

Sounds, Fonts and Other Objects

All sound-related functions are found in `pygame.mixer`, including `init()`, which should be called before any other functions are invoked. Sounds themselves are in `pygame.mixer.Sound`. There are `play()`, `stop()`, `fadeout()` and other methods available for playing sounds. All sounds must be in OGG or WAV format. There are many programs and apps that will convert between different formats, or you can use an [online service](#) instead.

Like sounds, music can be played using functions in `pygame.mixer.music`. Here, there are `load()`, `play()`, `stop()` and `pause()` functions, among many others. Note that only one music file can be played at a time, unlike sounds.

Any text displayed to a surface must be loaded using functions in `pygame.font`. Like mixer, the font module must be initialized first using `init()`. After this, it is possible to `render()` a font using the methods available in `pygame.font.Font`.

Write programs that accomplish each task, using appropriate programming conventions. Download the file `sounds_fonts_files.zip` to help.

1. Load the font `Roboto-Regular.ttf`, and create a program that displays your name. Try experimenting with the size and colour of the text, to get a feel of how they appear.
2. Load the font `Roboto-Regular.ttf`, and create a program that displays the number of seconds that have elapsed since it started. Beginning at zero, the program should increase the number displayed by one each second. Use `pygame.time.delay` (measured in milliseconds) to help.
3. Modify your program above so that it plays the music `HoursOfFun.ogg` in the background.
4. A common screensaver displays a line of text, and moves it diagonally across the screen. When the edge of the text hits the boundary of the screen, it “bounces” off the boundary by reversing either its horizontal or vertical direction, but maintaining the other. For example, if the text is moving down and right and it reaches the bottom of the display, the text would bounce and continue up and right. Create a `Font` object and draw it to the surface at `(0, 0)`. Set its initial movement to down and right. Use `Font.size(TEXT)` to obtain the dimensions of the `Font` object, so that when it hits the edge of the surface, it will adjust its direction accordingly.
5. Modify your program above so that it plays the sound `hit.ogg` when the text hits the edge of the screen.