

# [멋사 13기 나한진]기초 테스트 오답 노트

## 1번 문제

▼ 웹 프론트에서 사용하는 언어가 아닌 것을 고르시오

1. HTML
2. CSS
3. Java
4. JavaScript

▼ 정답

3. Java

▼ 풀이

1. **HTML**: HyperText Markup Language의 약자로, 웹 페이지의 구조를 정의하는 데 사용됩니다. 웹 프론트엔드의 핵심 언어입니다.
2. **CSS**: Cascading Style Sheets의 약자로, 웹 페이지의 스타일과 레이아웃을 꾸미는 데 사용됩니다. 웹 프론트엔드에서 필수적입니다.
3. **Java**: 객체 지향 프로그래밍 언어로, 주로 백엔드 개발이나 서버 측 애플리케이션에서 사용됩니다. 웹 프론트엔드에서는 직접적으로 사용되지 않습니다.
4. **JavaScript**: 웹 페이지에 동적 기능과 인터랙티브 요소를 추가하는 데 사용되는 언어로, 웹 프론트엔드의 핵심 언어 중 하나입니다.

따라서 웹 프론트에서 사용하는 언어가 아닌 것은 **3. Java**입니다.

## 2번 문제

▼ 다음 중 올바르지 않은 것을 고르시오

1. HTML은 웹 페이지의 전체적인 틀을 구성하는 역할을 한다.
2. CSS는 cross side scripting의 약자이다.
3. Java Script는 정적인 웹페이지를 동적으로 만들어주는 역할을 한다.

4. HTML은 HyperText Markup Language의 약자이다

▼ 정답

2.

▼ 풀이

1. HTML은 웹 페이지의 전체적인 틀을 구성하는 역할을 한다.

- 맞습니다. HTML(HyperText Markup Language)은 웹 페이지의 구조와 기본 틀을 정의하는 데 사용됩니다.

2. CSS는 cross side scripting의 약자이다.

- 틀렸습니다. CSS는 **Cascading Style Sheets**의 약자로, 웹 페이지의 스타일과 레이아웃을 담당합니다. "Cross side scripting"은 오타로 보이며, 아마도 "Cross-Site Scripting"(XSS)을 의미한 것 같지만, 이는 보안 공격의 일종이지 CSS와는 무관합니다.

3. Java Script는 정적인 웹페이지를 동적으로 만들어주는 역할을 한다.

- 맞습니다. JavaScript는 웹 페이지에 동적 기능과 인터랙티브 요소를 추가하는 데 사용되며, 정적인 페이지를 동적으로 만듭니다. "Java Script"가 아니라 "JavaScript"가 정확한 표기지만, 여기서는 의미 전달에 문제가 없으므로 올바른 것으로 간주합니다.

4. HTML은 HyperText Markup Language의 약자이다.

- 맞습니다. HTML은 HyperText Markup Language의 약자로, 웹 페이지의 구조를 정의하는 마크업 언어입니다.

따라서 옳바르지 않은 것은 2. CSS는 cross side scripting의 약자이다입니다.

4번문제

▼ 아래 보기의 빈칸을 채우시오.

```
<!DOCTYPE html>
<html>
<head>
<>페이지 제목<>
</head>
<body>
<h1>이것은 제목입니다</h1>
<p>이것은 단락입니다</p>
```

```
</body>
</html>
```

▼ 정답

```
<title></title>
```

▼ 풀이

페이지 제목 부분에는 <title></title>이 들어가야된다.

제목 <h1></h1>, 단락<p></p>

5번문제

▼ 다음 중 CSS에서 padding 속성을 적용했을 때 올바른 것을 고르시오.

1. 요소의 외부 여백을 설정
2. 요소의 내부 여백을 설정
3. 요소의 테두리 두께를 조정
4. 요소의 배경 색상을 변경

▼ 정답

2. 요소의 내부 여백을 설정

▼ 풀이

CSS에서 `padding` 속성은 요소의 **내부 여백**을 설정하는 데 사용됩니다. 이를 통해 요소의 콘텐츠와 테두리 사이의 간격을 조정할 수 있습니다. 주어진 선택지를 하나씩 살펴봅시다:

1. 요소의 외부 여백을 설정

- 틀렸습니다. 외부 여백은 `margin` 속성으로 설정합니다. `padding` 은 내부 여백을 다룹니다.

2. 요소의 내부 여백을 설정

- 맞습니다. `padding` 은 요소의 콘텐츠와 테두리 사이의 여백을 조정하는 속성입니다.

3. 요소의 테두리 두께를 조정

- 틀렸습니다. 테두리 두께는 `border-width` 와 같은 `border` 속성으로 설정합니다.

4. 요소의 배경 색상을 변경

- 틀렸습니다. 배경 색상은 `background-color` 속성으로 변경합니다.

따라서 올바른 것은 2. 요소의 내부 여백을 설정입니다.

## 7번 문제

▼ 다음 중 <img> 태그에서 alt 속성의 역할을 올바르게 설명한 것을 고르시오.

1. alt 속성을 이미지의 크기를 조정하는데 사용된다
2. alt 속성은 이미지가 로딩되지 않을 때 대체 텍스트를 제공한다.
3. alt 속성은 이미지의 해상도를 결정한다.
4. alt 속성은 이미지의 배경색을 지정한다.

▼ 정답

2번

▼ 풀이

HTML에서 <img> 태그의 alt 속성은 이미지와 관련된 대체 텍스트(alternative text)를 제공하는 역할을 합니다. 이제 주어진 선택지를 하나씩 검토해보겠습니다:

### 1. alt 속성을 이미지의 크기를 조정하는데 사용된다

- 틀렸습니다. 이미지 크기는 width 와 height 속성으로 조정하며, alt 속성은 크기와 무관합니다.

### 2. alt 속성은 이미지가 로딩되지 않을 때 대체 텍스트를 제공한다

- 맞습니다. alt 속성은 이미지가 로드되지 않을 경우(예: 파일 경로 오류, 느린 네트워크 등) 사용자에게 표시될 텍스트를 지정합니다. 또한, 스크린 리더를 사용하는 시각 장애인을 위해 이미지의 내용을 설명하는 데도 사용됩니다.

### 3. alt 속성은 이미지의 해상도를 결정한다

- 틀렸습니다. 이미지 해상도는 이미지 파일 자체의 속성으로, alt 속성과는 관련이 없습니다.

### 4. alt 속성은 이미지의 배경색을 지정한다

- 틀렸습니다. 배경색은 CSS의 background-color 속성으로 설정하며, alt 속성은 텍스트 정보를 제공하는 데 사용됩니다.

따라서 올바른 설명은 2. alt 속성은 이미지가 로딩되지 않을 때 대체 텍스트를 제공한다입니다.

## 11번 문제

▼ 다음 중 올바르지 않은 것을 고르시오.

1. PHP는 개발을 위해 고안된 서버 측 스크립트 언어이자 수많은 웹 시스템의 기반이 되는 언어이다.
2. 파이썬은 가독성이 높고 다른 언어나 라이브러리에 접근하기 쉬운 언어이다.
3. Java는 객체 지향 프로그래밍 언어이다.
4. 파이썬은 빠른 속도를 요구하는 프로그램에 적합하다.

▼ 정답

4번

▼ 풀이

주어진 선택지를 하나씩 검토해보겠습니다:

**1. PHP는 개발을 위해 고안된 서버 측 스크립트 언어이자 수많은 웹 시스템의 기반이 되는 언어이다.**

- 맞습니다. PHP는 서버 측에서 동작하는 스크립트 언어로, 웹 개발에 널리 사용되며 WordPress 같은 많은 웹 시스템의 기반입니다.

**2. 파이썬은 가독성이 높고 다른 언어나 라이브러리에 접근하기 쉬운 언어이다.**

- 맞습니다. 파이썬은 간결하고 읽기 쉬운 문법으로 유명하며, 다양한 라이브러리와 프레임워크를 쉽게 활용할 수 있습니다.

**3. Java는 객체 지향 프로그래밍 언어이다.**

- 맞습니다. Java는 객체 지향 프로그래밍(OOP)을 지원하는 대표적인 언어로, 클래스와 객체를 기반으로 설계되었습니다.

**4. 파이썬은 빠른 속도를 요구하는 프로그램에 적합하다.**

- 틀렸습니다. 파이썬은 가독성과 개발 속도 면에서 강점이 있지만, 인터프리터 언어 특성상 실행 속도가 C나 Java 같은 컴파일 언어에 비해 느립니다. 따라서 빠른 속도를 요구하는 프로그램(예: 고성능 게임 엔진)에는 주로 적합하지 않고, 대신 데이터 분석이나 프로토타입 개발에 더 적합합니다.

16번 문제

▼ HTTP 요청 메서드 중 GET과 POST의 차이점에 대해 서술하시오.

정답: GET은 데이터를 조회할 때 간단하고 빠르게 사용되며, POST는 데이터를 생성하거나 수정할 때 보안성과 대용량 처리를 위해 사용됩니다. 상황에 따라 적절한 메서드를 선택하는 것이 중요합니다

## 17번 문제

▼ 웹 서버의 주요 기능 중 하나가 아닌 것을 고르시오.

1. HTTP 요청 처리
2. 정적 콘텐츠 제공
3. 데이터베이스 쿼리 최적화
4. 로드 밸런싱 지원

▼ 정답

3번

▼ 풀이

### 1. HTTP 요청 처리

- 맞습니다. 웹 서버의 핵심 기능 중 하나는 클라이언트로부터 HTTP 요청 (예: GET, POST)을 받아 처리하고 응답을 반환하는 것입니다.

### 2. 정적 콘텐츠 제공

- 맞습니다. 웹 서버는 HTML, CSS, 이미지 등의 정적 파일을 클라이언트에게 제공하는 역할을 합니다(예: Apache, Nginx).

### 3. 데이터베이스 쿼리 최적화

- 틀렸습니다. 데이터베이스 쿼리 최적화는 일반적으로 데이터베이스 관리 시스템(DBMS)이나 애플리케이션 서버의 역할입니다. 웹 서버는 주로 요청을 받아 처리하고 응답을 전달하며, 데이터베이스 내부 동작(쿼리 최적화 등)은 관여하지 않습니다.

### 4. 로드 밸런싱 지원

- 맞습니다. 일부 웹 서버(예: Nginx)는 로드 밸런싱 기능을 제공하여 트래픽을 여러 백엔드 서버로 분산할 수 있습니다. 이는 고가용성과 성능 향상에 기여합니다.

## 결론

웹 서버의 주요 기능은 HTTP 요청 처리, 정적 콘텐츠 제공, 로드 밸런싱과 같은 네트워크 및 콘텐츠 전달과 관련된 작업입니다. 반면, **데이터베이스 쿼리 최적화**는 웹 서버가 아닌 데이터베이스 시스템의 역할이므로 웹 서버의 주요 기능이 아닙니다.

## 18번 문제

▼ 다음 중 웹서버에 관한 내용 중 옳바르지 않은 것을 고르시오.

1. 웹서버 프로그램은 현재 크게 Apache, Nginx로 크게 양분하고 있다.
2. 최신 백엔드 프레임워크에는 웹서버가 내장되어 있다.(ex Node, Laravel, Flask)
3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache이다.
4. 순수 HTML, CSS, JavaScript로 작성된 코드는 VSCode의 Live Server 확장 프로그램을 통해 간단한 웹 서버로 실행되어 브라우저에서 그 결과를 확인할 수 있다.

▼ 정답

3번

▼ 풀이

**1. 웹서버 프로그램은 현재 크게 Apache, Nginx로 크게 양분하고 있다.**

- 맞습니다. Apache와 Nginx는 현재 가장 널리 사용되는 오픈소스 웹 서버 소프트웨어로, 시장을 양분하고 있다고 볼 수 있습니다. (IIS 같은 다른 서버도 있지만, 오픈소스 기준으로는 이 둘이 주도적입니다.)

**2. 최신 백엔드 프레임워크에는 웹서버가 내장되어 있다. (ex Node, Laravel, Flask)**

- 맞습니다.
  - Node.js는 자체적으로 HTTP 서버를 생성할 수 있는 기능을 제공합니다.
  - Laravel(PHP 기반)은 개발 환경에서 `php artisan serve` 명령으로 내장 웹 서버를 실행할 수 있습니다.
  - Flask(Python 기반)도 개발용 내장 웹 서버(Werkzeug)를 포함합니다.
- 이들은 주로 개발 환경에서 사용되며, 프로덕션 환경에서는 Apache나 Nginx 같은 별도의 웹 서버와 함께 배포되는 경우가 많지만, "내장되어 있다"는 설명은 틀리지 않습니다.

**3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache이다.**

- 틀렸습니다. Apache는 1995년에 처음 개발되었고, Nginx는 그보다 훨씬 나중에 2004년에 등장했습니다. Nginx는 Apache의 성능 문제(특히 동시 연결 처리의 비효율성)를 개선하기 위해 설계된 웹 서버입니다. 따라서 시간 순서와 개발 목적상 "Nginx의 문제점을 해결해서 Apache를 만들었다"는 설명은 사실과 반대입니다.

4. 순수 HTML, CSS, JavaScript로 작성된 코드는 VSCode의 Live Server 확장 프로그램을 통해 간단한 웹 서버로 실행되어 브라우저에서 그 결과를 확인할 수 있다.

- 맞습니다. VSCode의 Live Server 확장 프로그램은 정적 파일(HTML, CSS, JS)을 로컬 웹 서버로 실행해 실시간으로 브라우저에서 결과를 확인할 수 있게 해줍니다.

## 결론

올바르지 않은 것은 3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache입니다.

20번 문제

▼ 세션(Session)과 쿠키(Cookie)의 차이점에 대해 서술하시오.

정답:

**쿠키**는 클라이언트에 저장되며 간단한 데이터를 다루고, 모든 요청에 자동 전송됩니다.

**세션**은 서버에 데이터를 저장하며, 세션 ID를 통해 상태를 관리하고 보안성과 유연성이 높습니다.

두 기술은 보완적으로 사용되며, 예를 들어 세션 ID를 쿠키로 전달하는 방식이 일반적입니다. 상황에 따라 적절히 선택하거나 조합해 사용하는 것이 중요합니다.

21번문제

▼ [.gitignore] 파일이 하는 역할을 서술하시오.

정답: .gitignore 파일은 Git이 특정 파일이나 디렉토리를 버전 관리에서 제외하도록 지시하는 설정 파일로, 프로젝트의 효율성과 보안을 높이는 데 중요한 역할을 합니다. 이를 통해 개발자는 핵심 소스 코드만 관리하고 불필요한 요소를 배제할 수 있습니다.

24번 문제

▼ "Pull Request"에 대해 올바른 것을 고르시오

1. 내 저장소의 변경 사항을 다른 사람의 저장소에 요청하는 것
2. 다른사람의 저장소를 내 저장소로 합치는 것
3. 새로운 저장소를 만들 때 사용하는 옵션
4. 파일을 삭제하는것



▼ 정답

1번

▼ 풀이

Pull Request는 협업 시 내 저장소의 변경 사항을 다른 저장소(주로 원본 저장소)에 병합하도록 요청하는 프로세스입니다. 따라서 **올바른 것은 1. 내 저장소의 변경 사항을 다른 사람의 저장소에 요청하는 것입니다.**

25번 문제

▼ "git stash push" 명령어의 기본 기능을 고르시오

▼ 정답

▼ 풀이

▼

16번 문제

▼ HTTP 요청 메서드 중 GET 과 POST 차이점에 대해 서술하시오.

▼ 정답

GET은 데이터를 조회하는 데 적합하고 간단하며 가시적인 반면, POST는 데이터를 생성하거나 수정하는 데 사용되며 더 안전하고 유연합니다.

17번 문제

▼ 웹 서버의 주요 기능 중 하나가 아닌 것을 고르시오.

1. HTTP 요청 처리
2. 정적 콘텐츠 제공
3. 데이터베이스 쿼리 최적화
4. 로드 밸런싱 지원

▼ 정답

3번

▼ 풀이

**HTTP 요청 처리:** 웹 서버의 핵심 기능 중 하나입니다. 클라이언트로부터 HTTP 요청을 받아 처리하고 응답을 반환합니다.**정적 콘텐츠 제공:** HTML, CSS, 이미지 등의 정적 파일을 제공하는 것도 웹 서버의 기본 역할입니다.**데이터베이스 쿼리 최적화:** 이는 주로 데이터베이스 관리 시스템(DBMS)이나 애플리케이션 서버의 역할입니다. 웹 서버는 일반적으로 데이터베이스와 상호작용할 수 있지만, 쿼리 최적화는 웹 서버의 주요 기능이 아닙니다.**로드 밸런싱 지원:** 일부 웹 서버(예: Nginx, Apache)는 로드 밸런싱 기능을 지원하거나 관련 모듈을 통해 이를 수행할 수 있어 주요 기능으로 간주될 수 있습니다.

- **HTTP 요청 처리:** 웹 서버의 핵심 기능 중 하나입니다. 클라이언트로부터 HTTP 요청을 받아 처리하고 응답을 반환합니다.
- **정적 콘텐츠 제공:** HTML, CSS, 이미지 등의 정적 파일을 제공하는 것도 웹 서버의 기본 역할입니다.
- **데이터베이스 쿼리 최적화:** 이는 주로 데이터베이스 관리 시스템(DBMS)이나 애플리케이션 서버의 역할입니다. 웹 서버는 일반적으로 데이터베이스와 상호작용할 수 있지만, 쿼리 최적화는 웹 서버의 주요 기능이 아닙니다.
- **로드 밸런싱 지원:** 일부 웹 서버(예: Nginx, Apache)는 로드 밸런싱 기능을 지원하거나 관련 모듈을 통해 이를 수행할 수 있어 주요 기능으로 간주될 수 있습니다.

#### 18번 문제

▼ 다음 중 웹서버에 관한 내용 중 옳바르지 않은 것을 고르시오.

1. 웹서버 프로그램은 현재 Apache, Nginx로 크게 양분하고 있다.
2. 최신 백엔드 프레임 워크에는 웹서버가 내장되어 있다.(ex. Node, Laravel, Flask)
3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache이다.
4. 순수 HTML, CSS, JavaScript로 작성된 코드는 VSCode Live Server 확장 프로그램을 통해 간단한 웹 서버로 실행되어, 브라우저에서 그 결과를 확인할 수 있다.

▼ 정답

3번

▼ 풀이

1. 웹서버 프로그램은 현재 Apache, Nginx로 크게 양분하고 있다.
  - 사실입니다. Apache와 Nginx는 현재 가장 널리 사용되는 웹 서버 소프트웨어로, 시장을 양분하고 있다고 볼 수 있습니다.
  - 올바른 설명.

**2. 최신 백엔드 프레임워크에는 웹서버가 내장되어 있다. (ex. Node, Laravel, Flask)**

- 부분적으로 맞습니다. Node.js는 자체적으로 HTTP 서버를 포함하고 있으며, Flask도 개발용 웹 서버(Werkzeug)를 내장하고 있습니다. 하지만 Laravel은 기본적으로 PHP로 동작하며, 내장 웹 서버(Artisan serve)를 개발용으로 제공하지만, 프로덕션 환경에서는 Apache나 Nginx 같은 별도의 웹 서버가 필요합니다. "최신 백엔드 프레임워크"라는 일반화된 표현은 대체로 맞지만, Laravel의 경우는 예외가 있으므로 약간 모호할 수 있습니다.

- 대체로 올바른 설명.

**3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache이다.**

- 이는 틀렸습니다. Apache는 1995년에 처음 개발되었고, Nginx는 2004년에 등장했습니다. 즉, Apache가 Nginx보다 훨씬 먼저 만들어졌으므로 Nginx의 문제점을 해결하기 위해 Apache가 만들어졌다는 주장은 시간적으로 맞지 않습니다. 오히려 Nginx는 Apache의 성능 문제(특히 동시 연결 처리)를 개선하기 위해 설계된 경우가 더 타당합니다.

- 올바르지 않은 설명.

**4. 순수 HTML, CSS, JavaScript로 작성된 코드는 VSCode Live Server 확장 프로그램을 통해 간단한 웹 서버로 실행되어, 브라우저에서 그 결과를 확인할 수 있다.**

- 맞습니다. VSCode의 Live Server 확장 프로그램은 정적 파일(HTML, CSS, JS)을 제공하는 간단한 웹 서버를 로컬에서 실행하여 실시간으로 결과를 확인할 수 있게 해줍니다.

- 올바른 설명.

## 결론

**3. Nginx의 문제점을 해결해서 만든 웹서버가 Apache이다**가 올바르지 않은 설명입니다.

20번 문제

▼ 세션(Session)과 쿠키(Cookie)의 차이점에 대해 서술하시오.

▼ 정답

- **저장 위치:** 쿠키는 클라이언트(브라우저)에, 세션은 서버에 저장.
- **보안:** 쿠키는 클라이언트에서 접근 가능해 덜 안전, 세션은 서버 저장으로 더 안전.

- **용량:** 쿠키는 4KB 제한, 세션은 서버 설정에 따라 더 큰 용량 가능.
- **유효 기간:** 쿠키는 만료 날짜 설정 가능, 세션은 기본적으로 브라우저 종료 시 삭제.
- **데이터 전송:** 쿠키는 매 요청마다 전송, 세션은 ID만 전송.
- **사용 예:** 쿠키는 설정 저장(언어 등), 세션은 로그인 상태 유지 등.

## 21번 문제

▼ [.gitignore] 파일이 하는 역할을 서술하시오.

▼ 정답

Git 버전 관리 시스템에서 특정 파일이나 디렉토리를 추적(버전 관리)하지 않도록 지정하는 역할을 합니다. 이를 통해 불필요하거나 민감한 파일이 Git 저장소에 커밋되는 것을 방지할 수 있습니다.

## 24번 문제

▼ "Pull Request"에 대해 올바른 것을 고르시오.

1. 내 저장소의 변경 사항을 다른 사람의 저장소에 요청하는 것
2. 다른 사람의 저장소를 내 저장소로 합치는 것
3. 새로운 저장소를 만들 때 사용하는 옵션
4. 파일을 삭제하는 것

▼ 정답

1번

▼ 풀이

"Pull Request"에 대한 설명을 분석하여 올바른 것을 찾아보겠습니다.

### 1. 내 저장소의 변경 사항을 다른 사람의 저장소에 요청하는 것

- 맞습니다. Pull Request는 주로 내 저장소(또는 포크한 저장소)에서 작업한 변경 사항을 원본 저장소(다른 사람 또는 팀의 저장소)에 병합해 달라고 요청하는 프로세스입니다. 예를 들어, GitHub에서 내 브랜치의 코드를 원본 저장소의 메인 브랜치로 통합하기 위해 사용됩니다.

- **올바른 설명.**

### 2. 다른 사람의 저장소를 내 저장소로 합치는 것

- 틀렸습니다. 이는 "Pull" 또는 "Merge"에 가까운 개념입니다. Pull Request는 반대로 내가 작업한 내용을 다른 저장소에 제안하는 것이지, 다른 저장소를 내 저장소로 가져오는 것이 아닙니다.

### 3. 새로운 저장소를 만들 때 사용하는 옵션

- 틀렸습니다. Pull Request는 저장소를 생성하는 것과 무관하며, 기존 저장소 간의 변경 사항을 통합하기 위한 요청입니다.

### 4. 파일을 삭제하는 것

- 틀렸습니다. Pull Request는 파일 삭제와 직접적인 관련이 없으며, 변경 사항 (추가, 수정, 삭제 등)을 제안하는 협업 도구입니다.

## 결론

1. 내 저장소의 변경 사항을 다른 사람의 저장소에 요청하는 것이 "Pull Request"에 대한 올바른 설명입니다.

25번 문제

▼ "git stash push" 명령어의 기본 기능을 고르시오.

1. 커밋된 변경 사항을 원격 저장소로 저장한다.
2. 커밋되지 않은 변경 사항을 임시로 저장한다.
3. 새 브랜치를 생성하고 변경 사항을 브랜치에 적용시킨다.
4. 현재 브랜치의 커밋 히스토리를 표시한다.

▼ 정답

2번

▼ 풀이

1. 커밋된 변경 사항을 원격 저장소로 저장한다.

- 틀렸습니다. "git stash push"는 커밋된 변경 사항과 관련이 없으며, 원격 저장소로 데이터를 보내는 것도 아닙니다. 원격 저장소로의 전송은 "git push" 명령어의 역할입니다.

2. 커밋되지 않은 변경 사항을 임시로 저장한다.

- 맞습니다. "git stash push"는 현재 작업 디렉토리와 인덱스에 있는 커밋되지 않은 변경 사항(예: 수정된 파일, 스테이징된 파일)을 임시 저장소(stash)에 저장하고, 작업 디렉토리를 깨끗한 상태로 되돌립니다. 이를 통해 다른 작업을 수행하거나 브랜치를 전환할 수 있습니다.

- 올바른 설명.

3. 새 브랜치를 생성하고 변경 사항을 브랜치에 적용시킨다.

- 틀렸습니다. "git stash push"는 브랜치를 생성하지 않으며, 변경 사항을 브랜치에 적용시키는 대신 임시로 저장만 합니다. 브랜치 생성은 "git branch"나 "git checkout -b"로 수행됩니다.

#### 4. 현재 브랜치의 커밋 히스토리를 표시한다.

- 틀렸습니다. 커밋 히스토리를 확인하려면 "git log" 명령어를 사용해야 하며, "git stash push"는 히스토리 표시와 무관합니다.

## 결론

2. 커밋되지 않은 변경 사항을 임시로 저장한다가 "git stash push" 명령어의 기본 기능에 대한 올바른 설명입니다.

(참고: "git stash"만 입력해도 기본적으로 "git stash push"와 동일하게 동작합니다.)

#### 26번 문제

- ▼ git init 명령어 실행 시, 어떤 폴더가 생기는지 적으시오.

명령어를 실행하면 현재 디렉토리에 **.git**이라는 숨김 폴더가 생성됩니다. 이 폴더는 Git 저장소의 모든 메타데이터와 버전 관리 정보를 저장하는 핵심 디렉토리입니다.

#### 27번 문제

- ▼ main과 side라는 두개의 Branch가 있다고 가정할 때, 이를 병합하는 명령어를 적으시오. (사용자는 side 브랜치를 사용 중임)

- ▼ 정답

git merge main