

## ■ Image Download

### ● 다운로드할 이미지 파일 확인

← → ↻ ⓘ 주의 요함 ggoreb.com/http/html1.jsp



```
...<!doctype html> == $0
<html>
  <head>...</head>
  <body>
    <div class="info">
      
      
      
      
    </div>
  </body>
</html>
```

## ■ Image Download

- 다운로드할 이미지 파일의 위치 지정

```
String imgUrl = "http://ggoreb.com/images/luffy.jpg";  
URL url = new URL(imgUrl);
```

- ImageIO 라이브러리를 이용하여 이미지를 읽은 후 메모리에 임시 저장

```
BufferedImage img = ImageIO.read(url);
```

- 파일 저장

```
ImageIO.write(img, "jpg", new File("luffy.jpg"));
```

## ■ Image Download

### ● 전체 소스코드

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import java.net.URL;

import javax.imageio.ImageIO;

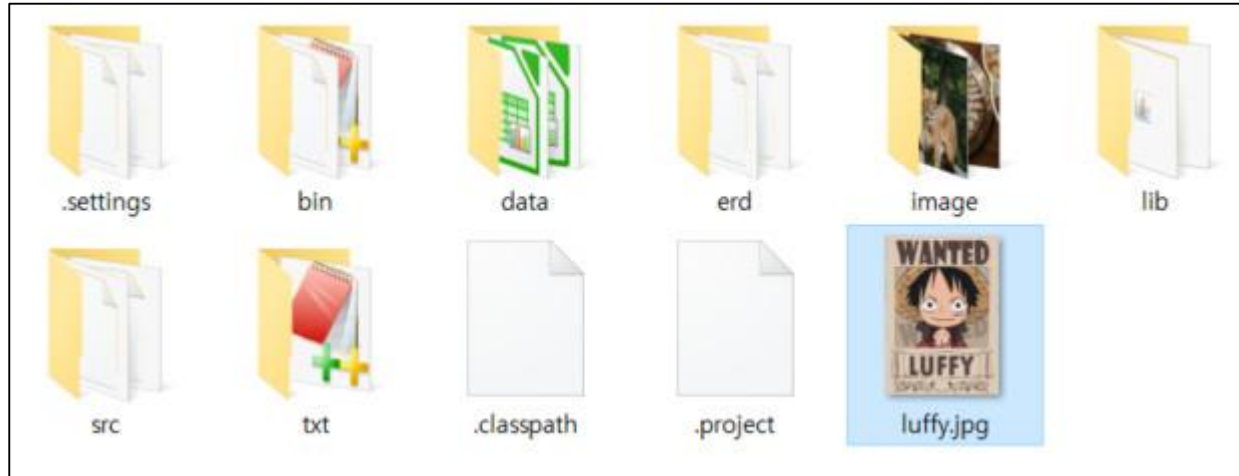
public class ImageDownload1 {
    public static void main(String[] args) throws IOException {
        String imgUrl = "http://ggoreb.com/images/luffy.jpg";
        URL url = new URL(imgUrl);

        BufferedImage img = ImageIO.read(url);

        ImageIO.write(img, "jpg", new File("luffy.jpg"));
    }
}
```

## ■ Image Download

### ● 실행 결과



## ■ Image Resize – 이미지 크기 확인

### ● 이미지 속성



## ■ Image Resize – 이미지 크기 확인

### ● 전체 소스코드

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class ImageCrop1 {
    public static void main(String[] args) throws IOException {
        BufferedImage img = ImageIO.read(new File("image/crop1.jpg"));
        int width = img.getWidth();
        int height = img.getHeight();
        System.out.println(width);
        System.out.println(height);
    }
}
```

width - 868, height - 1280

## ■ Image Resize – 이미지 크기 변경

- 이미지의 width / height 확인 후 크기 변경

```
BufferedImage img = ImageIO.read(new File("image/crop1.jpg"));  
int width = img.getWidth() / 2;  
int height = img.getHeight() / 2;
```

- 변경된 크기를 적용한 Image 객체 생성

```
Image resizeImg = img.getScaledInstance(width, height, Image.SCALE_DEFAULT);
```

- 새로운 이미지를 저장할 BufferedImage 객체 생성

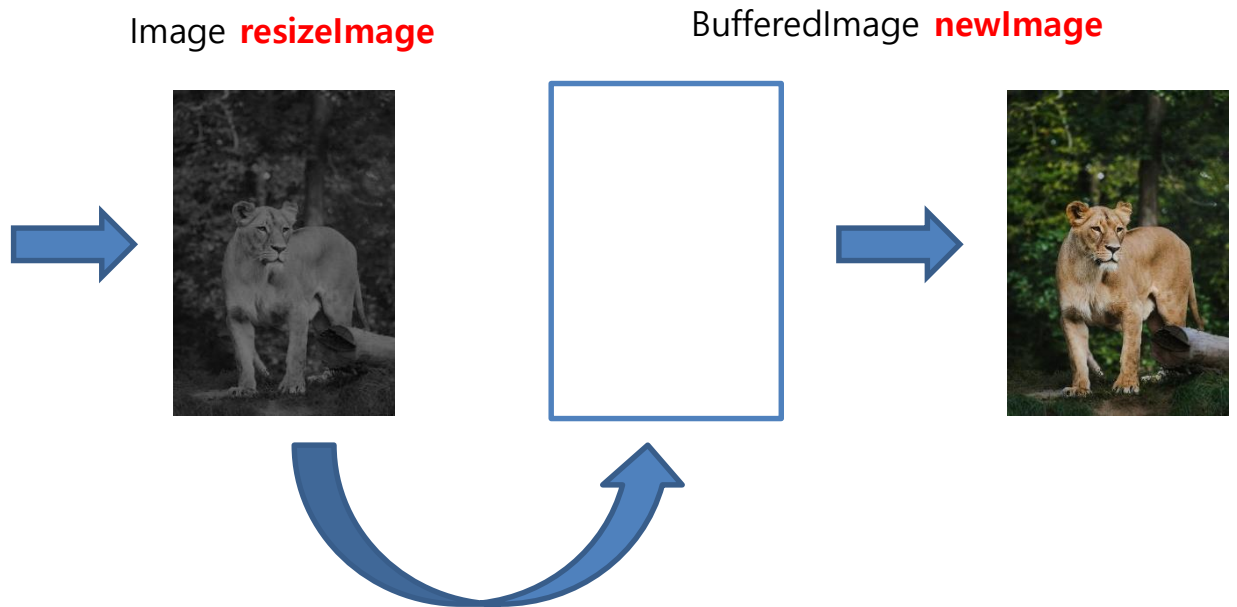
```
BufferedImage newImage =  
    new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);  
Graphics g = newImage.getGraphics();  
g.drawImage(resizeImg, 0, 0, null);
```

- 새로운 파일로 저장

```
ImageIO.write(newImage, "jpg", new File("image/crop1c.jpg"));
```

## ■ Image Resize – 이미지 크기 변경

BufferedImage **img**





## ■ Image Resize – 이미지 크기 변경

### ● 전체 소스코드 (1 / 2)

```
import java.awt.Graphics;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class ImageCrop2 {
    public static void main(String[] args) throws IOException {
        BufferedImage img = ImageIO.read(new File("image/crop1.jpg"));
        int width = img.getWidth() / 2;
        int height = img.getHeight() / 2;

        // 이미지 크기와 이미지 스케일의 알고리즘 지정
        Image resizeImg =
            img.getScaledInstance(width, height, Image.SCALE_DEFAULT);
    }
}
```

## ■ Image Resize – 이미지 크기 변경

### ● 전체 소스코드 (2 / 2)

```
// 새로운 이미지를 저장할 객체 생성
// ImageIO의 write 메소드는 Image 형태의 객체를 직접 사용할 수 없고
// RenderedImage를 처리하므로 자식 클래스인 BufferedImage를 생성하고
// Image의 데이터를 복사한 후 사용
BufferedImage newImage =
    new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
Graphics g = newImage.getGraphics();
g.drawImage(resizeImg, 0, 0, null);

ImageIO.write(newImage, "jpg", new File("image/crop1c.jpg"));
}
```

■ Image Crop – 이미지 잘라내기 (세로 크기 기준)

- 정사각형(1:1 비율)을 기준으로 벗어나는 영역은 제거



## ■ Image Crop – 이미지 잘라내기 (세로 크기 기준)

- 이미지의 width / height 확인 후 크기 변경

```
BufferedImage img = ImageIO.read(new File("image/crop2.jpg"));  
int width = img.getWidth();  
int height = img.getHeight();
```

- x / y 좌표와 width / height 를 지정하여 서브 이미지 생성

```
BufferedImage newImage =  
    img.getSubimage(width / 2 - height / 2, 0, height, height);
```

- 새로운 파일로 저장

```
ImageIO.write(newImage, "jpg", new File("image/crop2c.jpg"));
```

## ■ Image Crop – 이미지 잘라내기 (세로 크기 기준)

### ● 전체 소스코드

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class ImageCrop3 {
    public static void main(String[] args) throws IOException {
        BufferedImage img = ImageIO.read(new File("image/crop2.jpg"));
        int width = img.getWidth();
        int height = img.getHeight();

        BufferedImage newImage =
            img.getSubimage(width / 2 - height / 2, 0, height, height);
        ImageIO.write(newImage, "jpg", new File("image/crop2c.jpg"));
    }
}
```

■ Image Crop – 이미지 잘라내기 (가로 크기 기준)

- 정사각형(1:1 비율)을 기준으로 벗어나는 영역은 제거



## ■ Image Crop – 이미지 잘라내기 (가로 크기 기준)

- 이미지의 width / height 확인 후 크기 변경

```
BufferedImage img = ImageIO.read(new File("image/crop3.jpg"));  
int width = img.getWidth();  
int height = img.getHeight();
```

- x / y 좌표와 width / height 를 지정하여 서브 이미지 생성

```
BufferedImage newImage =  
    img.getSubimage(0, height / 2 - width / 2, width, width);
```

- 새로운 파일로 저장

```
ImageIO.write(newImage, "jpg", new File("image/crop3c.jpg"));
```

## ■ Image Crop – 이미지 잘라내기 (가로 크기 기준)

### ● 전체 소스코드

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class ImageCrop4 {
    public static void main(String[] args) throws IOException {
        BufferedImage img = ImageIO.read(new File("image/crop3.jpg"));
        int width = img.getWidth();
        int height = img.getHeight();

        BufferedImage newImage =
            img.getSubimage(0, height / 2 - width / 2, width, width);
        ImageIO.write(newImage, "jpg", new File("image/crop3c.jpg"));
    }
}
```



## ■ watermark



## ■ watermark

- 원본 이미지 및 워터마크로 사용할 로고 이미지 가져오기

```
BufferedImage sourceImage = ImageIO.read(new File("image/wm1.jpg"));  
BufferedImage watermarkImage = ImageIO.read(new File("image/Logo.jpg"));
```

- 워터마크를 추가하기 위해 원본 이미지의 graphics 객체 가져오기

```
Graphics g = sourceImage.getGraphics();
```

- 워터마크가 그려질 위치 지정 (원본 이미지의 중앙)

```
int x = (sourceImage.getWidth() - watermarkImage.getWidth()) / 2;  
int y = (sourceImage.getHeight() - watermarkImage.getHeight()) / 2;
```

- 원본 이미지에 워터마크 이미지 그리기

```
g.drawImage(watermarkImage, x, y, null);
```

- 새로운 파일로 저장

```
ImageIO.write(sourceImage, "jpg", new File("image/wm1c.jpg"));
```

## ■ watermark

### ● 전체 소스코드 (1 / 2)

```
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

public class WaterMark1 {
    public static void main(String[] args) throws IOException {
        // 원본 이미지 및 워터마크로 사용할 로고 이미지 가져오기
        BufferedImage sourceImage = ImageIO.read(new File("image/wm1.jpg"));
        BufferedImage watermarkImage = ImageIO.read(new File("image/Logo.jpg"));

        // 워터마크를 추가하기 위해 원본 이미지의 graphics 객체 가져오기
        Graphics g = sourceImage.getGraphics();

        // 워터마크가 그려질 위치 지정 (원본 이미지의 중앙)
        int x = (sourceImage.getWidth() - watermarkImage.getWidth()) / 2;
        int y = (sourceImage.getHeight() - watermarkImage.getHeight()) / 2;
```

## ■ watermark

### ● 전체 소스코드 (2 / 2)

```
// 원본 이미지에 워터마크 이미지 그리기
```

```
g.drawImage(watermarkImage, x, y, null);
```

```
// 새로운 파일로 저장
```

```
ImageIO.write(sourceImage, "jpg", new File("image/wm1c.jpg"));
```

```
}
```

```
}
```



## ■ watermark – 투명도 설정

- 워터마크를 추가하기 위해 원본 이미지의 graphics 객체 가져오기

```
Graphics g = sourceImage.getGraphics();
```



- 투명도를 설정하기 위해 Graphics2D 객체 사용

```
Graphics2D g = (Graphics2D) sourceImage.getGraphics();
```

```
// 투명도 50% (0.0 ~ 1.0)
```

```
AlphaComposite alphaChannel =
```

```
    AlphaComposite.getInstance(AlphaComposite.SRC_OVER, 0.5f);
```

```
g.setComposite(alphaChannel);
```





■ watermark

● wm2.jpg (우, 상)



● wm3.jpg (좌, 하)



● wm4.jpg (우, 하)



● wm5.jpg (우, 중)

