

■ Google 애플리케이션 비밀번호 설정

● 사이트 접속



■ Google 애플리케이션 비밀번호 설정

● 2단계 인증

비밀번호 및 로그인 방식

비밀번호로 계정을 보호하거나 2단계 인증으로 보안을 강화할 수 있습니다. 2단계 인증을 사용하면 로그인 시 입력해야 하는 일회용 코드가 휴대전화로 전송됩니다. 따라서 다른 사람이 비밀번호를 알게 되더라도 내 계정에 액세스할 수 없습니다.

참고: 이 설정을 변경하려면 비밀번호를 확인해야 합니다.

비밀번호

최종 변경일: 3월 3일 오후 12:42



2단계 인증

사용 중지



2단계 인증

비밀번호 및 휴대전화를 사용하여 악의적인 사용자가 계정에 액세스하지 못하도록 차단합니다.

설정 시작 »

자세히 알아보기

■ Google 애플리케이션 비밀번호 설정

● 2단계 인증

휴대전화 설정

1

2

3

4

코드를 전송할 휴대전화 번호는 무엇입니까?

신뢰할 수 없는 컴퓨터 또는 기기에서 로그인할 때마다 Google에서 휴대전화로 숫자 코드를 전송합니다.

전화번호 예: 010-0000-0000

010-8478-8182

코드 전송 방법 선택

☒ 문자 메시지(SMS)

☐ 음성 통화

« 뒤로 **코드 전송**

- 이 전화번호를 2016. 3. 2에 계정 보안 용도로 입력하셨습니다.
- Google에서는 전화번호를 계정 보안 용도로만 사용합니다.
- 문자 메시지 요금이나 데이터 요금이 부과될 수 있습니다.

전화 확인

1

2

3

4

010-8478-8182(으)로 코드가 포함된 문자 메시지를 전송했습니다.

인증 코드 입력

인증 코드는 6자리 숫자입니다.

« 뒤로 **확인** 코드를 받지 못하십니까?

이 컴퓨터를 신뢰하시겠습니까?

1

2

3

4

신뢰할 수 있는 컴퓨터는 로그인할 때마다 인증 코드를 입력할 필요가 없습니다.

휴대전화를 분실한 경우 신뢰할 수 있는 컴퓨터에서 인증 코드 없이 계정에 액세스할 수 있습니다. 컴퓨터에 액세스 권한이 있는 사용자를 모두 신뢰할 수 있는 경우에만 이 컴퓨터를 신뢰할 수 있는 컴퓨터로 지정하는 것이 좋습니다.

☒ 이 컴퓨터를 신뢰함
Google 계정 설정에서 신뢰할 수 있는 컴퓨터를 언제든지 변경할 수 있습니다.

« 뒤로 **다음**

확인

1

2

3

4

2단계 인증 사용

신뢰할 수 없는 컴퓨터 또는 기기에서 계정을 사용하여 로그인할 때만 코드를 입력하라는 메시지가 표시됩니다.

휴대전화를 분실한 경우 계정 설정에서 언제든지 변경할 수 있습니다.

« 뒤로 **확인**

■ Google 애플리케이션 비밀번호 설정

● 앱 비밀번호 등록

2단계 인증

인증 코드	앱 비밀번호	등록된 컴퓨터	보안 키
애플리케이션 구성			
<p>앱 비밀번호를 사용하면 2단계 인증 사용자가 iPhone 또는 Mac의 Mail이나 Outlook과 같은 앱을 통해 Google 계정에 액세스할 수 있습니다.</p>			
<div><div></div><div>iPhone 또는 iPad의 Mail</div></div> <div></div> <div>Microsoft Outlook</div> <div></div> <div>Mac의 Mail</div>			
<p>일부 앱은 사용자 이름, 비밀번호, 인증 코드 대신 사용자 이름과 비밀번호만으로 구성해야 합니다.</p> <p>이런 경우 코드 대신 앱 비밀번호를 사용하여 계정에 액세스할 수 있습니다.</p>			
<p>앱에서 비밀번호와 관련된 오류가 발생하는 경우, 앱 비밀번호로 설정해야 합니다. 앱 비밀번호는 Google에서 만드므로 기억해야 할 필요가 없습니다. 또는 대신 Google 앱을 사용해 보세요.</p>			
애플리케이션 비밀번호 관리			

■ Google 애플리케이션 비밀번호 설정

● 앱 비밀번호 등록

앱 비밀번호를 사용하면 2단계 인증 사용자가 iPhone 또는 Mac의 Mail이나 Outlook과 같은 앱을 통해 Google 계정에 액세스할 수 있습니다. 앱 비밀번호는 Google에서 만드므로 일일이 기억할 필요가 없습니다.
[자세히 알아보기](#)

앱 비밀번호가 없습니다.

내 Windows 컴퓨터 ▼ 의 메일 ▼ 생성

생성된 앱 비밀번호

Windows 컴퓨터용 앱 비밀번호

udej wjat ayrw ofyz

Add your Google account

Enter the information below to connect to your Google account.

Email address

securesally@gmail.com

Password

☐ Include your Google contacts and calendars

사용 방법

1. '메일' 앱을 엽니다.
2. '설정' 메뉴를 엽니다.
3. '계정'을 선택한 뒤 내 Google 계정을 선택합니다.
4. 비밀번호를 위에 표시된 16자리 비밀번호로 교체합니다.

일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

[자세히 알아보기](#)

확인

■ Google 애플리케이션 비밀번호 설정

● 앱 비밀번호 등록

앱 비밀번호를 사용하면 2단계 인증 사용자가 iPhone 또는 Mac의 Mail이나 Outlook과 같은 앱을 통해 Google 계정에 액세스할 수 있습니다. 앱 비밀번호는 Google에서 만드므로 일일이 기억할 필요가 없습니다.
[자세히 알아보기](#)

앱 비밀번호가 없습니다.

내 Windows 컴퓨터 ▼ 의 메일 ▼ 생성

생성된 앱 비밀번호

Windows 컴퓨터용 앱 비밀번호

udej wjat ayrw ofyz

Add your Google account

Enter the information below to connect to your Google account.

Email address

securesally@gmail.com

Password

☐ Include your Google contacts and calendars

사용 방법

1. '메일' 앱을 엽니다.
2. '설정' 메뉴를 엽니다.
3. '계정'을 선택한 뒤 내 Google 계정을 선택합니다.
4. 비밀번호를 위에 표시된 16자리 비밀번호로 교체합니다.

일반적인 비밀번호와 마찬가지로 이 앱 비밀번호는 Google 계정에 대한 완전한 액세스 권한을 부여합니다. 비밀번호를 기억하지 않아도 되므로 적어 놓거나 다른 사용자와 공유하지 마세요.

[자세히 알아보기](#)

확인

■ 메일 발송 코드 작성

● 필요 라이브러리

- mail.jar
- activation.jar

● email.SMTPAuthenticator.java

```
package email;
import javax.mail.Authenticator;
import javax.mail.PasswordAuthentication;
public class SMTPAuthenticator extends Authenticator {
    @Override
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication("본인계정@gmail.com", "구글앱비밀번호");
    }
}
```

■ 메일 발송 코드 작성

● email.Mailer.java (1 / 3)

```
import java.util.Properties;
import javax.mail.Address;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class Mailer {
    public void sendMail(String to, String subject, String content, SMTPAuthenticator smtp) {
        // SMTP 서버 정보 저장
        Properties p = new Properties();
        p.put("mail.smtp.host", "smtp.gmail.com");
        p.put("mail.smtp.port", "465");
        p.put("mail.smtp.starttls.enable", "true");
```


■ 메일 발송 코드 작성

● email.Mailer.java (2 / 3)

```
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.debug", "true");
p.put("mail.smtp.socketFactory.port", "465");
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
p.put("mail.smtp.socketFactory.fallback", "false");

try {
    Session ses = Session.getInstance(p, smtp);
    ses.setDebug(true);

    MimeMessage msg = new MimeMessage(ses); // 메일의 내용을 담을 객체
    msg.setSubject(subject); // 제목

    Address fromAddr = new InternetAddress("본인계정@gmail.com");
    msg.setFrom(fromAddr); // 보내는 사람
```

■ 메일 발송 코드 작성

● email.Mailer.java (3 / 3)

```
        Address toAddr = new InternetAddress(to);
        msg.addRecipient(Message.RecipientType.TO, toAddr); // 받는 사람
        msg.setContent(content, "text/html;charset=UTF-8"); // 내용과 인코딩
        Transport.send(msg); // 전송
    } catch (Exception e){
        e.printStackTrace();
    }
}
```

■ 메일 발송 코드 작성

● email.EmailMain.java

```
public class EmailMain {  
    public static void main(String[] args) {  
        String to = "받는사람@이메일.주소";  
        String subject = "반갑습니다.";  
        String content = "가임을 축하드립니다.";  
  
        SMTPAuthenticator smtp = new SMTPAuthenticator();  
        Mailer mailer = new Mailer();  
        mailer.sendMail(to, subject, content, smtp);  
    }  
}
```