```
# The uagame module provides support for creating a pygame window and
# simplifies graphical text output and input.
# A Window represents a display window with a title bar,close box
# and interior drawing surface.

Window(title, width, height) -> Window
# Creates and opens a window to draw in.
# - title is the str title of the window
# - width is the int pixel width of the window
# - height is the int pixel height of the window
```

Methods:

```
uagame.Window.close() -> None
# Closes the window

uagame.Window.set_font_name(name) -> None
# Set the name of the window font used to draw strings
# - name is the str name of the font

uagame.Window.set_font_size(point_size) -> None
# Set the point size of the window font used to draw strings
# - point_size is the int point size of the font

uagame.Window.set_font_color(color_string) -> None
# Set the font color used to draw in the window
# - color_string is the str name of the font color

uagame.Window.set_bg_color(color_string) -> None
# Set the background color used to draw in the window
# - color_string is the str name of the background color

uagame.Window.set_auto_update(true_false) -> None
# sets whether or not calls to draw_string will automatically
# update the screen or not.
# - true_or_false is a Boolean indicating if auto update should be
#   on or off; auto update is true by default.

uagame.Window.get_font_height() -> int
# Return the int pixel height of the current font.

uagame.Window.get_font_color() -> str
# Return a str that represents the current window font color
```

```
uagame.Window.get_bg_color() -> str
# Return a str that represents the current window background color

uagame.Window.get_width() -> int
# Return the int pixel width of the window's drawable surface

uagame.Window.get_height() -> int
# Return the int pixel height of the window's drawable surface

uagame.Window.clear() -> None
# Erase the window contents

uagame.Window.get_surface() -> pygame.Surface
# Return the Pygame.Surface object that represents the interior
# drawing surface of the window

uagame.Window.draw_string(string, x, y) -> None
# Draw a string in the window using the current font and colors
# if auto update is true, which is default window mode, the display
# is updated immediately; otherwise the display must be updated with
# a call to Window.update().
# - string is the str object to draw
# - x is the int x coord of the upper left corner of the string
#   string in the window
# - y is the int y coord of the upper left corner of the string
#   in the window

uagame.Window.input_string(prompt, x, y) -> str
# Draw a prompt string in the window using the current font and
# colors. Check keys pressed by the user until an enter key is
# pressed and return the sequence of key presses as a str object.
# - prompt is the str to display
# - x is the int x coord of the upper left corner of the
#   string in the window
# - y is the int y coord of the upper left corner of the
#   string in the window

uagame.Window.get_string_width(string) -> int
# Return the int pixel width of the string using the current font.
# - string is the str object

uagame.Window.update() -> None
```

```
# Update the window by copying all drawn objects from the frame
buffer
# to the display.
```