

## Gura モジュールリファレンス - opengl

Updated: February 21, 2013

copyright © 2011- Yutaka Saito ([ypsitau@nifty.com](mailto: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. opengl モジュール .....	4
2.1. 命名規則.....	4
2.2. モジュール関数.....	4
3. glu モジュール.....	18
3.1. 命名規則.....	18
3.2. モジュール関数.....	18

## 1. このリファレンスについて

本リファレンスは **Gura** の `opengl` モジュールおよび `glu` モジュールで定義されている関数の一覧を掲載しています。関数の詳細な動作は、**OpenGL** のドキュメントを参照ください。

**OpenGL** のオフィシャルサイトは <http://www.opengl.org/> です。

## 2. opengl モジュール

### 2.1. 命名規則

OpenGL の C 言語の関数はすべて `gl` で始まる名前がつけられており、`opengl` モジュールの関数も同じ名前を割り当てています。この関数名はベースの名前空間で他の要素と衝突する可能性が小さいため、以下のよう  
にインポートして使用するとモジュール名をつける必要がなくなり便利です。

```
import (opengl) {*}
```

### 2.2. モジュール関数

```
glAccum(op:number, value:number):map:void
glAlphaFunc(func:number, ref:number):map:void
glAreTexturesResident(textures[:number])
glArrayElement(i:number):map:void
glBegin(mode:number):void {block?}
glBindTexture(target:number, texture:number):map:void
glBitmap(width:number, height:number, xorig:number, yorig:number, xmove:number,
         ymove:number, bitmap:binary):void
glBlendFunc(sfactor:number, dfactor:number):map:void
glCallList(list:number):map:void
glCallLists(type:number, lists[:number]):void
glClear(mask:number):map:void
glClearAccum(red:number, green:number, blue:number, alpha:number):map:void
glClearColor(red:number, green:number, blue:number, alpha:number):map:void
glClearDepth(depth:number):map:void
glClearIndex(c:number):map:void
glClearStencil(s:number):map:void
glClipPlane(plane:number, equation[:number]):map:void
glColor3b(red:number, green:number, blue:number):map:void
glColor3bv(v[:number]):map:void
```

## Gura モジュールリファレンス – opengl

```
glColor3d(red:number, green:number, blue:number):map:void  
glColor3dv(v[:number]):map:void  
glColor3f(red:number, green:number, blue:number):map:void  
glColor3fv(v[:number]):map:void  
glColor3i(red:number, green:number, blue:number):map:void  
glColor3iv(v[:number]):map:void  
glColor3s(red:number, green:number, blue:number):map:void  
glColor3sv(v[:number]):map:void  
glColor3ub(red:number, green:number, blue:number):map:void  
glColor3ubv(v[:number]):map:void  
glColor3ui(red:number, green:number, blue:number):map:void  
glColor3uiv(v[:number]):map:void  
glColor3us(red:number, green:number, blue:number):map:void  
glColor3usv(v[:number]):map:void  
glColor4b(red:number, green:number, blue:number, alpha:number):map:void  
glColor4bv(v[:number]):map:void  
glColor4d(red:number, green:number, blue:number, alpha:number):map:void  
glColor4dv(v[:number]):map:void  
glColor4f(red:number, green:number, blue:number, alpha:number):map:void  
glColor4fv(v[:number]):map:void  
glColor4i(red:number, green:number, blue:number, alpha:number):map:void  
glColor4iv(v[:number]):map:void  
glColor4s(red:number, green:number, blue:number, alpha:number):map:void  
glColor4sv(v[:number]):map:void  
glColor4ub(red:number, green:number, blue:number, alpha:number):map:void  
glColor4ubv(v[:number]):map:void
```

## Gura モジュールリファレンス – opengl

`glColor4ui(red:number, green:number, blue:number, alpha:number):map:void`

`glColor4uiv(v[:number]):map:void`

`glColor4us(red:number, green:number, blue:number, alpha:number):map:void`

`glColor4usv(v[:number]):map:void`

`glColorMask(red:boolean, green:boolean, blue:boolean, alpha:boolean):map:void`

`glColorMaterial(face:number, mode:number):map:void`

`glCopyPixels(x:number, y:number, width:number, height:number, type:number):map:void`

`glCopyTexImage1D(target:number, level:number, internalFormat:number, x:number, y:number, width:number, border:number):map:void`

`glCopyTexImage2D(target:number, level:number, internalFormat:number, x:number, y:number, width:number, height:number, border:number):map:void`

`glCopyTexSubImage1D(target:number, level:number, xoffset:number, x:number, y:number, width:number):map:void`

`glCopyTexSubImage2D(target:number, level:number, xoffset:number, yoffset:number, x:number, y:number, width:number, height:number):map:void`

`glCullFace(mode:number):map:void`

`glDeleteLists(list:number, range:number):map:void`

`glDeleteTextures(textures[:number]):void`

`glDepthFunc(func:number):map:void`

`glDepthMask(flag:boolean):map:void`

`glDepthRange(zNear:number, zFar:number):map:void`

`glDisable(cap:number):map:void`

`glDisableClientState(array:number):map:void`

`glDrawArrays(mode:number, first:number, count:number):map:void`

`glDrawBuffer(mode:number):map:void`

`glDrawPixels(image:image):void`

`glEdgeFlag(flag:boolean):map:void`

```
glEdgeFlagv(flag[]:boolean):map:void

glEnable(cap:number):map:void

glEnableClientState(array:number):map:void

glEnd():void

glEndList():void

glEvalCoord1d(u:number):map:void

glEvalCoord1dv(u[]:number):map:void

glEvalCoord1f(u:number):map:void

glEvalCoord1fv(u[]:number):map:void

glEvalCoord2d(u:number, v:number):map:void

glEvalCoord2dv(u[]:number):map:void

glEvalCoord2f(u:number, v:number):map:void

glEvalCoord2fv(u[]:number):map:void

glEvalMesh1(mode:number, i1:number, i2:number):map:void

glEvalMesh2(mode:number, i1:number, i2:number, j1:number, j2:number):map:void

glEvalPoint1(i:number):map:void

glEvalPoint2(i:number, j:number):map:void

glFeedbackBuffer(size:number, type:number, buffer[]:number):void

glFinish():void

glFlush():void

glFogf(pname:number, param:number):map:void

glFogfv(pname:number, params[]:number):map:void

glFogi(pname:number, param:number):map:void

glFogiv(pname:number, params[]:number):map:void

glFrontFace(mode:number):map:void

glFrustum(left:number, right:number, bottom:number, top:number, zNear:number,
          zFar:number):map:void
```

`glGenLists(range:number):map`

`glGenTextures(n:number)`

`glGetBooleanv(pname:number):map`

`glGetClipPlane(plane:number):map`

`glGetDoublev(pname:number):map`

`glGetError()`

`glGetFloatv(pname:number):map`

`glGetIntegerv(pname:number):map`

`glGetLightfv(light:number, pname:number):map`

`glGetLightiv(light:number, pname:number):map`

`glGetMapdv(target:number, query:number)`

`glGetMapfv(target:number, query:number)`

`glGetMapiv(target:number, query:number)`

`glGetMaterialfv(face:number, pname:number)`

`glGetMaterialiv(face:number, pname:number)`

`glGetPixelMapfv(map:number, values[:number])`

`glGetPixelMapuiv(map:number, values[:number])`

`glGetPixelMapusv(map:number, values[:number])`

`glGetPointerv(pname:number, params[:number])`

`glGetPolygonStipple(mask[:number])`

`glGetString(name:number):map`

`glGetTexEnvfv(target:number, pname:number, params[:number])`

`glGetTexEnviv(target:number, pname:number, params[:number])`

`glGetTexGendv(coord:number, pname:number, params[:number])`

`glGetTexGenfv(coord:number, pname:number, params[:number])`

`glGetTexGeniv(coord:number, pname:number, params[:number])`



## Gura モジュールリファレンス – opengl

```
glGetTexImage(target:number, level:number, format:number, type:number,
  pixels[]:number)

glGetTexLevelParameterfv(target:number, level:number, pname:number,
  params[]:number)

glGetTexLevelParameteriv(target:number, level:number, pname:number,
  params[]:number)

glGetTexParameterfv(target:number, pname:number, params[]:number)

glGetTexParameteriv(target:number, pname:number, params[]:number)

glHint(target:number, mode:number):map:void

glIndexMask(mask:number):map:void

glIndexd(c:number):map:void

glIndexdv(c[]:number):map:void

glIndexf(c:number):map:void

glIndexfv(c[]:number):map:void

glIndexi(c:number):map:void

glIndexiv(c[]:number):map:void

glIndexs(c:number):map:void

glIndexsv(c[]:number):map:void

glIndexub(c:number):map:void

glIndexubv(c[]:number):map:void

glInitNames():void

glIsEnabled(cap:number):map:void

glIsList(list:number):map:void

glIsTexture(texture:number):map:void

glLightModelf(pname:number, param:number):map:void

glLightModelfv(pname:number, params[]:number):map:void

glLightModeli(pname:number, param:number):map:void
```

## Gura モジュールリファレンス – opengl

```
glLightModeliv(pname:number, params[:number]):map:void  
glLightf(light:number, pname:number, param:number):map:void  
glLightfv(light:number, pname:number, params[:number]):map:void  
glLighti(light:number, pname:number, param:number):map:void  
glLightiv(light:number, pname:number, params[:number]):map:void  
glLineStipple(factor:number, pattern:number):map:void  
glLineWidth(width:number):map:void  
glListBase(base:number):map:void  
glLoadIdentity():void  
glLoadMatrixd(m[:number]):map:void  
glLoadMatrixf(m[:number]):map:void  
glLoadName(name:number):map:void  
glLogicOp(opcode:number):map:void  
glMap1d(target:number, u1:number, u2:number, stride:number, order:number,  
  points[:number]):map:void  
glMap1f(target:number, u1:number, u2:number, stride:number, order:number,  
  points[:number]):map:void  
glMap2d(target:number, u1:number, u2:number, ustride:number, uorder:number,  
  v1:number, v2:number, vstride:number, vorder:number,  
  points[:number]):map:void  
glMap2f(target:number, u1:number, u2:number, ustride:number, uorder:number,  
  v1:number, v2:number, vstride:number, vorder:number,  
  points[:number]):map:void  
glMapGrid1d(un:number, u1:number, u2:number):map:void  
glMapGrid1f(un:number, u1:number, u2:number):map:void  
glMapGrid2d(un:number, u1:number, u2:number, vn:number, v1:number,  
  v2:number):map:void  
glMapGrid2f(un:number, u1:number, u2:number, vn:number, v1:number,  
  v2:number):map:void
```

## Gura モジュールリファレンス – opengl

`glMaterialf(face:number, pname:number, param:number):map:void`

`glMaterialfv(face:number, pname:number, params[:number]):map:void`

`glMateriali(face:number, pname:number, param:number):map:void`

`glMaterialiv(face:number, pname:number, params[:number]):map:void`

`glMatrixMode(mode:number):map:void`

`glMultMatrixd(m[:number]):map:void`

`glMultMatrixf(m[:number]):map:void`

`glNewList(list:number, mode:number):map:void {block?}`

`glNormal3b(nx:number, ny:number, nz:number):map:void`

`glNormal3bv(v[:number]):map:void`

`glNormal3d(nx:number, ny:number, nz:number):map:void`

`glNormal3dv(v[:number]):map:void`

`glNormal3f(nx:number, ny:number, nz:number):map:void`

`glNormal3fv(v[:number]):map:void`

`glNormal3i(nx:number, ny:number, nz:number):map:void`

`glNormal3iv(v[:number]):map:void`

`glNormal3s(nx:number, ny:number, nz:number):map:void`

`glNormal3sv(v[:number]):map:void`

`glOrtho(left:number, right:number, bottom:number, top:number, zNear:number, zFar:number):map:void`

`glPassThrough(token:number):map:void`

`glPixelMapfv(map:number, mapsize:number, values[:number]):map:void`

`glPixelMapuiv(map:number, mapsize:number, values[:number]):map:void`

`glPixelMapusv(map:number, mapsize:number, values[:number]):map:void`

`glPixelStoref(pname:number, param:number):map:void`

`glPixelStorei(pname:number, param:number):map:void`

`glPixelTransferf(pname:number, param:number):map:void`

## Gura モジュールリファレンス – opengl

```
glPixelTransferi(pname:number, param:number):map:void  
glPixelZoom(xfactor:number, yfactor:number):map:void  
glPointSize(size:number):map:void  
glPolygonMode(face:number, mode:number):map:void  
glPolygonOffset(factor:number, units:number):map:void  
glPolygonStipple(mask:binary):void  
glPopAttrib():void  
glPopClientAttrib():void  
glPopMatrix():void  
glPopName():void  
glPrioritizeTextures(textures[:number], priorities[:number]):void  
glPushAttrib(mask:number):void {block?}  
glPushClientAttrib(mask:number):void {block?}  
glPushMatrix():void {block?}  
glPushName(name:number):void {block?}  
glRasterPos2d(x:number, y:number):map:void  
glRasterPos2dv(v[:number]):map:void  
glRasterPos2f(x:number, y:number):map:void  
glRasterPos2fv(v[:number]):map:void  
glRasterPos2i(x:number, y:number):map:void  
glRasterPos2iv(v[:number]):map:void  
glRasterPos2s(x:number, y:number):map:void  
glRasterPos2sv(v[:number]):map:void  
glRasterPos3d(x:number, y:number, z:number):map:void  
glRasterPos3dv(v[:number]):map:void  
glRasterPos3f(x:number, y:number, z:number):map:void
```

## Gura モジュールリファレンス – opengl

```
glRasterPos3fv(v[]:number):map:void  
glRasterPos3i(x:number, y:number, z:number):map:void  
glRasterPos3iv(v[]:number):map:void  
glRasterPos3s(x:number, y:number, z:number):map:void  
glRasterPos3sv(v[]:number):map:void  
glRasterPos4d(x:number, y:number, z:number, w:number):map:void  
glRasterPos4dv(v[]:number):map:void  
glRasterPos4f(x:number, y:number, z:number, w:number):map:void  
glRasterPos4fv(v[]:number):map:void  
glRasterPos4i(x:number, y:number, z:number, w:number):map:void  
glRasterPos4iv(v[]:number):map:void  
glRasterPos4s(x:number, y:number, z:number, w:number):map:void  
glRasterPos4sv(v[]:number):map:void  
glReadBuffer(mode:number):map:void  
glReadPixels(x:number, y:number, width:number, height:number, format:symbol)  
glRectd(x1:number, y1:number, x2:number, y2:number):map:void  
glRectdv(v1[]:number, v2[]:number):void  
glRectf(x1:number, y1:number, x2:number, y2:number):map:void  
glRectfv(v1[]:number, v2[]:number):void  
glRecti(x1:number, y1:number, x2:number, y2:number):map:void  
glRectiv(v1[]:number, v2[]:number):void  
glRects(x1:number, y1:number, x2:number, y2:number):map:void  
glRectsv(v1[]:number, v2[]:number):void  
glRenderMode(mode:number):map:void  
glRotated(angle:number, x:number, y:number, z:number):map:void  
glRotatef(angle:number, x:number, y:number, z:number):map:void
```

```
glScaled(x:number, y:number, z:number):map:void
glScalef(x:number, y:number, z:number):map:void
glScissor(x:number, y:number, width:number, height:number):map:void
glSelectBuffer(size:number)
glShadeModel(mode:number):map:void
glStencilFunc(func:number, ref:number, mask:number):map:void
glStencilMask(mask:number):map:void
glStencilOp(fail:number, zfail:number, zpass:number):map:void
glTexCoord1d(s:number):map:void
glTexCoord1dv(v[:number]):map:void
glTexCoord1f(s:number):map:void
glTexCoord1fv(v[:number]):map:void
glTexCoord1i(s:number):map:void
glTexCoord1iv(v[:number]):map:void
glTexCoord1s(s:number):map:void
glTexCoord1sv(v[:number]):map:void
glTexCoord2d(s:number, t:number):map:void
glTexCoord2dv(v[:number]):map:void
glTexCoord2f(s:number, t:number):map:void
glTexCoord2fv(v[:number]):map:void
glTexCoord2i(s:number, t:number):map:void
glTexCoord2iv(v[:number]):map:void
glTexCoord2s(s:number, t:number):map:void
glTexCoord2sv(v[:number]):map:void
glTexCoord3d(s:number, t:number, r:number):map:void
glTexCoord3dv(v[:number]):map:void
```

```
glTexCoord3f(s:number, t:number, r:number):map:void  
glTexCoord3fv(v[:number]):map:void  
glTexCoord3i(s:number, t:number, r:number):map:void  
glTexCoord3iv(v[:number]):map:void  
glTexCoord3s(s:number, t:number, r:number):map:void  
glTexCoord3sv(v[:number]):map:void  
glTexCoord4d(s:number, t:number, r:number, q:number):map:void  
glTexCoord4dv(v[:number]):map:void  
glTexCoord4f(s:number, t:number, r:number, q:number):map:void  
glTexCoord4fv(v[:number]):map:void  
glTexCoord4i(s:number, t:number, r:number, q:number):map:void  
glTexCoord4iv(v[:number]):map:void  
glTexCoord4s(s:number, t:number, r:number, q:number):map:void  
glTexCoord4sv(v[:number]):map:void  
glTexEnvf(target:number, pname:number, param:number):map:void  
glTexEnvfv(target:number, pname:number, params[:number]):map:void  
glTexEnvi(target:number, pname:number, param:number):map:void  
glTexEnviv(target:number, pname:number, params[:number]):map:void  
glTexGend(coord:number, pname:number, param:number):map:void  
glTexGendv(coord:number, pname:number, params[:number]):map:void  
glTexGenf(coord:number, pname:number, param:number):map:void  
glTexGenfv(coord:number, pname:number, params[:number]):map:void  
glTexGeni(coord:number, pname:number, param:number):map:void  
glTexGeniv(coord:number, pname:number, params[:number]):map:void  
glTexImage1D(target:number, level:number, internalformat:number, border:number,  
    image:image):void  
glTexImage2D(target:number, level:number, internalformat:number, border:number,
```

```
image:image):void

glTexParameterf(target:number, pname:number, param:number):map:void

glTexParameterfv(target:number, pname:number, params[:number]):map:void

glTexParameteri(target:number, pname:number, param:number):map:void

glTexParameteriv(target:number, pname:number, params[:number]):map:void

glTexSubImage1D(target:number, level:number, xoffset:number, image:image):void

glTexSubImage2D(target:number, level:number, xoffset:number, yoffset:number,
    image:image):void

glTranslated(x:number, y:number, z:number):map:void

glTranslatef(x:number, y:number, z:number):map:void

glVertex2d(x:number, y:number):map:void

glVertex2dv(v[:number]):map:void

glVertex2f(x:number, y:number):map:void

glVertex2fv(v[:number]):map:void

glVertex2i(x:number, y:number):map:void

glVertex2iv(v[:number]):map:void

glVertex2s(x:number, y:number):map:void

glVertex2sv(v[:number]):map:void

glVertex3d(x:number, y:number, z:number):map:void

glVertex3dv(v[:number]):map:void

glVertex3f(x:number, y:number, z:number):map:void

glVertex3fv(v[:number]):map:void

glVertex3i(x:number, y:number, z:number):map:void

glVertex3iv(v[:number]):map:void

glVertex3s(x:number, y:number, z:number):map:void

glVertex3sv(v[:number]):map:void

glVertex4d(x:number, y:number, z:number, w:number):map:void
```



`glVertex4dv(v[]:number):map:void`

`glVertex4f(x:number, y:number, z:number, w:number):map:void`

`glVertex4fv(v[]:number):map:void`

`glVertex4i(x:number, y:number, z:number, w:number):map:void`

`glVertex4iv(v[]:number):map:void`

`glVertex4s(x:number, y:number, z:number, w:number):map:void`

`glVertex4sv(v[]:number):map:void`

`glViewport(x:number, y:number, width:number, height:number):map:void`

### 3. glu モジュール

#### 3.1. 命名規則

GLU ライブラリの C 言語の関数はすべて `glu` で始まる名前がつけられており、`glu` モジュールの関数も同じ名前を割り当てています。この関数名はベースの名前空間で他の要素と衝突する可能性が小さいため、以下のようにインポートして使用すると便利です。

```
import (glu) {*}
```

#### 3.2. モジュール関数

```
gluBuild1DMipmaps(target:number, internalFormat:number, image:image):map
```

```
gluBuild2DMipmaps(target:number, internalFormat:number, image:image):map
```

```
gluErrorString():void
```

```
gluErrorStringWIN():void
```

```
gluErrorUnicodeStringEXT():void
```

```
gluGetString():void
```

```
gluLookAt(eyex:number, eyey:number, eyez:number,
          centerx:number, centery:number, centerz:number,
          upx:number, upy:number, upz:number):void
```

```
gluNewNurbsRenderer()
```

```
gluNewQuadric()
```

```
gluNewTess()
```

```
gluOrtho2D(left:number, right:number, bottom:number, top:number):void
```

```
gluPerspective(fovy:number, aspect:number, zNear:number, zFar:number):void
```

```
gluPickMatrix(x:number, y:number, width:number, height:number,
              viewport[:]:number):void
```

```
gluProject(objX:number, objY:number, objZ:number, model:number, proj:number,
           view:number)
```

```
gluScaleImage(image:image, w:number, h:number)
```

```
gluUnProject(winX:number, winY:number, winZ:number, model:number, proj:number,
             view:number)
```