

Cross-Platform & Native Windows

* Working with Windows

⇒ The HighGUI user input tools support only three basic interactions:

- Key presses
- mouse clicks
- Use of Simple trackbar.

⇒ The main advantages of the native tools are that they are fast and easy to use, and don't require you to install any additional libraries.

* Creating a window with cv::namedWindow()

```
int cv::namedWindow(  
    const string& name,  
    int flag = 0  
);
```

⇒ **name** appears at the top of the window, and is also used as a handle for the window that can be passed to other HighGUI functions.

⇒ flag argument indicates whether the window should autosize itself to fit an image we put onto it.

flag \rightarrow 0 { User able to resize }
 \rightarrow cv::WINDOW_AUTOSIZE

{ User cannot resize }

\Rightarrow Getting rid of the window:-

```
int cv::destroyWindow(  
    const string& name,  
);
```

* Drawing an image with cv::imshow()

```
void cv::imshow(  
    const string& name,  
    cv::InputArray image  
);
```

* Updating a Window and cv::waitKey()

```
int cv::waitKey(  
    int delay = 0 // Milliseconds  
    { 0 = "never" }  
);
```

~~_____~~