies global variables r
equired or modified by the function
 Â Â Â variables: ... Â <empty>
 * _sh_line: # @deps @deps lists function, script, or pa
```

```
ckage dependencies
 Â Â Â dependencies: Â Â <empty>
 * _sh_line: # @req @req specifies prerequisites or stat
e requirements (e.g., must be root)
 Â Â Â requirements: Â Â <empty>
 * _sh_line: # @rule @rule outlines logic constraints or
 mandatory behaviors
 Â Â Â rules: ....... Â <empty>
 * _sh_line: # @note @note provides administrative or ge
neral notes for the user or maintainer
 Â Â Â notes: ....... Â <empty>
 * _sh_line: # @warn @warn provides administrative warni
ng for user or maintainer
 Â Â Â warning: ..... Â <empty>
 * _sh_line: SCP() Â Â Â # (ANSI) save the current cursor p
osition
 * _sh_fn_name: [SCP()]
 Â SCP() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 5
............... Â (ANSI) save the current cursor
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â position
 * _sh_line: RCP() Â Â Â # (ANSI) restore the cursor to the
 saved position
 * _sh_fn_name: [RCP()]
 Â RCP() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 5
............... Â (ANSI) restore the cursor to
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â the saved position
 * _sh_line: HCU() Â Â Â # (ANSI) Hide the cursor (Note: th
e trailing character is lowercase L)
 * _sh_fn_name: [HCU()]
 Â HCU() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 5
............... Â (ANSI) Hide the cursor (Note:
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â the trailing character is
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â lowercase L)
 * _sh_line: isnumber() Â Â Â Â Â Â Â Â Â # (POSIX) deprecated -
use isint(); tests if $1 is an integer
 * _sh_fn_name: [isnumber()]
 Â isnumber() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 10
.......... Â (POSIX) deprecated - use
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â isint(); tests if $1 is an
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â integer
 * _sh_line: W_message() Â Â # (POSIX) display text warnin
g message $@
 * _sh_fn_name: [W_message()]
 Â W_message() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 11
......... Â (POSIX) display text warning
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@
 * _sh_line: E_message() Â Â # (POSIX) display text error
message $@
 * _sh_fn_name: [E_message()]
 Â E_message() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 11
......... Â (POSIX) display text error
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@
 * _sh_line: E_message_n() Â Â Â Â Â Â # (POSIX) display text
error message $@ (no CR)
 * _sh_fn_name: [E_message_n()]
 Â E_message_n() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 13
....... Â (POSIX) display text error
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â message $@ (no CR)
 * _sh_line: _d_build_and_print() # Internal helper: Not
 for direct use.
 * _sh_fn_name: [_d_build_and_print()]
 Â _d_build_and_print() Â * _sh_fn_max_len: 20
 * #_sh_fn_name: 20
```

```
. Â Internal helper: Not for
 Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â Â direct use.

 * To run functional tests, use: demonstrate_me
---[ gmki91 (complete) ]--------------------------------
joe@gmki91 ~/myUtilities/dev-util/script_header_joetoo $
 x=" Â RCP() ................ Â "; echo ${#x}
26
joe@gmki91 ~/myUtilities/dev-util/script_header_joetoo $
 y=" Â Â Â notes: ....... Â "; echo ${#y}
20
/// can you explain why I basically have to undo what I thought was necessary to accurate
ly keep track of where the cursor is so I can tell echo_n_long  ?
```