

```
app.MapGet("/trust/complete", async (TrustedDb db) =>
    await db.TrustCode.Where(t => t.IsComplete).ToListAsync());

app.MapGet("/trust/incomplete", async (TrustedDb db) =>
    await db.TrustCode.Where(t => !t.IsComplete).ToListAsync());

app.MapGet("trust/{id}", async (int id, TrustedDb db) =>
    await db.TrustCode.FindAsync(id) is Trusted trust
        ? Results.Ok(trust)
        : Results.NotFound());

app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);

    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();

    return Results.Created($"/trust/{trust.Id}", trust);
});
```

# TRUST CODE

Microsoft의 비즈니스 행동 표준

```
app.MapDelete("/trust/delete-all", async (TrustedDb db) =>
    Results.Ok(await db.Database.ExecuteSqlRawAsync("DELETE FROM TrustCode")));

app.Run();

Task EnsureDB(string connectionString)
{
    var options = new
        DbContextOptionsBuilder<TrustedDb>().UseSqlite(connectionString).Options;
    using var db = new TrustedDb(options);
    return db.Database.MigrateAsync();
}

enum Courses
{
    Accessibility,
    CompetingFairly,
    DigitalSafety,
    EthicalSelling,
    GlobalReadiness,
    InsiderTrading,
    ModelCoachCare,
    ResponsibleAI,
    Privacy,
    SecurityZeroTrust
}
```



팀 여러분,

우리 각자는 말과 행동을 통해 문화를 형성합니다. 우리는 학습을 포용하고 신뢰를 다지는 다양하고 포용적인 문화, 즉 모든 직원이 최선을 다할 수 있는 문화를 구축하기 위해 노력합니다.

업무에서 올바른 결정과 윤리적 선택을 하는 것은 서로에 대한 신뢰와 고객 및 파트너와의 신뢰를 구축합니다. 단기적인 이익에 대한 대가로 개인의 무결성이나 Microsoft의 명성과 신뢰를 손상시켜서는 안 됩니다.

무결성, 정직성 및 규정 준수에 의거해 의사 결정을 내릴 때 윤리적인 선택을 할 가능성이 더 커집니다. 우리는 항상 투명한 동기를 가지고, 실수로부터 배우고, 어려운 상황에 직면했을 때 도움을 요청해야 합니다. 저는 리더와 관리자가 직원들이 자유롭게 질문하고 무언가가 옳지 않은 것처럼 보일 때 우려를 제기하는 문화를 조장할 것으로 기대합니다.

Microsoft의 비즈니스 행동 표준("신뢰 코드")은 신뢰를 구축하는 데 있어 우리 각자가 수행하는 역할과 의사 결정 시 취해야 할 접근 방식을 강조합니다. 우리가 일상 업무에 이러한 원칙을 적용할 때, 우리는 신뢰를 구축하고 고객과 파트너가 더 많은 것을 성취할 수 있도록 역량을 강화하는 올바른 결정을 내릴 수 있는 능력에 대한 확신을 가지고 앞으로 나아갈 수 있습니다.

감사합니다.

**Satya Nadella**

회장 겸 CEO



# 목차

## Microsoft 문화 및 가치

우리의 사명

우려 사항 보고

업무 성취도 개선

관리자의 역할

신뢰 코드 사용

말하기

## 고객과의 신뢰

개인 정보 존중

안전한 제품과 서비스 구축

부적절한 지급을 하지 않기

공정하게 경쟁

## 정부 기관 및 지역사회와의 신뢰

전 세계의 법률 준수

인권 존중과 증진

접근 가능한 제품 및  
서비스 설계

책임 있는 AI 개발 및 배포

## 서로에 대한 신뢰

다양성 및 포용성 육성

안전하고 생산적인  
업무 환경에 기여

이해 충돌 방지

## 투자자 및 대중과의 신뢰

내부 정보를 거래하지 않기

Microsoft의 리소스 보호

정확한 기록 및 계약 유지

기밀 정보 및  
지적 재산 보호

대중에게 정확하게 의사 소통

## 대표자와의 신뢰

책임감 있게 선물, 환대 및  
여행 제공

신뢰할 수 있는 대표자 활용

무결성을 갖춘 공급업체 선택

## 신뢰 코드 준수

신뢰 코드 준수

# TRUST

Microsoft 문화와 가치 신뢰

우리의 사명

5

업무 성취도 개선

6

신뢰 코드를 사용하는 방법

7

우려 사항을 보고하는 방법

9

무언가가 옳지 않을 때 말하기

10

관리자의 역할

12

```
app.MapGet("/trust/{id}", async. (  
    await db.TrustCode.FindAsync(id)  
    ? Results.Ok(trust)  
    : Results.NotFound());  
  
app.MapPost("/trust", async. (Trusted t  
{  
    if (!MinimalValidation.TryValida  
        return Results.Validation  
db.TrustCode.Add(trust);  
await ChangesAsync();  
return Results.Created($"trust/  
});  
app.MapPut("/{id}", async (int i
```







# 우리의 사명

지구 상의 모든 사람과 모든 조직의  
역량을 강화하여  
더 많은 성과를 달성하세요.

## 우리의 사명 달성

### Microsoft 가치

존중

무결성

책임성

### Microsoft 문화

성장 중심 사고 방식

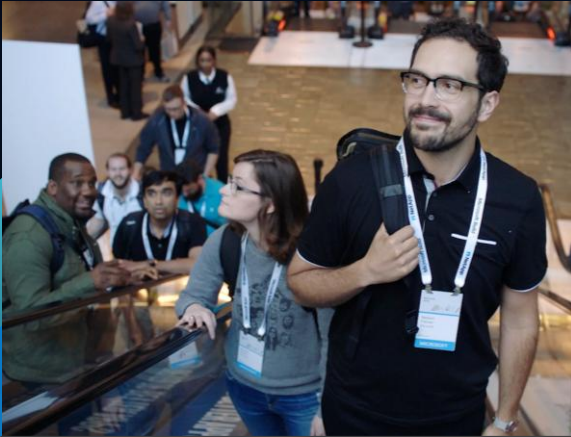
고객 중심

One Microsoft

다양성 및 포용성

차이를 만들기

# 업무 성취도 개선



어떻게 신뢰를 얻고 구축할 수 있을까요?

신뢰 구축은 우리의 문화와 가치를 적용하여 Microsoft 안팎에서 지속적인 관계를 구축하는 것으로 시작됩니다.

**Microsoft** 가치는 우리가 매일 신뢰를 얻기 위해 노력할 때 성실하게 비즈니스를 수행하도록 안내하는 지속적인 원칙입니다.

**Microsoft** 문화는 우리의 운영 틀, 즉 우리가 누구이며 어떻게 행동하는지를 말합니다.



Microsoft의 비즈니스 행동  
표준

Microsoft의 비즈니스 행동 표준("신뢰 코드")은 우리가 함께 더 많은 것을 성취할 수 있도록 고객, 정부, 투자자, 파트너, 대표자 및 서로와의 신뢰를 구축하고 유지하기 위해 Microsoft의 문화와 가치를 활용하는 방법을 보여줍니다.

# 신뢰 코드를 사용하여 올바른 결정을 내리는 방법



올바른 결정과 윤리적 선택을  
하는 것은 우리 각자와 우리가  
상호 작용하는 사람들 사이의  
신뢰를 구축합니다.

그러나 당신이 직면하는 모든 상황이  
간단하지는 않습니다. 어렵거나  
불분명한 상황에 직면할 때 어떻게  
최선의 선택을 할 수 있을까요? 윤리적  
딜레마를 어떻게 헤쳐 나갈 수 있을까요?

Microsoft의 신뢰 코드는 모든 상황에서  
무엇을 해야 하는지 정확히 알려주지는  
않지만 올바른 결정을 내리고 정답이  
항상 명확하지 않을 수도 있는 복잡한  
상황을 탐색하는 데 도움이 되는 가이드  
역할을 합니다.

app.MapGet  
awa

app.Map  
{

d, Trusted  
usted, tr

ustedD



# 어려운 결정이나 상황에 직면했을 때 다음 단계를 따르세요.

```
app.MapGet("/trust/{id}", async (int id, Tru
    await db.TrustCode
    ? Results.Ok
    : Results.No

app.MapPost("/trust", async (Trust trust, Tru
{
    if (!MinimalValidation.TryValidate(trust
        return Results.ValidationProblem(
db.TrustCode.Add(trust);

//{trust.Id
id, Truste
e(inputTru
lem(violat
.FindAsync
ts.NotFou
```

Microsoft  
runs on  
trust.

1



## 일시 중지

상황이 당신을 불안하게 만드나요? 본능이 당신에게 무언가가 옳지 않다고 말하고 있나요?

잠시 멈추고 상황에 접근하는 방법을 고려하세요.

2



## 생각하기

본인의 접근 방식이 Microsoft 신뢰 코드의 문화 및 가치와 일치합니까? 이러한 방식이 신뢰를 구축하거나 유지합니까? 단기적인 이익을 위해 장기적인 명성과 신뢰를 희생해서는 안 됩니다.

3



## 질문하기

질문하고 도움을 받으세요. CELA, 재무 또는 HR 팀과 상담하세요. 자신의 관리자 또는 다른 관리자의 도움을 받으세요.

이러한 리소스는 올바른 결정을 내리는 데 도움이 됩니다.

도움말은 [buscond@microsoft.com](mailto:buscond@microsoft.com)으로 이메일을 보내주세요.

문제를 해결할 수 있도록 무언가가 옳지 않을 때 소리 내어 말해 주세요.

무결성 우려 사항 보고 >



```
12 { if (ModelState.IsValid) {  
    return Results.  
    db.TrustCode.Add(trustCode);  
    await db.SaveChangesAsync();  
    return Results.CreateActionResultForPage();  
});  
app.MapPut("/trust/{id}", async (int id, TrustCode trustCode) =>  
{
```

# 우려 사항을 보고하는 방법

우려를 제기하는 과정은 가능한 한 쉬워야 합니다. 그렇기 때문에 무언가가 잘못되었을 때 보고할 수 있는 다양한 방법이 있습니다.

가장 편안하게 사용할 수 있는 보고 옵션을 선택하세요.  
어떤 옵션을 선택하든 기밀성이 보호됩니다.



해당 국가의 법률이 허용하는 경우, 귀하의 보고는 당사의 외부 핫라인을 통해 익명으로 작성되거나 규정 준수 및 윤리 팀에 귀하의 신고를 익명으로 처리하도록 요청할 수 있습니다.



**웹사이트** [www.microsoftintegrity.com](http://www.microsoftintegrity.com)



**팩스** 1-425-708-7177



**이메일** [buscond@microsoft.com](mailto:buscond@microsoft.com)



**우편**

Business & Regulatory Investigations  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052 United States



**전화** 1-877-320-6738  
국제 수신자 부담 +1-720-904-6844

## 추가 리소스

또한 관리자, 모든 Microsoft 관리자, HR, 재무 또는 CELA에 언제든지 문제를 제기할 수 있습니다.

모바일에서? 이 QR 코드를  
스캔하여 이메일을 보내  
주세요.



SCAN ME

# 말하기

무언가가 옳지 않을 때

Microsoft  
runs on  
trust.

신뢰의 문화를 구축하는 것의 일부는  
무언가가 옳지 않을 때 목소리를  
높이는 법을 배워 문제를 해결할 수  
있도록 하는 것입니다.

## 무결성 우려사항 제기



무결성 문제를 제기하기 위해 모든  
세부 사항을 알고 있거나 무언가가  
잘못되었는지 확인할 필요는  
없습니다.

Microsoft는 귀하의 보고서를  
진지하고 공정하게 취급하며  
수정해야 할 사항이 있는 경우  
우리가 배운 것을 기반으로 신속하게  
행동하기 위해 노력합니다.



## 보복 금지

무언가가 옳지 않을 때 목소리를 높이는 데는 용기가 필요합니다. 마음이 불편하거나 불안할 수 있음을 이해합니다. 그래서 우리는 보복을 금지합니다. 다음에 대한 보복을 금지합니다.



거부로 인해 Microsoft에 비즈니스 손실이 발생하더라도 신뢰 코드, Microsoft의 정책 또는 법률을 위반하는 행위를 거부합니다.



잠재적인 위법 행위에 대해 선의의 우려를 제기합니다.



조사에 협조합니다.

Microsoft의 정책에 따라 해당 행위에 관여하는 직원에게 보복하는 사람에게 최대 해고를 포함한 징계 조치가 내려집니다.





# 관리자의 역할

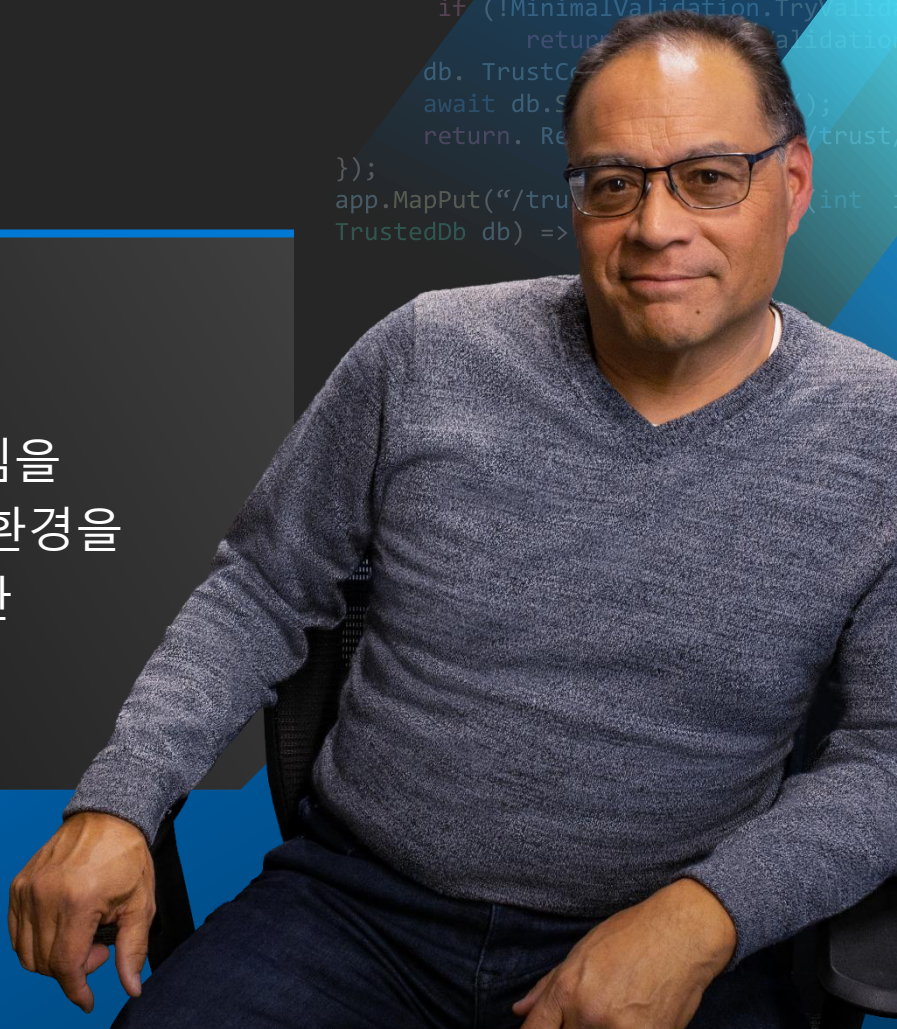


리더로서 당신은 팀을  
위한 문화와 업무 환경을  
구축하는 데 특별한  
책임이 있습니다.

```
app.MapGet("/trust/{id}", async () =>
{
    await db.TrustCode.FindAsync(id)
        .ConfigureAwait();
    if (Results.NotFound())
    {
        return Results.NotFound();
    }
});

app.MapPost("/trust", async (TrustedCode db) =>
{
    if (!MinimalValidation.TryValidate(
        db.TrustCode, out var validationErrors))
    {
        return Results.ValidationErrors(validationErrors);
    }
    await db.TrustCode.AddAsync(TrustedCode);
    return Results.Created("/trust/{id}", TrustedCode.Id);
});

app.MapPut("/trust/{id}", async (TrustedCode db) =>
```



의사 결정을 내리고 우려 사항, 다양한 의견 및 나쁜 소식을 처리하는  
방식은 팀, 고객 및 이해 관계자와의 신뢰의 토대를 마련할 것입니다.  
여러분의 성공과 팀의 성공은 함께 구축하는 신뢰에 달려 있습니다.



팀에서 신뢰와 무결성의 문화를 구축하기 위해  
다음과 같은 간단한 단계를 수행해야 합니다.

## 의사 소통

주기적으로 팀원들에게 윤리와 성실성의 중요성을 강조하고 업무가 윤리적으로 수행될 것으로 기대한다는 것을 분명히 밝힙니다.



## 모델링

윤리적 의사 결정과 올바른 판단을 모델링하여 모범을 보입니다.



## 질문

결과가 의미 있으려면 올바른 방법으로 결과를 얻어야 한다는 사실을 팀에 알립니다. 그런 다음 질문을 하고 결과가 올바른 방법으로 달성되었음을 확인하여 만족을 느낍니다.



## 듣기

관리자로서 팀을 코치하고 관심을 드러내는 것이 중요합니다. 팀원에게 말하기 어려운 사항이 있더라도 귀를 기울일 것임을 알고 있도록 합니다.



# 관리자가 우려 사항을 처리하는 방법

누군가가 당신에게 우려 사항이 있다고 말한다면, 당신은 듣고 행동할 특별한 책임이 있습니다. 우려 사항을 적절하게 처리하는 것은 신뢰를 유지하고 Microsoft를 보호하는 데 매우 중요합니다.

누군가 우려를 제기하는 경우 다음 단계를 따르세요.



주의를 산만하게 만드는 요소를 제거하고 신중하게 **경청**합니다. **우려 사항**을 공유한 직원에게 감사를 표합니다. 그들이 어렵고 Microsoft에 매우 중요한 일을 방금 수행했음을 기억하세요.



내용에 동의하지 않더라도 **정중하게** 대응하고 모든 우려 사항을 **진지하게** 받아들입니다. 문제를 해결하기 위해 최선을 다하고 있음을 보여주세요.



상대방의 **기밀성**을 보호하기 위한 조치를 취합니다. 팀의 다른 사람들과 대화에 대해 논의하지 않도록 합니다.

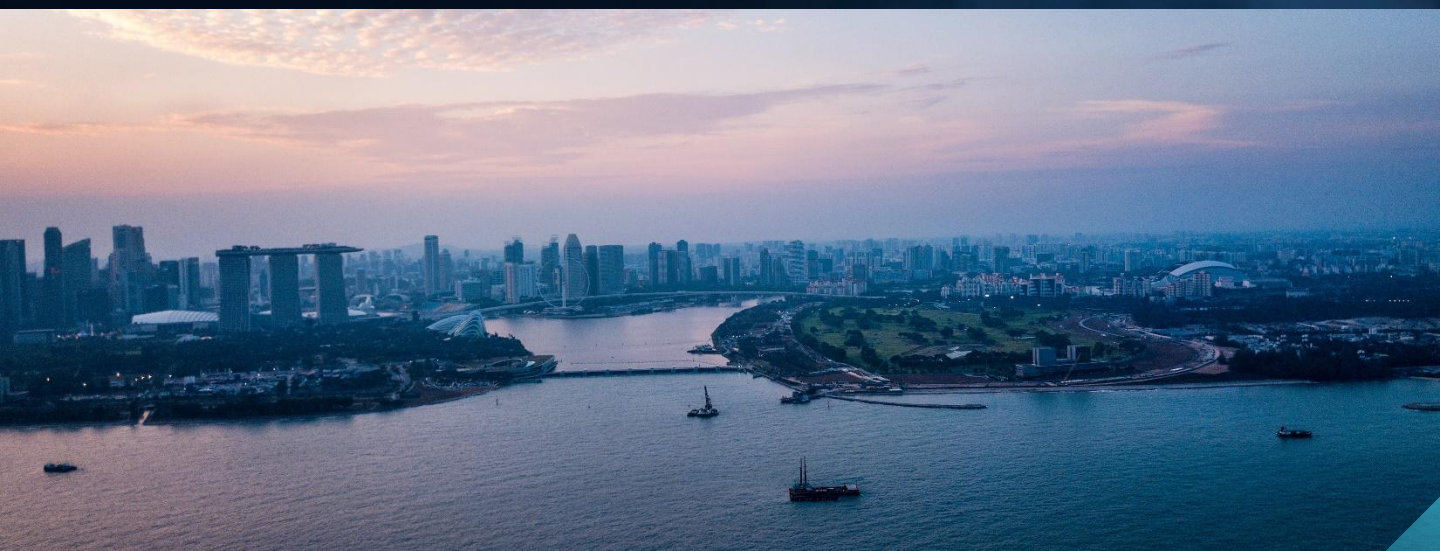


**괴롭힘 방지** 및 **차별 금지** 정책을 위반할 수 있는 우려 사항을 알게 되거나 행위를 목격한 경우 적시에 [AskHR](#)에 문의해야 합니다.

관리자는 스스로 성능 문제를 해결할 수 있는 역량이 있다고 느껴야 하지만 **비즈니스 윤리** 또는 **위법 행위에 대한 무결성 문제**를 Microsoft 비즈니스 및 규정 조사 팀으로 **에스컬레이션**해야 합니다.

질문이 있거나 문제를 직접 해결할 수 있는지 또는 해결해야 하는지 확실하지 않은 경우 언제든지 비즈니스 행동 및 규정 준수 팀 별칭(buscond@microsoft.com)으로 이메일을 보내 조언을 구할 수 있습니다.

비즈니스 행동 및 규정 준수 팀에 이메일 보내기 >



문제를 해결할 수 있도록 무언가가 옳지 않을 때  
소리 내어 말해 주세요.

무결성 우려 사항 보고



위법 행위에 대한 우려 사항 보고  
정책



# TRUST

## 고객과의 신뢰

개인 정보 존중

18

부적절한 지급을 하지 않기

19

공정하게 경쟁

21

안전한 제품과 서비스 구축

22



```
app.MapGet("/trust/{id}", async (int id, Tru
    await db.TrustCode.FindAsync(id) is Truste
        ? Results.Ok(trust)
        : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust, Tru
{
    if (!MinimalValidation.TryValidate(trust)
        return Results.ValidationProblem(
db. TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return. Results.Created($" /trust/{trust.Id
});
app.MapPut("/trust/{id}", async (int id, Truste
{
    if (!MinimalValidation.TryValidate(inputTru
        return Results.ValidationProblem(violat
        var trust = await db.TrustCode.FindAsync
        if (trust == null) return Results.NotFound();
```

```
app.MapGet("/trust/{id}", async (int id, Trusted trust, Tru
    await db.TrustCode.FindAsync(id) is Trusted trust
        ? Results.Ok(trust)
        : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
}
```

```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
{
    await db.TrustCode.FindAsync(id) is Trusted trust
        ? Results.Ok(trust)
        : Results.NotFound();
}
```

```
app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
});
app.MapPut("/trust/{id}", async (int id, Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
}
```

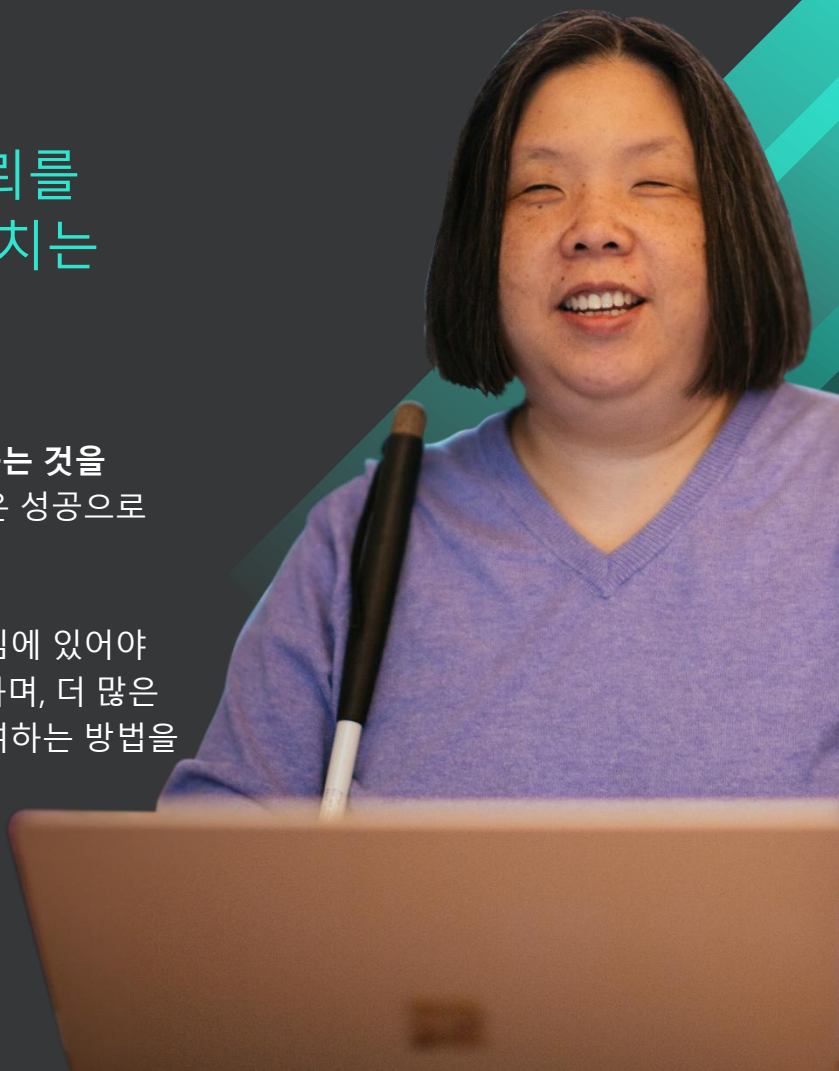


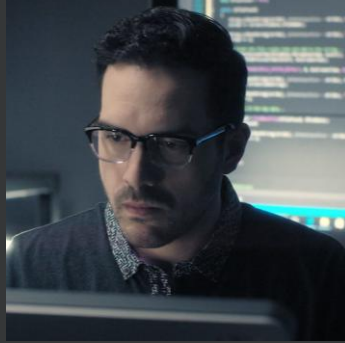


## 의사 결정을 내릴 때 그것이 **고객과의 신뢰를** **구축하니까, 혹은 해치는** **것입니까?**

사람들과 조직이 원하고 필요로 하는 것을  
가지고 그들과 만날 수 있는 능력은 성공으로  
향하는 길입니다.

고객은 우리가 하는 모든 것의 중심에 있어야  
합니다. **고객을 참여시키고**, 지지하며, 더 많은  
것을 성취할 수 있도록 역량을 부여하는 방법을  
찾아보세요.



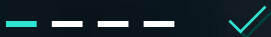


# 개인 정보 존중

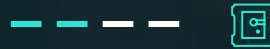
Microsoft는 개인 데이터를 포함한 고객 데이터를 처리하는 방법에 대해 투명하게 공개함으로써 개인 정보를 존중합니다.

고객이 자신의 개인 정보를 보호하고 그들이 허용하는 방식으로 데이터를 사용할 수 있도록 우리를 신뢰하기를 바랍니다.

## <우리가 개인 정보를 존중하는 방법>



현지 개인 정보 보호 및 데이터 보호법을 준수함으로써.



직원의 개인 데이터를 포함한 개인 데이터를 수집하거나 처리할 때 명확하고 정확한 개인 정보 보호 고지를 제공함으로써.



고객이 제공에 동의한 고객 데이터를 사용하여 서비스를 제공하여 개인 정보 보호 선택을 존중함으로써.



안전한 제품과 서비스를 구축하여 고객의 데이터를 보호함으로써.

아래 링크를 방문하여 자세히 알아보기 ↓

[개인 정보 취급 방침\(내부 전용\) >](#)

[개인 정보 보호 원칙 >](#)





```
app.MapGet("/trust/incomplete", async (TrustedDb db) =>
await db.TrustCode.Where(t => !t.IsComplete).ToListAsync());
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
await db.TrustCode.FindAsync(id) is Trusted trust
? Results.Ok(trust)
: Results.NotFound());
```

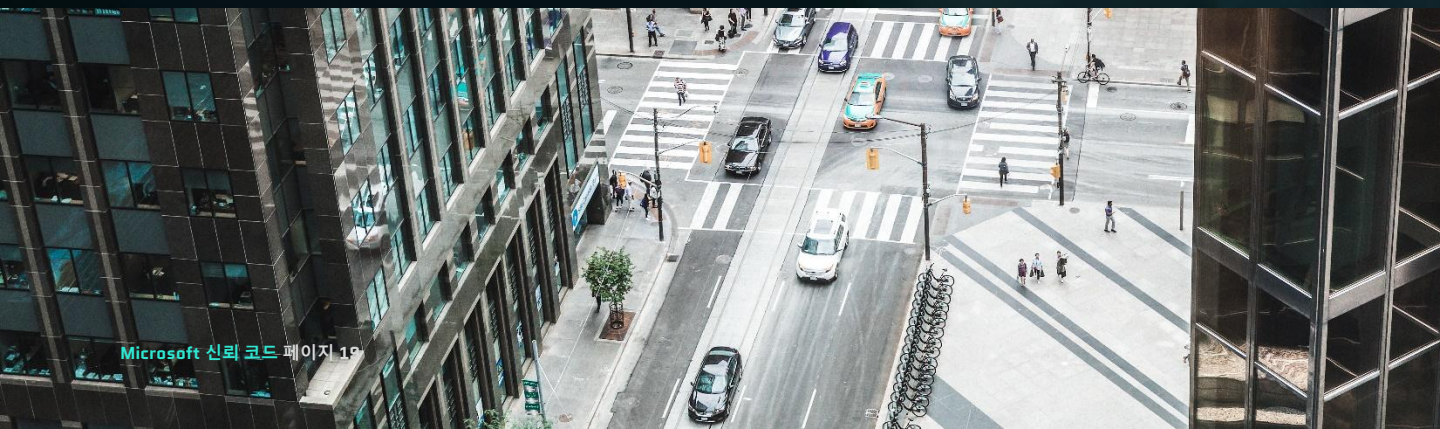
# 부적절한 지급을 하지 않기

우리는 고객과의 진실하고 투명한 상호 작용에 전념하고 있으며, 부적절한 지급을 통해 고객의 결정에 영향을 미치려는 시도를 금지합니다.

부패는 여러 가지 방법으로 고객에게 해를 끼칠 수 있으며, 때로는 고객이 지불해야 하는 것보다 더 많은 비용을 지불하게 하는 것을 포함합니다.

매일 고객의 신뢰를 얻고 유지하는 것은 부적절하게 비즈니스를 수행함으로써 얻을 수 있는 이익보다 중요합니다. 세계의 일부 지역에서는 사업을 따내기 위해 뇌물을 지불하는 것이 관행일 수 있습니다. Microsoft는 이것을 금지합니다.

뇌물, 사례 또는 기타 부적절한 혜택이 있을 경우 Microsoft는 비즈니스를 보호하기보다는 오히려 비즈니스를 잃을 것입니다.



## <신뢰를 구축하는 방법>

누구에게든 뇌물이나 사례를 제공하거나 지불하기를 거부함으로써.

허가 또는 승인을 확보하기 위한 지급과 일상적인 정부 프로세스(종종 "지불 촉진"으로 알려짐)를 가속화하기 위한 소액 지불을 포함하여 모든 종류의 부패한 지불을 금지함으로써.

투명한 거래가 뇌물이나 사례의 위험을 줄이기 때문에 어떤 거래에서든 우리는 숨겨진 조건이나 약정을 피하고 가능한 한 복잡성을 줄일 것입니다.

무결성을 갖춘 것으로 평판이 나 있는 파트너를 이용하고 대표자가 비윤리적이거나 뇌물을 지불할 수 있다는 징후를 보고함으로써.

정부 공무원이나 고객에게 제공하는 선물, 접대 또는 여행이 합리적이고 적절하며 필요한 경우 사전 승인되었는지 확인함으로써.

자신의 이득에 따라 지원자를 고용하고 고객 또는 공무원에게 이익이 되는 고용 결정을 내리지 않음으로써.

자선 기부를 한다면, 우리는 호의를 교환하는 것의 일부가 아닌 합법적인 자선 단체를 지원하기 위해 그렇게 할 것입니다.

부패 방지 정책



국가 공무원에게 제공되는 선물, 환대 및 여행 정책(내부 전용)



부패 및 뇌물 방지를 위한 노력







# 공정하게 경쟁

우리가 경쟁하는 방식은 우리가 성취하는 결과만큼 중요합니다.

고객에게 공정한 가격으로 다양한 제품과 서비스에 대한 액세스를 제공하는 건전한 경쟁과 공정한 비즈니스 관행을 통해 고객을 최우선으로 생각합니다. 공정하게 경쟁하면 무결성을 그대로 유지하면서 비즈니스 목표를 달성할 수 있습니다.

## <공정하게 경쟁하는 방법>

- > 경쟁을 제한하는 경쟁업체와의 공식 또는 비공식 계약을 피함으로써.
- > 경쟁 입찰 프로세스를 존중하고 결과를 조작하거나 수정하거나 다른 사람이 그렇게 하도록 돕지 않음으로써.
- > 독립적인 채널 파트너가 고객에게 부과하는 가격을 지시하지 않음으로써.
- > 적절한 리소스만을 사용하여 경쟁력 있는 정보를 수집하고, 진실하게 행동하며, 우리가 누구인지 잘못 드러내지 않음으로써.

아래 링크를 방문하여 자세히 알아보기

공정한 경쟁 정책  
(내부 전용)



경쟁력 있는 정보 정책  
(내부 전용)



# 안전한 제품과 서비스 구축

Microsoft는 가능한 가장 높은 보안  
표준을 충족하는 솔루션을 제공하는  
기술을 설계, 구축, 테스트 및  
운영하기 위해 최선을 다하고  
있습니다.

Microsoft는 고객 데이터의 모든  
측면이 기밀로 유지되고, 데이터  
무결성과 보안이 유지되며, 비즈니스  
기능이 효과적으로 작동하고 공격에  
저항할 수 있도록 하기 위해  
노력합니다. 우리 모두는 Microsoft와  
당사 고객의 보안에 직접적으로  
기여할 책임을 가집니다.

```
if (!MinimalValidation.TryValidate(trust, out var violations))  
    return Results.ValidationProblem(violations);  
db.TrustCode.Add(trust);  
await db.SaveChangesAsync();  
  
return Results.Created($" /trust/{trust.Id}", trust);  
});  
  
app.MapPut("/trust/{id}", async (int id, Trusted inputTr  
{  
    if (!MinimalValidation.TryValidate(inputTrusted, out var violations))  
        return Results.ValidationProblem(violations);  
  
    var trust = await db.TrustCode.FindAsync(id);  
  
    if (trust is null) return Results.NotFound();  
  
    app.MapGet("/trust/{id}", async (int id, TrustedDb d  
        await db.TrustCode.FindAsync(id) is Trusted trust  
        ? Results.Ok(trust)  
        : Results.NotFound());  
  
app.MapPost("/trust", async (Trusted trust, TrustedDb d  
{  
  
    if (!MinimalValidation.TryValidate(trust, out var violations))  
        return Results.ValidationProblem(violations);  
    db.TrustCode.Add(trust);  
    await db.SaveChangesAsync();  
  
    return Results.Created($" /trust/{trust.Id}", trust);  
});  
  
(int id, Trusted inputTr  
    validate(inputTrusted, out var violations)  
    Problem(violations);  
  
    tCode.FindAsync(id);  
  
    results.NotFound();  
}
```



## < Microsoft가 안전한 제품과 서비스를 구축하는 방법 >

보안 핵심 우선 순위에  
영향력을 발휘함.

보안 재단과 행동  
강령(비즈니스 행동 표준)  
과정 등 필요한 교육에서  
학습을 완료하고 적용함.

Microsoft Secure Future  
Initiative(SFI)  
프레임워크와 목표를  
학습함.

본인이 하는 일에서  
보안에 우선 순위를 두고,  
SFI(Secure Future  
Initiative)에 따라 제품과  
프로세스, 도구에 대한  
보안 접근 방식을  
예상하고 구현함.

MSProtect를 사용하여  
Microsoft 보안에 기여할  
수 있는 방법을 배움.

지금 보고를 통해  
의심스럽거나 악의적인  
사이버 활동, 시스템과  
데이터의 무결성에 관한  
우려 사항, 의심스러운  
디지털 또는 물리적 보안  
위반 및 취약점을 보고함.

아래 링크를 방문하여 자세히  
알아보세요



보안 핵심 우선 순위(내부 전용)



Microsoft 보안 정책(내부 전용)



Secure Future Initiative



지금 보고(내부 전용)



MSProtect(내부 전용)





# TRUST

## 정부 기관 및 지역사회와의 신뢰

전 세계의 법률 준수

25

접근 가능한 제품 및 서비스 설계

27

인권 존중과 증진

28

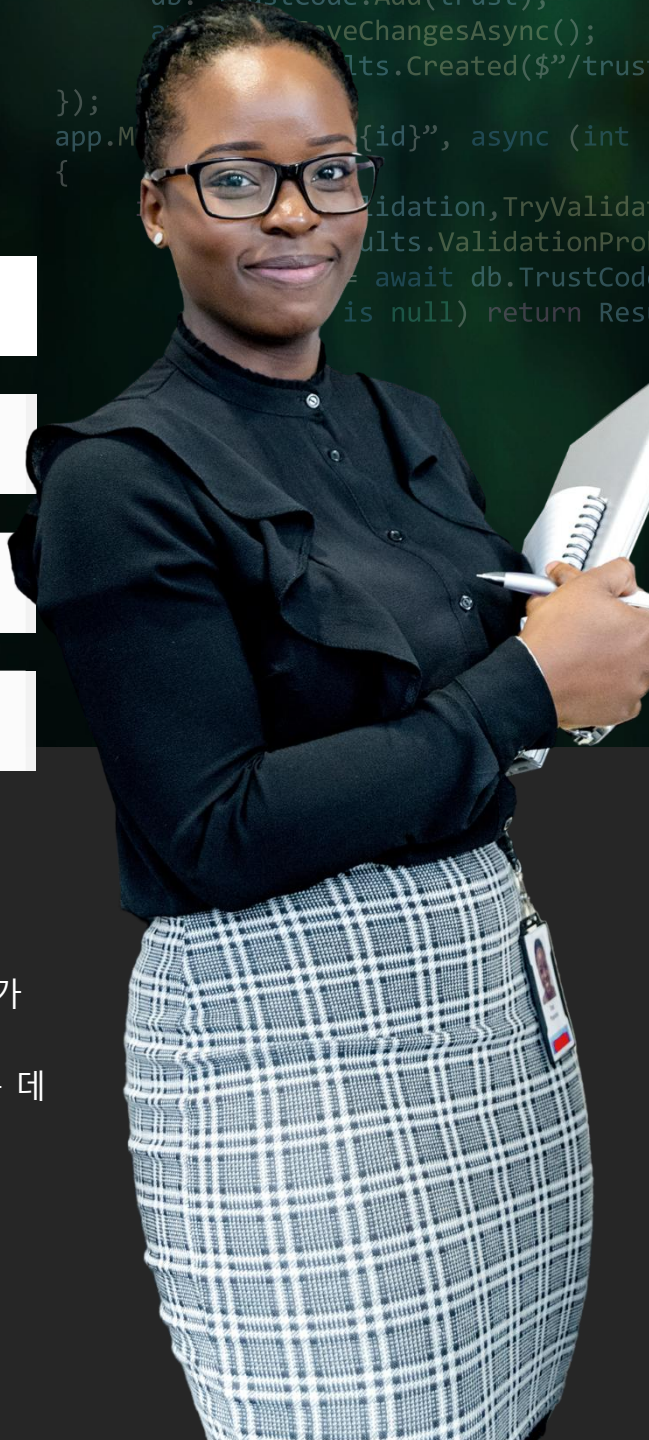
책임 있는 AI 개발 및 배포

29

결정을 내릴 때 스스로에게 물어보세요. 이것이  
정부와 지역사회와의 신뢰를 증진합니까?

우리의 사명을 달성하는 것은 정부와 지역 사회가  
우리가 사업을 운영하는 국가의 법률과 관습을  
준수하고 전 세계의 인권을 존중할 것임을 아는 데  
달려 있습니다.

```
app.MapGet("/trust/{id}", async.  
    await db.TrustCode.FindAsync(i  
        ? Results.Ok(trust)  
        : Results.NotFound());  
  
app.MapPost("/trust", async. (Trusted  
{  
    if (!MinimalValidation.TryVali  
        return Results.Validation  
    db.TrustCode.Add(trust);  
    await db.SaveChangesAsync();  
    Results.Created($""/trust  
});  
app.MapGet("/trust/{id}", async (int  
{  
    MinimalValidation, TryValida  
    Results.ValidationPro  
    = await db.TrustCod  
    is null) return Res
```





# 전 세계의 법률 준수

우리는 전 세계에서 활동하며 비즈니스를 수행하는 모든 곳에서 법률을 존중할 것을 약속합니다.

이것은 우리가 정부와의 신뢰를 구축하고, 우리가 살고 일하는 지역사회를 강화하며, 동일한 법률의 보호를 누리는 공정한 사회에 기여하는 데 도움이 됩니다.

```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>  
    await db.TrustCode.FindAsync(id) is Trusted trust  
    ? Results.Ok(trust)  
    : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
```





# <전 세계의 법률을 준수하는 방법>



특정 국가, 정부, 단체 및 개인과 비즈니스를 수행하는 것을 금지하는 세계 무역 통제 및 경제 제재를 존중하고 준수함으로써.



정부 대표 및 공무원과의 논의에서 정직하고 투명한 태도를 취함으로써.



정보에 대한 정부의 문의 요청에 진실되고, 적절하고, 신속하게 응답함으로써.



Microsoft 기술이 국제적으로 배포되고 사용될 수 있는 방법에 적용되는 법률을 존중하고 준수함으로써.



미국 및 Microsoft가 사업을 운영하는 모든 장소의 법률 및 규정을 준수함으로써.





# 접근 가능한

## 제품 및 서비스 설계

우리는 제품과 서비스 개발에서 포용적인 설계를 우선시합니다.

우리는 전 세계 10억 명이 넘는 장애인들을 위해 포용적으로 설계할 수 있는 기회와 책임이 있습니다. 접근성에 투자하면 Microsoft에서 고객이 포용성 목표를 달성하도록 지원하는 데 도움이 됩니다. 기술을 사용하는 사람들의 다양성이 기술에 반영된다면 한계를 허물 수 있다고 우리는 믿습니다.

### <접근 가능한 설계를 하는 방법>



다양한 관점을 가진 사람들을 포함시키고 배우며, 포용적으로 설계하고, 설계 프로세스의 시작부터 사람들을 중심에 둡으로써.



시력, 청각, 학습, 이동성, 정신 건강, 연령 관련, 감각 및 언어 장애가 있는 사용자를 포함한 모든 사용자가 제품을 이용할 수 있도록 함으로써.

Microsoft 접근성



접근성 리소스 및 학습



공공 정책 권장 사항



# 인권 존중과 증진

우리는 인권을 존중하고 증진하여 전 세계 사람들이 인류의 이익을 위해 기술을 사용할 수 있도록 지원하기 위해 최선을 다하고 있습니다.

## <인권을 존중하고 증진하는 방법>



우리는 국제 법률, 원칙 및 규범에 따른 법규와 인권을 존중하고 옹호합니다.



우리는 장기적으로 볼 때, 심각한 인권 문제를 안고 있는 국가의 사람들, 정부 및 기타 이해관계자들에게 우리의 존재 그리고 그들과의 책임감 있는 참여가 종종 인권 증진에 대한 더 큰 약속을 의미한다고 믿습니다.



사용자 데이터를 제공하거나 콘텐츠를 제거하라는 정부의 요구 사항에 직면할 때, 당사는 정부의 요구가 유효하고, 구속력이 있으며, 그렇지 않으면 법규를 준수하는지 여부를 확인합니다.



우리는 우리 기술이 인권에 미치는 영향을 평가하기 위해 실사를 실시합니다. 우리는 이해 관계자들로부터 우리의 사고에 도전하고, 정책과 관행을 개발 및 개선하고, 기술을 개선하고, 인권에 대한 우리의 헌신을 이행하기 위해 기술을 제공하는 방법을 배웁니다.

인권을 위한 우리의 노력



# 책임 있는 AI

## 개발 및 배포

Microsoft는 AI 원칙, 책임 있는 AI 표준, 회사 사명, 가치, 인권에 대한 헌신 등에 일관된 방식으로 AI 기술을 개발하고 배포하기 위해 최선을 다하고 있습니다.

Microsoft는 사람들이 신뢰하는 시스템을 만들기 위해 노력합니다. 즉, 사람과 목표를 시스템 설계의 중심에 두고 공정성, 신뢰성 및 안전, 개인정보보호 및 보안, 포용성, 투명성, 책임성과 같은 지속적인 가치를 존중합니다.



## <책임 있는 AI를 개발하고 배포하는 방법>

AI 기술을 개발하고 배포할 때 Microsoft의 책임 있는 AI 표준 준수.

책임 있는 AI 사무소에 민감 사례를 보고하고 수령한 가이드 준수.

추가 지침이 필요할 때 도움을 요청.

아래 링크를 방문하여 자세히 알아보기 ↓

책임 있는 AI 사무소  
(내부 전용)



Microsoft의 책임 있는 AI 정책  
(내부 전용)



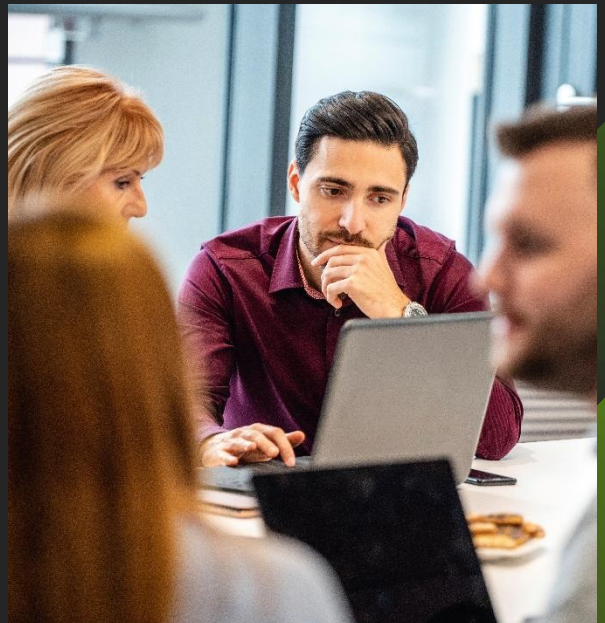
민감 사례 보고  
(내부 전용)



Microsoft의 책임 있는 AI 원칙



Microsoft의 책임 있는 AI 표준



# TRUST

서로에 대한 신뢰

다양성 및 포용성 육성

32

안전하고 생산적인 업무 환경

34

이해 충돌 방지

36

결정을 내릴 때 스스로에게 물어보세요. 이것이 우리가 One Microsoft로서 함께 일하는 방식을 개선합니까?

가장 강력한 기여는 서로를 신뢰하는 팀, 즉 서로 개방적이고 정직하며 건설적인 팀에서 비롯됩니다.

상호 존중, 포용 및 협력의 분위기를 조성하기 위해 우리 모두가 함께 노력해야 합니다.

```
app.MapGet("/trust/{id}", async (int id, Tr
    await db.TrustCode.FindAsync(id) is TrustCode trust
    ? Results.Ok(trust)
    : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust,
{
    if (!MinimalValidation.TryValidate(trust))
        return Results.ValidationProblem();
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
});
app.MapPut("/trust/{id}", async (int id, Tr
{
    if (!MinimalValidation.TryValidate(input))
        return Results.ValidationProblem(validationProblems);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust.Name = input.Name;
    trust.Description = input.Description;
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```

```
app.MapGet("/trust/{id}", async (int id, Tr
    await db.TrustCode.FindAsync(id) is TrustCode trust
    ? Results.Ok(trust)
    : Results.NotFound());
```

```
app.MapPost("/trust", async (Trusted trust,
{
    if (!MinimalValidation.TryValidate(trust))
        return Results.ValidationProblem();
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
});
app.MapPut("/trust/{id}", async (int id, Tr
{
    if (!MinimalValidation.TryValidate(input))
        return Results.ValidationProblem(validationProblems);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust.Name = input.Name;
    trust.Description = input.Description;
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```

```
if (!MinimalValidation.TryValidate(input))
    return Results.ValidationProblem(validationProblems);
var trust = await db.TrustCode.FindAsync(id);
if (trust is null) return Results.NotFound();
trust.Name = input.Name;
trust.Description = input.Description;
db.TrustCode.Update(trust);
await db.SaveChangesAsync();
return Results.NoContent();
});
```



다양성 및  
포용성  
육성

우리는 차이에도  
불구하고 우리의 차이  
때문에 더 잘 함께 일할 수  
있습니다.

우리는 우리 각자가  
가져오는 다양한 기술,  
경험 및 배경을 통해 지구  
상의 모든 사람들에게  
가장 잘 봉사하고 우리  
자신의 문화를 풍요롭게  
한다고 믿습니다.

```

        return Results.ValidationProblem(
            db.TrustCode.Add(trust);
            await db.SaveChangesAsync();
            return Results.Created($"/trust/{trust.Id}");
    });
    app.MapPut("/trust/{id}", async (int id, Trusted trust) => {
        if (!MinimalValidation.TryValidate(inputTrust, out var trust))
            return Results.ValidationProblem(validationProblems);
        var trust = await db.TrustCode.FindAsync(id);
        if (trust is null) return Results.NotFound();

        trust.Name = inputTrust.Name;
        trust.Password = inputTrust.Password;
        trust.Code = inputTrust.Code;
        await db.SaveChangesAsync();
        return Results.Created($"/trust/{trust.Id}");
    });
    app.MapGet("/trust/{id}", async (int id, Trusted trust) => {
        await db.TrustCode.FindAsync(id) is Trusted trust
            ? Results.Ok(trust)
            : Results.NotFound();
    });
    app.MapPost("/trust", async (Trusted trust, Trusted trust) => {
        if (!MinimalValidation.TryValidate(trust, out var trust))
            return Results.ValidationProblem(validationProblems);
        db.TrustCode.Add(trust);
        await db.SaveChangesAsync();
        return Results.Created($"/trust/{trust.Id}");
    });
    app.MapPut("/trust/{id}", async (int id, Trusted trust) => {
        if (!MinimalValidation.TryValidate(trust, out var trust))
            return Results.ValidationProblem(validationProblems);
        var trust = await db.TrustCode.FindAsync(id);
        if (trust is null) return Results.NotFound();

        trust.Name = inputTrust.Name;
        trust.Password = inputTrust.Password;
        trust.Code = inputTrust.Code;
        await db.SaveChangesAsync();
        return Results.Created($"/trust/{trust.Id}");
    });
}

```





## <다양성과 포용성을 육성하는 방법>



자격을 갖춘 모든 지원자와 직원에게 동등한 고용 기회를 제공함으로써.



장애인을 고용하고 자격을 갖춘 직원에게 합리적인 편의를 제공하기 위해 최선을 다함으로써.



우리의 무의식적인 편견을 검토하고 모든 직원을 환영한다고 느끼게 하는 포용적인 문화를 창출하기 위한 조치를 취함으로써.



나이, 조상, 피부색, 가족 또는 의료 휴가, 성 정체성 또는 성적 표현, 유전 정보, 결혼 상태, 건강 상태, 출신 국가, 신체적 또는 정신적 장애, 정치적 소속, 보훈자 신분, 인종, 종교, 성별(임신 포함), 성적 취향 또는 관련 법률, 규정 및 조례에 의해 보호되는 기타 특성에 따라 차별하지 않음으로써.



아래 링크를 방문하여 자세히 알아보기 ↓

글로벌 다양성 및 포용성



고용기회 균등 정책(내부 전용)





# 안전하고 생산적인 업무 환경에 기여

우리는 모든 사람이 최선을 다할 수 있도록 힘을  
실어주는 업무 환경을 위해 노력하고 있습니다.  
안전한 직장은 신뢰를 불러일으키고 우리  
모두가 기여하고 성공할 수 있게 해 줍니다.

차별, 괴롭힘 및 안전하지 않은 근로 조건은  
우리가 함께 성취할 수 있는 것을 감소시킵니다.

아래 링크를 방문하여 자세히 알아보기 ↓

괴롭힘 및 차별 방지 정책  
(내부 전용)



엔터테인먼트 정책(내부 전용)



직장 폭력 정책  
(내부 전용)




마약 없는 직장 정책  
(내부 전용)



```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) => {
    await db.TrustCode.FindAsync(id) is Trusted trust
        ? Results.Ok(trust)
        : Results.NotFound();
});

app.MapPost("/trust", async (Trusted trust, TrustedDb db) => {
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($" /trust/{trust.Id}", trust);
});

app.MapPut("/trust/{id}", async (int id, Trusted inputTrust, TrustedDb db) => {
    if (!MinimalValidation.TryValidate(inputTrust, out var violations))
        return Results.ValidationProblem(violations);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
    trust = inputTrust;
    db.TrustCode.Update(trust);
    await db.SaveChangesAsync();
    return Results.NoContent();
});
```



```
app.MapGet("/trust/{id}", async. (int id, TrustedDb db) =>
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost("/trust", async. (Trusted trust, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(trust, out var violations))
        return Results.ValidationProblem(violations);
    db.TrustCode.Add(trust);
    await db.SaveChangesAsync();
    return Results.Created($"trust/{trust.Id}", trust);
});

app.MapPut("/trust/{id}", async (int id, Trusted inputTrusted, TrustedDb db) =>
{
    if (!MinimalValidation.TryValidate(inputTrusted, out var violations))
        return Results.ValidationProblem(violations);
    var trust = await db.TrustCode.FindAsync(id);
    if (trust is null) return Results.NotFound();
```

## <안전하고 생산적인 업무 환경에 기여하는 방법>

— — — —

건전한 안전 및 건강 관행을  
운영에 통합하고 직장 안전  
규정을 준수함으로써.

— — — —

다른 사람들을 존중하고  
괴롭힘이나 차별을 용납하지  
않음으로써.

— — — —

문제를 정중하게 해결하고  
폭력의 행위나 위협에 결코  
의지하지 않음으로써.

— — — —

마약이나 술에 취해 일하지  
않음으로써.



# 이해 충돌 방지

이해 충돌은 우리의 개인적인 관계나 재정적 이익이 직무 책임과 겹칠 때 발생할 수 있습니다.

잠재적인 이해 충돌을 신중하게 탐색하지 않으면 이러한 상황이 의사 결정에 영향을 미치고 팀 내의 신뢰를 훼손하며 Microsoft의 평판을 손상시킬 수 있습니다.

```
app.MapGet("/trust/{id}", async (int id, TrustedDb db) =>
{
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());

app.MapPost("/trust", async (Trusted trust, TrustedDb db) =>
{
    if (!ModelState.IsValid)
    {
        return Results.BadRequest(ModelState);
    }
    await db.TrustCode.AddAsync(trust);
    return Results.CreatedAt("/trust", db.TrustCode, trust);
});

app.MapPut("/trust/{id}", async (int id, Trusted trust, TrustedDb db) =>
{
    if (!ModelState.IsValid)
    {
        return Results.BadRequest(ModelState);
    }
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.PutAsync(db.TrustCode, trust)
    : Results.NotFound();
});
```



## <이해 충돌을 방지하는 방법>



Microsoft의 최선의 이익을 위해 행동함으로써.



다른 회사와 관련한 개인적인 관계나 재정적 이익이 우리가 업무에서 결정을 내리는 방식에 영향을 미칠 수 있는 상황을 피하거나 중단함으로써.

(목록은 아래에서 계속) ↓



외부 관계에 의해 영향을 받지 않고 결정할 수 있을 것이라고 확신하더라도 이해 충돌이 존재할 수 있음을 이해함으로써.



이해 상충이 발생하면 이를 공개하고 조언이나 승인을 얻음으로써.



외부에서 업무를 수행하기 전에 Microsoft는 해당 작업이 Microsoft의 비즈니스 이익에 해를 끼치거나 서명한 직원 계약을 위반하지 않도록 보장함으로써.

아래 링크를 방문하여 자세히 알아보기



```
app.MapGet("/trust/{id}", async (int id, TrustedDb
    await db.TrustCode.FindAsync(id) is Trusted trust
    ? Results.Ok(trust)
    : Results.NotFound());
```

이해 충돌  
(내부 전용)



직장 외부 업무 정책  
(내부 전용)



이해 충돌 승인 도구  
(내부 전용)



# TRUST

## 투자자 및 대중과의 신뢰

내부 정보를 거래하지 않기

39

정확한 기록 및 계약 유지

41

대중에게 정확하게 의사 소통

43

Microsoft의 리소스 보호

45

기밀 정보 및 IP 보호

46

**결정을 내릴 때 스스로에게 물어보세요.  
이것이 투자자 및 대중에 대한 Microsoft의 평판을  
강화하거나 해칠 수 있습니까?**

투자자들에게 우리가 책임감 있고 사려 깊은 방식으로  
가치를 창출할 것이라는 확신을 주어야 합니다.

Microsoft의 자산과 정보를 보호하고, Microsoft가  
회사로서 어디로 향하고 있는지 정확하게 말하고,  
정직하게 투자자 및 대중과 의사 소통함으로써 신뢰를  
얻으세요.



```
return Results.Validation  
db.TrustCode.Add(trust);  
await db.SaveChangesAsync();  
return Results.Created($"tr  
});  
app.MapPut("/trust/{id}", async (int  
{  
    if (!MinimalValidation, TryValid  
        return Results.ValidationPr  
    var trust = await db.TrustC  
    if (trust is null) return R
```



# 내부 정보를 거래하지 않기

투자자 및 대중과의 신뢰를 유지하기 위해  
우리는 증권법을 존중하며, 이는 우리가  
귀중한 비공개 정보를 소유하고 있는 동안  
거래하지 않는다는 것을 의미합니다.

업무 중에 우리는 Microsoft 또는 Microsoft와  
비즈니스를 수행하는 회사에 대한 공개되지  
않은 귀중한 정보를 알게 될 수 있습니다.  
합리적인 투자자가 회사의 유가 증권을 사고  
팔 것인지 결정할 때 정보를 중요하게  
고려하는 경우 정보는 "귀중"합니다.  
투자하는 대중에게 정보가 광범위하게  
전달되지 않은 경우 정보는 "비공개"입니다.

이 정보를 기반으로 Microsoft 주식 또는 다른  
회사의 주식을 거래하는 것은 투자자 및  
대중과의 신뢰를 깨뜨릴 뿐만 아니라  
불법이며 회사 정책을 위반하고 다른  
투자자에게 불공평한 것입니다.



아래 링크를 방문하여 자세히 알아보기 ↓

내부자 거래 정책  
(내부 전용)



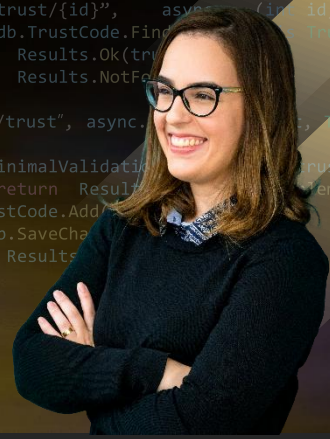
제한된 거래 창구 정책  
(내부 전용)



# <신뢰를 구축하는 방법>

```
app.MapGet("/trust/{id}", async (id, trustedDb db) =>
{
    await db.TrustCode.Find(trustId: id, trustedDb db)
    ? Results.Ok(trust)
    : Results.NotFound();

    app.MapPost("/trust", async (trust, trustedDb db) =>
    {
        if (!MinimalValidation(trust, out var violations))
        {
            return Results.BadRequest(violations);
        }
        db.TrustCode.Add(trust);
        await db.SaveChangesAsync();
        return Results.Created($"{id}", trust);
    });
});
```



Microsoft의 귀중하거나 비공개인 정보를 기반으로, 또는 고용을 통해 취득한 유가 증권을 구매 또는 판매하지 않음으로써.



다른 사람(예: 친구, 배우자 또는 중개인)에게 귀중한 비공개 정보에 관한 "팁"을 제공하지 않음으로써.



정보 자체를 공유하지 않더라도 Microsoft의, 또는 고용을 통해 취득한 귀중한 비공개 정보를 기반으로 증권을 거래하도록 다른 사람을 추천하거나 제안하지 않음으로써.

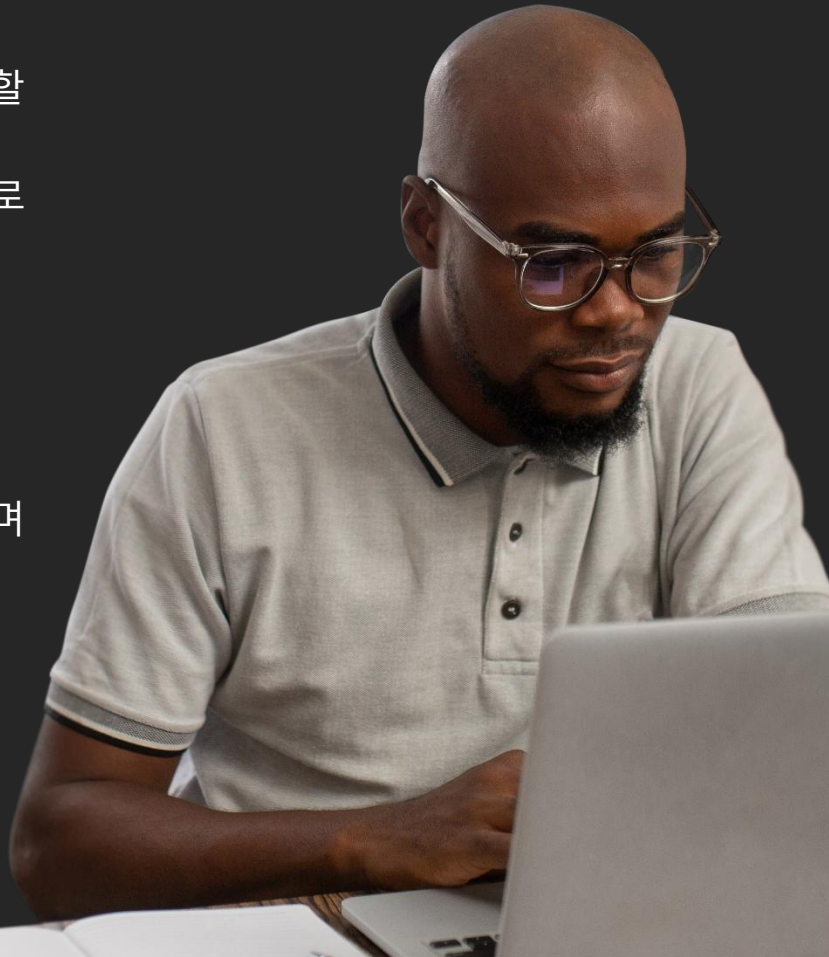


# 정확한 기록 및 거래 유지

우리의 기록은 분명하고  
정확하며 완전합니다.

투자자는 정확하고 쉽게 이해할  
수 있는 정보에 의존하여 재무  
결과와 우리가 회사로서 어디로  
향하고 있는지 이해하고 그  
방향에 대한 확신을 갖습니다.

정확한 기록을 유지하는 것은  
투자자의 신뢰를 유지하고  
올바른 비즈니스 결정을 내리며  
상장 회사로서의 규제 의무를  
이행하는 데 중요합니다.





## <정확한 기록과 거래를 유지하는 방법>



우리는 관련 법률, 규정 및 회계 관행에 따라 금융 거래 및 비즈니스 정보를 정직하고 정확하게 기록하고 보고합니다.



우리는 파트너와 고객에 대한 우리의 약속에 대해 투명성을 지키며, 서면 계약은 거래의 실물 경제학을 반영합니다.



부수적인 합의 또는 다른 "장부에 기록되지 않은" 약정을 맺지 않습니다.



문서를 보관하고 처분할 때 우리는 법적 의무와 장기적인 비즈니스 요구를 충족하는 확립된 기업에 대한 보존 요구 사항을 준수합니다.



적절한 권한이 있는 경우에만 금융 거래를 수행하고 Microsoft의 회계 관행을 준수하여 이를 기록합니다.



금융 조직의 CEO, CFO, 기업 관리자 및 기타 직원은 재무 관련 직업 행동 강령을 준수합니다.



표준 계약 및 계약 프로세스를 사용하거나 맞춤형 조건에 대한 승인을 얻습니다.

아래 링크를 방문하여 자세히 알아보기 ↓

재무 관련 직업  
행동 강령  
(내부 전용)



맞춤 계약 및 부가  
협약 정책  
(내부 전용)



문서 보존 정책  
(내부 전용)





## 대중에게 정확하게 의사 소통

우리의 공개 진술과 의사 소통은 정직하며  
오해의 소지가 없습니다.

우리가 무엇을 말하는지와 그것을 전달하는  
방식은 중요합니다. 투자자와 대중이 우리가  
말하는 것을 아는 경우에만 그들의 신뢰를  
얻고 유지할 수 있습니다.



## <대중에게 정확하게 의사 소통하는 방법>



미국 증권  
거래위원회(US  
Securities and  
Exchange  
Commission)에  
제출한 보고서 및  
문서 또는 기타 공개  
진술서에서 완전하고  
정확하며 적시에 재무  
정보를 공개합니다.



우리는 우리의 제품과  
서비스에 대한 모든  
커뮤니케이션이  
정직하고 정확한지  
확인합니다.



투자자 및 기타 이해  
관계자의 피드백과  
질문을 주의 깊게  
경청합니다.



소셜 미디어에서  
제품과 서비스를  
보증하는 경우, 회사  
제휴를 분명히 하고  
승인된 제품 또는  
서비스에 대해 정직한  
의견이나 신념을  
진술합니다.



권한이 없는 한  
회사를 대신하여  
말하지 않습니다.  
애널리스트 및 미디어  
문의는 투자자 관계  
또는 홍보 팀에  
전달합니다.



우리의 광고 및 홍보  
자료는 정확하며 거짓  
주장을 하지 않습니다.

아래 링크를 방문하여 자세히 알아보기 ↓

재무 정보 공개 정책  
(내부 전용)



대중 홍보 정책  
(내부 전용)



소셜 미디어 지침  
(내부 전용)







# Microsoft의 리소스 보호

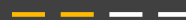
우리는 Microsoft의 리소스를 책임지는 담당자입니다. 매일 업무를 수행하기 위해 회사 리소스와 자금을 사용합니다. 이러한 리소스는 Microsoft에 속하며 우리는 투자자들을 위해 책임감 있게 그리고 신뢰를 유지하고 가치를 더하는 방식으로 이러한 리소스를 사용할 의무가 있습니다.

## <Microsoft의 리소스를 보호하는 방법>

비즈니스 관련 데이터가 Microsoft의 자산이며 정책 및 관련 법률에 따라 액세스, 보존 및 검토될 수 있음을 이해함으로써.



Microsoft의 자금을 현명하게 지출하고 낭비와 남용을 방지함으로써.



비즈니스 통신 및 데이터에 개인 계정이나 서비스가 아닌 회사 계정 및 서비스를 사용함으로써.



회사가 제공한 기술 리소스를 이용하고 직무 책임을 방해하거나 업무 환경을 해치지 않는 경우에만 회사 방침에 따라 개인적인 이유로 이러한 리소스를 사용할 때 올바른 판단을 내림으로써.



아래 링크를 방문하여 자세히 알아보기 ↓

기술의 책임 있는 사용 정책  
(내부 전용)



경비 정책  
(내부 전용)



# 기밀 정보 및

지적 재산 보호



Microsoft는 Microsoft 또는 다른  
회사에 속한 정보 및 아이디어의  
비즈니스 가치를 보호하고 존중합니다.

Microsoft의 기밀 정보를 공개하면 경쟁 우위를  
잃을 수 있습니다.

우리와 비즈니스를 수행하는 회사로부터 알아낸  
기밀 정보를 공개하면 신뢰를 잃게 됩니다. 우리  
회사와 다른 기업의 정보와 아이디어를 보호하는  
것은 비즈니스 성공에 매우 중요하며 신뢰할 수  
있는 파트너로서의 평판을 쌓을 수 있습니다.

## <기밀 정보 및 IP를 보호하는 방법>



모범 사례를 적용하여 기밀 정보에 대한 액세스를 보호하고 공용 공간에서 또는 기밀 정보를 알 필요가 없는 동료와 논의하는 것을 피함으로써.



Microsoft 비즈니스 이외의 용도로 기밀 정보를 사용하지 않고 Microsoft에서 근무를 중단하더라도 기밀성을 유지함으로써.



혁신의 가치를 보호하는 저작권, 상표 및 기타 형태의 지적 재산권을 통해 Microsoft 자체의 지적 재산권을 보존함으로써.



타인의 기밀 유지 및 지적 재산권을 존중하고 타인의 기밀 정보를 허가 없이 사용하지 않음으로써.



라이선스나 CELA의 허가 없이 소프트웨어, 그래픽, 비디오, 음악과 같은 저작권이 있는 타사 자료를 사용하거나 복사하지 않음으로써.

아래 링크를 방문하여 자세히 알아보기 ↓

기밀 정보  
정책(내부 전용)



저작권 정책  
(내부 전용)





# TRUST

## 대표자와의 신뢰

신뢰할 수 있는 대표자 활용

49

책임감 있게 선물, 환대 및 여행 제공

51

무결성을 갖춘 공급업체 선택

53

**결정을 내릴 때 스스로에게 물어보세요.  
이것이 우리의 대표자 및 고객과의 장기적인  
협력 관계를 강화하게 될까요?**

Microsoft는 높은 윤리적 기준을 고수하고 있으며 파트너, 공급업체, 컨설턴트, 그리고 우리를 대신하여 일하는 다른 회사로 이루어진 대표자가 우리가 스스로 정한 표준을 충족할 것으로 기대합니다.

우리는  
우리가 대표자에게 가치를 돌려줄 때 그들의 신뢰를 얻으며 우리 자신에게도 요구하지 않을 것을 그들에게 요구해서는 안 됩니다.



## 신뢰할 수 있는 대표자 활용

우리는 우리를 대신하여 일할 때 윤리적으로 비즈니스를 수행하기로 약속하는 정직하고 윤리적인 대표자를 이용하기 위해 노력합니다. 성공을 위해 Microsoft의 비즈니스는 파트너, 공급업체, 컨설턴트와 같은 대규모 대표자 네트워크에 의존합니다.

그들의 행동은 우리의 행동만큼 회사의 평판을 반영합니다. 우리의 대표자가 진실되게 비즈니스를 수행할 때, 우리는 고객 및 대중에게 확립한 신뢰를 유지하고 강화합니다.





## <신뢰할 수 있는 대표자를 활용하는 방법>



대표자가 우리를 대신하여 일하기 전에 우리의 자격과 평판을 이해함으로써.



거래 또는 트랜잭션에 특정 대표자를 포함시키는 이유를 이해하고 적절한 경우에만 그렇게 함으로써.



대표자들과 신뢰를 쌓고 법이나 정책에 따라 허용되지 않는 일을 하도록 요청하거나 제안하지 않음으로써.



파트너 또는 리셀러가 원하지 않거나 필요로 하지 않는 제품이나 서비스를 주문하도록 압력을 가하지 않고, 거부할 경우 보복하지 않음으로써.



대리인이 Microsoft를 대표하는 동안 부패하거나 비윤리적인 행동에 관여할 수 있다는 징후를 보고함으로써.

아래 링크를 방문하여 자세히 알아보기 ↓

신뢰할 수 있는 대표자 정책  
(내부 전용)



윤리적 판매 정책  
(내부 전용)





# 책임감 있게 선물, 환대 및 여행 제공

우리는 비즈니스 호의를 주고받을 때 건전한 판단과 중용을 보여주기 위해 노력합니다.

선물, 환대, 여행을 주고받는 것은 Microsoft와 우리가 비즈니스를 수행하는 사람들 사이에 강력한 업무 관계와 선의를 구축할 수 있습니다.

그러나 사치스럽거나 투명성이 부족하거나 합법적인 목적이 부족한 선물, 식사 또는 여행은 뇌물로 간주되거나 단순히 부적절하다고 간주될 수 있습니다. 이러한 것들은 신뢰를 침식하고 우리의 비즈니스에 해를 끼칩니다.



## <책임감 있게 선물, 환대 및 여행을 제공하는 방법>



부패와 상업적 뇌물수수로부터 자유로운 긍정적인 관계를 발전시킴으로써.



선물, 환대 또는 여행을 제공하거나 수락할 때 합리적이고 적절하며 합법적인 비즈니스 목적이 있는지 확인함으로써.



타사로부터 선물, 환대 또는 여행을 요청하지 않거나 우리와 비즈니스를 수행하기 위해 무언가를 제공할 의무가 있다고 느끼게 하지 않음으로써.



파트너 또는 공급업체와 같은 대표자에게 우리를 대신하여 선물, 환대 또는 여행을 제공하도록 요청하지 않음으로써.



현금을 주거나 받지 않음으로써.

아래 링크를 방문하여 자세히 알아보기 ↓

상업적 수혜자에게 제공되는 선물, 환대 및 여행 제공 정책(내부 전용)



선물, 환대 및 여행 수락 정책(내부 전용)



국가 공무원에게 제공되는 선물, 환대 및 여행 정책(내부 전용)



## 무결성을 갖춘 공급업체 선택

성공을 위해 우리는 공급업체 관계에 의존합니다. 사명을 달성하기 위해 우리가 하듯이 고객과의 신뢰를 구축하고, 훌륭하게 업무를 수행하고, 법규를 따를 헌신적인 공급업체가 필요합니다.



### <무결성을 갖춘 공급업체를 선택하는 방법>

업무에 가장 적합한 공급업체를 선택하고 편애에 근거하지 않고 검증된 실적, 무결성에 대한 평판 및 기타 장점을 신중하게 고려함으로써.

공급망에서 아동 노동, 인신 매매 및 기타 노동 남용을 제거하기 위해 노력함으로써.

아래 링크를 방문하여 자세히 알아보기 ↓

공급업체 계약 정책  
(내부 전용)



조달 정책  
(내부 전용)





# 신뢰 코드

# TRUST CODE



## 직원의 책임

직원은 신뢰 코드, Microsoft 정책 및 법률을 준수해야 하며 이에 따라 다음을 수행해야 합니다.

> 이 신뢰 코드 및 귀하의 직무에 적용되는 정책, 법률 및 규정을 읽고, 이해하고, 준수합니다.

> 목소리를 내어 신뢰 코드, Microsoft 정책 및 법률 및 규정 요구 사항을 위반할 가능성이 있는 경우 발언합니다.

> 진실되게 행동하고 내부 조사에 전적으로 협조합니다. 정보를 숨기거나 파괴하지 않습니다.

> 신뢰 코드에 대한 교육을 완료하고, 신뢰 코드를 이해하고 준수하기로 약속했음을 증명합니다. 신뢰 코드를 읽거나 증명하지 못한다고 해서 이러한 책임에서 벗어날 수 있는 것은 아닙니다.

Microsoft의 비즈니스 행동 표준과 규정 준수 및 윤리 프로그램은 Microsoft 이사회에서 보증하며 이들에 의해 전폭적인 지원을 받습니다. 이사회 및 경영진은 규정 준수 및 윤리 프로그램과 이러한 표준과 관련한 규정 준수를 감독할 책임이 있습니다. 이러한 표준은 Microsoft, 이사회 및 Microsoft의 모든 직원, 이사, 임원에게 적용됩니다.

"Microsoft"가 언급될 때 이는 Microsoft Corporation과 투표 통제권의 50% 이상을 직접 또는 간접적으로 소유하고 있는 모든 자회사 및 계열사를 의미합니다. "직원"에 대한 언급에는 Microsoft의 이사회, 직원, 이사 및 임원이 포함됩니다.

이사회만이 이사 또는 임원에 대한 본 표준의 조항을 철회할 수 있습니다. 이사 또는 임원에게 부여된 모든 철회는 나스닥 상장 요건 및 관련 법률, 규칙 및 규정에 따라 대중에 공개됩니다.

이러한 표준은 우리에게 중요합니다. 표준을 위반하면 최대 고용 종료를 포함한 징계 조치가 취해질 수 있습니다. 직원은 항상 상식과 올바른 판단력을 발휘하고 특정 행동을 금지하는 정책의 존재와 관계없이 책임감 있게 행동해야 합니다.

문제를 해결할 수 있도록 무언가가 옳지 않을 때 소리 내어 말해 주세요.

[무결성 우려 사항 보고 >](#)

# TRUST CODE

Microsoft의 비즈니스 행동 표준