L04 - Windowing APIs

- although OpenGL is "Platform independant", you need an open window to initialize an OpenGL canvas on
  ↳ windowing API's are Platform dependant.

Workflow:
- Open an OS window
- Initialize an OpenGL canvas
- Capture input devices
  - mouse, keyboad, controller, etc.

GLUT eg.

```
#include <GL/glut.h>  // different for mac

int main() {
    //GLUT initializations
    ...
    // OpenGL initializations
    ...
    // Other initializations
    ...

    glutMainLoop();  //no callbacks will be called until here

    return 0;
}
```

- glutInit(&argc, argv)
- glutCreateWindow(" ")
- glutDisplayFunc(_)
- glutIdleFunc(_) ...

- glClearColor(0.0, 0.0, 0.0)
- Create buffers
- Create textures
- Compile Shaders ...

Display Func eg.

```
Void myDisplay() {
  // clear viewPort
  glClear( <params | param | param |... etc> );

  // OpenGL Draw Calls

  // swap Buffers
  glutSwapBuffers();
```

Double Buffering

- front Buffer → Displayed to user
- back Buffer → not visible
- while front buffer is being displayed, use the back buffer to render new frame
  - ↳ then swap & repeat
  - ↳ this way, we never display an incomplete frame

Non-Blocking Calls

- most gl commands are non-blocking
  - ↳ queues up instructions to GPU, then may have some idle time
  - ↳ this when the Idle func. is called.

Idle Func eg.

```
void myIdle() {
    // some animation

    // tell GLUT to redraw
    glutPostRedisplay(); → don't call display() yourself!!
}
```