

Gura モジュールリファレンス - sdl

Updated: February 21, 2013

copyright © 2011- Yutaka Saito (ypsitau@nifty.com)

Official site: <http://www.gura-lang.org/>

目次

| | |
|---|---|
| 1. このリファレンスについて | 3 |
| 2. リファレンス | 4 |
| 2.1. 命名規則 | 4 |
| 2.2. モジュール関数 | 4 |
| 2.3. <code>sdl.Cursor</code> クラス | 7 |
| 2.3.1. | 7 |
| 2.3.2. インスタンスメソッド | 7 |
| 2.4. <code>sdl.Timer</code> クラス | 7 |
| 2.4.1. インスタンスメソッド | 7 |
| 2.5. <code>sdl.PixelFormat</code> クラス | 7 |
| 2.5.1. インスタンスメソッド | 7 |
| 2.6. <code>sdl.Surface</code> クラス | 7 |
| 2.6.1. インスタンスの生成 | 7 |
| 2.6.2. インスタンスメソッド | 8 |
| 2.7. <code>sdl.Overlay</code> クラス | 8 |
| 2.7.1. インスタンスメソッド | 8 |
| 2.8. <code>sdl.Joystick</code> クラス | 8 |
| 2.8.1. インスタンスメソッド | 8 |
| 2.9. <code>sdl.AudioSpec</code> クラス | 9 |
| 2.9.1. インスタンスメソッド | 9 |
| 2.10. <code>sdl.AudioCVT</code> クラス | 9 |
| 2.10.1. インスタンスメソッド | 9 |
| 2.11. <code>sdl.CD</code> クラス | 9 |
| 2.11.1. インスタンスメソッド | 9 |

1. このリファレンスについて

本リファレンスは **Gura** の `sdl` モジュールで定義されている関数やクラスの仕様について説明します。
SDL のオフィシャルサイトは <http://www.libsdl.org/> です。

2. リファレンス

2.1. 命名規則

2.2. モジュール関数

```

sdl.AddEvents(events[]:Event, mask:number)

sdl.AddTimer(interval:number, callback?:function):[thread] {block?}

sdl.AudioCVT(src_format:number,      src_channels:number,      src_rate:number,
             dst_format:number, dst_channels:number, dst_rate:number)

sdl.AudioSpec(freq:number => 22050, format:number => 32784, channels:number =>
             1, samples:number => 8192, callback?:function)

sdl.BlitSurface(src:Surface, srcrect, dst:Surface, dstrect)

sdl.BuildAudioCVT(src_format:number,  src_channels:number,  src_rate:number,
                 dst_format:number, dst_channels:number, dst_rate:number)

sdl.CDName(drive:number):map

sdl.CDNumDrives()

sdl.CDOpen(drive:number)

sdl.CheckKeyState(key:number):map

sdl.CloseAudio():void

sdl.Color(r:number, g:number, b:number):map

sdl.CreateCursor(data:binary, mask:binary, w:number, h:number, hot_x:number,
                hot_y:number)

sdl.CreateRGBSurface(flags:number, width:number, height:number, depth:number,
                    Rmask:number, Gmask:number, Bmask:number, Amask:number) {block?}

sdl.CreateRGBSurfaceFrom(image:image) {block?}

sdl.CreateYUVOverlay(width:number,      height:number,      format:number,
                    display:Surface)

sdl.Delay(ms:number):void

```

`sdl.EnableKeyRepeat(delay:number, interval:number)`

`sdl.EnableUNICODE(enable:number)`

`sdl.EventState(type:number, state:number)`

`sdl.GL_GetAttribute(attr:number)`

`sdl.GL_SetAttribute(attr:number, value:number)`

`sdl.GL_SwapBuffers():void`

`sdl.GetAppState()`

`sdl.GetAudioStatus()`

`sdl.GetCursor()`

`sdl.GetError()`

`sdl.GetEventFilter()`

`sdl.GetEvents(numevents:number, mask:number)`

`sdl.GetGammaRamp()`

`sdl.GetKeyName(key:number)`

`sdl.GetModState()`

`sdl.GetMouseState()`

`sdl.GetRelativeMouseState()`

`sdl.GetTicks()`

`sdl.GetVideoInfo()`

`sdl.GetVideoSurface() {block?}`

`sdl.Init(flags:number)`

`sdl.InitSubSystem(flags:number)`

`sdl.JoystickEventState(state:number)`

`sdl.JoystickName(index:number):map`

`sdl.JoystickOpen(index:number):map`

`sdl.JoystickOpened(index:number):map`

```
sdl.JoystickUpdate():void

sdl.ListModes()

sdl.LoadBMP(file:string) {block?}

sdl.LoadImage(file:string) {block?}

sdl.LoadWAV(file:string)

sdl.LockAudio():void

sdl.NumJoysticks()

sdl.OpenAudio(desired:AudioSpec)

sdl.PauseAudio(pause_on:number):void

sdl.PeekEvents(numevents:number, mask:number)

sdl.PollEvent()

sdl.PumpEvents():void

sdl.PushEvent(event:Event)

sdl.Quit():void

sdl.QuitSubSystem(flags:number):void

sdl.Rect(x:number, y:number, w:number, h:number):map

sdl.SetCursor(cursor:Cursor):void

sdl.SetEventFilter(filter:function)

sdl.SetGamma(redgamma:number, greengamma:number, bluegamma:number)

sdl.SetGammaRamp(redtable[:number, greentable[:number, bluetable[:number)

sdl.SetModState(modstate:number):void

sdl.SetVideoMode(width:number, height:number, bpp:number, flags:number)
    {block?}

sdl.ShowCursor(toggle:number)

sdl.UnlockAudio():void

sdl.VideoDriverName()

sdl.VideoModeOK(width:number, height:number, bpp:number, flags:number)
```

```
sdl.WM_GetCaption()  
  
sdl.WM_GrabInput(mode:number)  
  
sdl.WM_IconifyWindow()  
  
sdl.WM_SetCaption(title:string, icon:string):void  
  
sdl.WM_SetIcon(surface:Surface, mask?:binary)  
  
sdl.WM_ToggleFullScreen(surface:Surface)  
  
sdl.WaitEvent()  
  
sdl.WarpMouse(x:number, y:number):void  
  
sdl.WasInit(flags:number)
```

2.3. sdl.Cursor クラス

2.3.1.

2.3.2. インスタンスメソッド

```
sdl.Cursor#FreeCursor():void
```

2.4. sdl.Timer クラス

2.4.1. インスタンスメソッド

```
sdl.Timer#RemoveTimer()
```

2.5. sdl.PixelFormat クラス

2.5.1. インスタンスメソッド

```
sdl.PixelFormat#GetRGB(pixel:number)  
sdl.PixelFormat#GetRGBA(pixel:number)  
sdl.PixelFormat#MapRGB(r:number, g:number, b:number)  
sdl.PixelFormat#MapRGBA(r:number, g:number, b:number, a:number)
```

2.6. sdl.Surface クラス

2.6.1. インスタンスの生成

```
sdl.CreateRGBSurface(flags:number, width:number, height:number, depth:number,  
    Rmask:number, Gmask:number, Bmask:number, Amask:number) {block?}  
  
sdl.CreateRGBSurfaceFrom(image:image) {block?}
```

2.6.2. インスタンスメソッド

```
sdl.Surface#ConvertSurface(fmt:PixelFormat, flag:number) {block?}
sdl.Surface#DisplayFormat() {block?}
sdl.Surface#DisplayFormatAlpha() {block?}
sdl.Surface#FillRect(rect:Rect, color:Color):map:void
sdl.Surface#Flip()
sdl.Surface#GetClipRect()
sdl.Surface#LockSurface()
sdl.Surface#PutSurface(src:Surface, x:number => 0, y:number => 0):map
sdl.Surface#SaveBMP(file:string):void
sdl.Surface#SetAlpha(flag:number, alpha:number)
sdl.Surface#SetClipRect(rect:Rect):map:void
sdl.Surface#SetColorKey(flag:number, key:number)
sdl.Surface#SetColors(colors[]:Color, firstcolor:number => 0)
sdl.Surface#SetPalette(flags:number, colors[]:Color, firstcolor:number => 0)
sdl.Surface#UnlockSurface():void
sdl.Surface#UpdateRect(x:number => 0, y:number => 0, w:number => 0, h:number => 0):void
sdl.Surface#UpdateRects(rects[]:Rect):void
```

2.7. sdl.Overlay クラス

2.7.1. インスタンスメソッド

```
sdl.Overlay#DisplayYUVOverlay(dstrect:Rect)
sdl.Overlay#LockYUVOverlay()
sdl.Overlay#UnlockYUVOverlay():void
```

2.8. sdl.Joystick クラス

2.8.1. インスタンスメソッド

```
sdl.Joystick#JoystickClose():void
sdl.Joystick#JoystickGetAxis(axis:number)
sdl.Joystick#JoystickGetBall(ball:number)
sdl.Joystick#JoystickGetButton(button:number)
sdl.Joystick#JoystickGetHat(hat:number)
sdl.Joystick#JoystickIndex()
sdl.Joystick#JoystickNumAxes()
sdl.Joystick#JoystickNumBalls()
```



```
sdl.Joystick#JoystickNumButtons()
```

```
sdl.Joystick#JoystickNumHats()
```

2.9. sdl.AudioSpec クラス

2.9.1. インスタンスメソッド

```
sdl.AudioSpec#MixAudio(src:AudioSpec, volume:number)
```

2.10. sdl.AudioCVT クラス

2.10.1. インスタンスメソッド

```
sdl.AudioCVT#ConvertAudio()
```

2.11. sdl.CD クラス

2.11.1. インスタンスメソッド

```
sdl.CD#CDClose():void
```

```
sdl.CD#CDEject()
```

```
sdl.CD#CDPause()
```

```
sdl.CD#CDPlay(start:number, length:number)
```

```
sdl.CD#CDPlayTracks(start_track:number, start_frame:number, ntracks:number,  
    nframes:number)
```

```
sdl.CD#CDResume()
```

```
sdl.CD#CDStatus()
```

```
sdl.CD#CDStop()
```

```
sdl.CD#GetTrack(n:number):void
```