

## Lyhyt ohje ImageJ -pluginien tekemiseen

Tehdään esimerkkinä plugin, joka tekee harmaasävykuvasta negatiivin.

1. Valitse ImageJ:n valikoista: **Plugings** → **New...** → **Plugin Filter**.

ImageJ luo pohjan luokalle, joka toteuttaa rajapinnan `PuginFilter` sisältäen metodit `setup` ja `run`.

2. Aseta luokan nimi: `Invertteri_` (Kun Pluginin nimi päättyy alaviivaan ”\_”, ImageJ lisää sen automaattisesti valikkoon uudelleenkäynnistettäessä.)

3. `setup`-metodissa asetaan `ImagePlus`-olio, nyt nimeltään `imp`. Metodi palauttaa lisäksi tiedon, minkä tyyppisiä kuvia `run`-metodi käsittelee. Palautetaan vakio `DOES_8G`, jolloin muita kuin 8-bitin harmaasävykuvia ei voi antaa käsiteltäväksi.

4. Toteutetaan haluttu kuvankäsittelyoperaatio nyt kokonaan `run`-metodiin, josta voitaisiin toki kutsua myös muita metodeja. `run`-metodi saa ImageJ:ltä parametrina `ImageProcessor`-olion, joka sisältää varsinaisen kuvan sekä muuta tietoa. Jos `ImageProcessor`-olion nimi olisi vaikka `ip`, saadaan kuvan koko selville seuraavasti.

```
int W = ip.getWidth(); // Leveys  
int H = ip.getHeight(); // Korkeus
```

Yksittäisen pikselin arvo luetaan seuraavasti: `ip.getPixel(x,y)`;

Pikselin arvo asetetaan puolestaan: `ip.putPixel(x,y, <arvo>)`;

Kuva päivitetään `ImagePlus`-olion avulla: `imp.updateAndDraw()`;

5. Tallennetaan tehty luokka:

- File → Save As ...
- Luodaan [...]\\ImageJ\\plugins-hakemistoon hakemisto omille plugineille.
- Tallennetaan plugin nimellä `Invertteri_.java`

6. Avataan sopiva kuva. Muutetaan kuva tarvittaessa 8-bittiseksi harmaasävykuvaksi: **Image** → **Type** → **8-bit**.

7. Ajetaan plugin:

- Plugins → Compile and Run...
- Valitaan oma plugin.
- TAI painamalla: **Ctrl+R**.

8. Suljetaan ImageJ.

9. Avataan ImageJ. Nyt omien pluginien alivalikko ilmestyy `plugins`-valikon alle.

```
// Kahdeksanbittisestä harmaasävykuvasta negatiivin
// tekevä ImageJ-plugini.
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.*;

public class Invertteri_ implements PlugInFilter
{
    ImagePlus imp; // Kuvaikkuna

    public int setup(String arg, ImagePlus imp)
    {
        this.imp = imp;
        return DOES_8G; // Käsitellään 8-bittisiä harmaasävykuvia
    }

    // ip = kuvaolio, joka sisältää myös metodeja, kuten
    // getPixel ja putPixel.
    public void run(ImageProcessor ip)
    {
        int W=ip.getWidth();
        int H=ip.getHeight();

        // Käydään jokainen pikseli läpi kahdella sisäkkäisellä
        //for-silmukalla
        for(int y=0; y<H; y++)
        {
            for(int x=0; x<W; x++)
            {
                ip.putPixel(x,y, 255-ip.getPixel(x,y));
            }
        }
        imp.updateAndDraw();
    }
}
```