[교안] SK, "개발자의 글쓰기" (6/17)

<목차>

개요

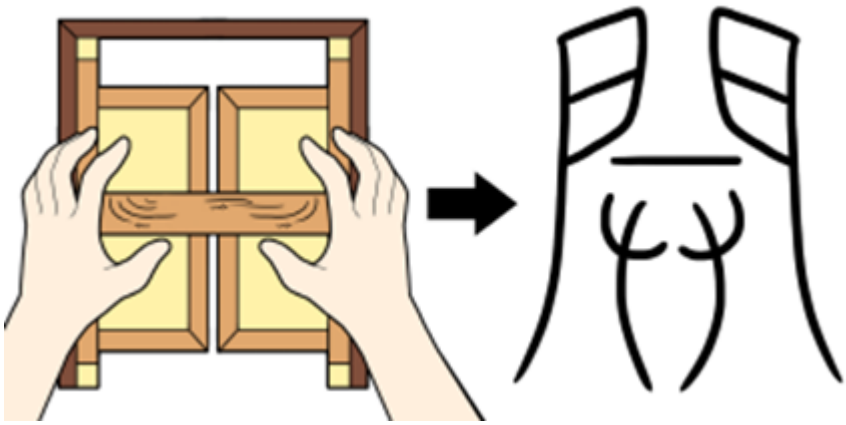
- 교육명: 개발자의 글쓰기
- 강사: 김철수 *소개 바로가기
- 목표: 개발자들이 쉽고 빠르게 글을 쓸 수 있는 방법을 익힌다.
- 참고서적: 개발자의 글쓰기

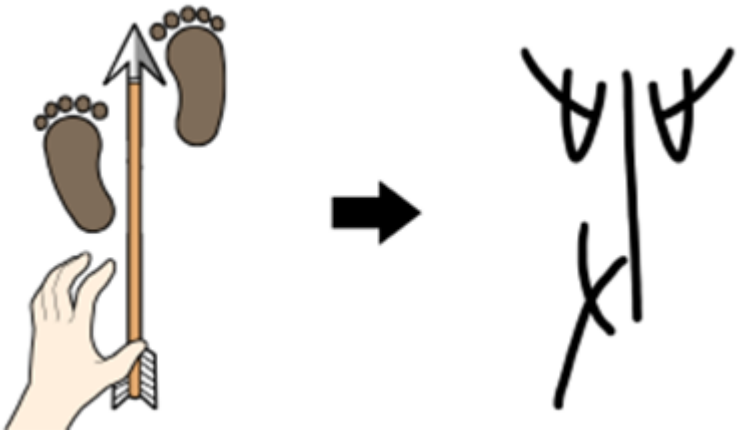
---

1. 개발자는 원래 글로 소통하는 사람

▼ DEVELOPER의 뜻

▼ 開發의 뜻





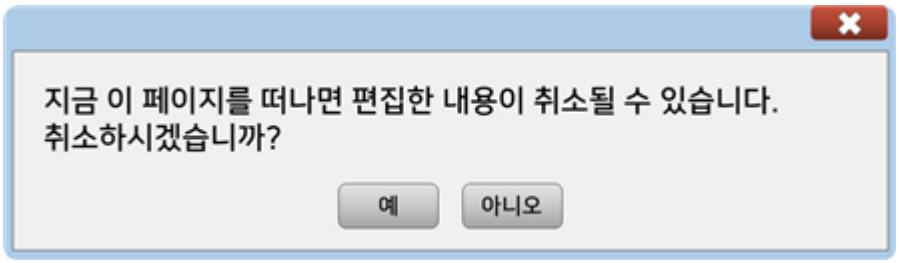
2. 개발자의 글쓰기 원칙: 정확성, 간결성, 가독성

▼ 비즈니스 글쓰기의 특징

- 논리, 설득, 실행

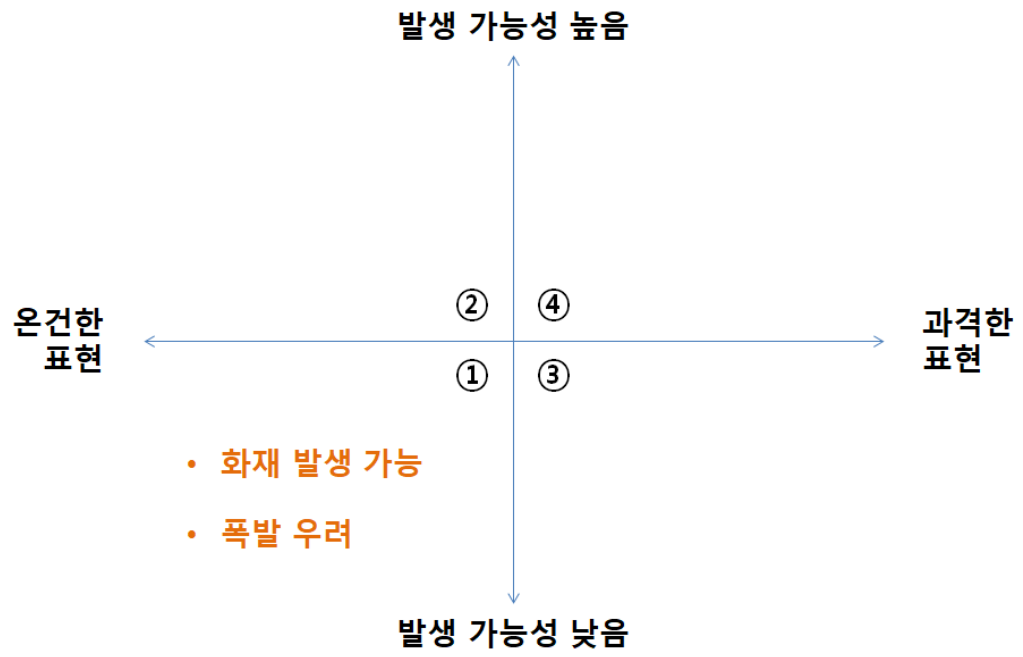
▼ 단어를 정확하게 쓰기

▼ 취소해? 말아?

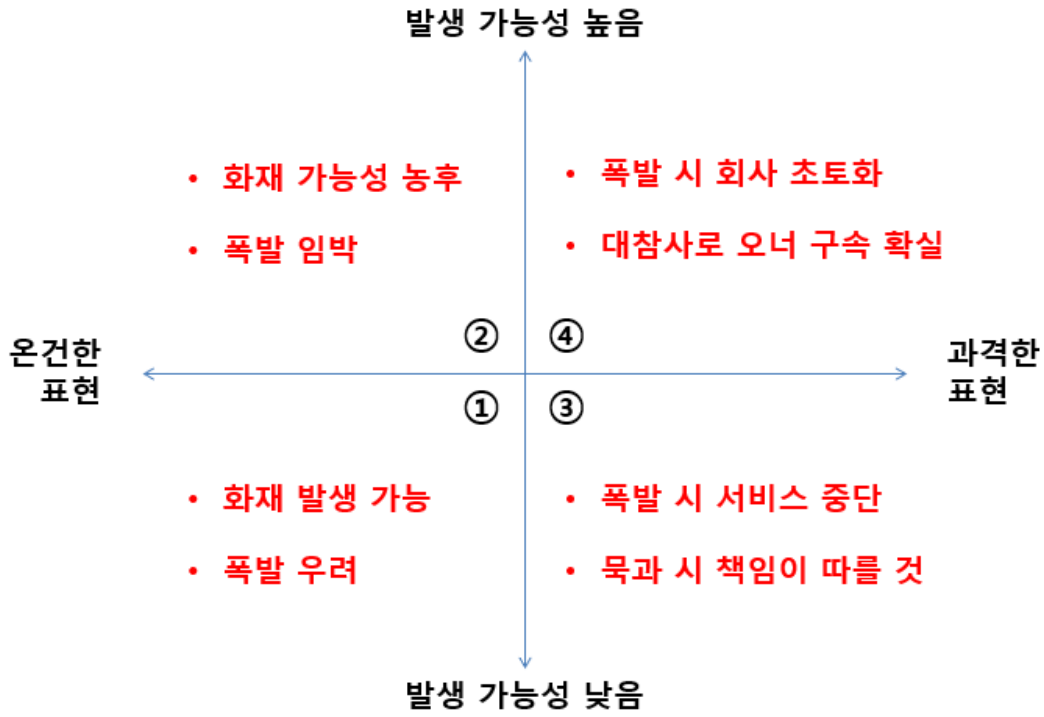


→ 정말 떠나시겠습니까? 예/아니오
→ 이 페이지를 떠나면 편집한 내용이 사라집니다. [떠나겠다] [안 떠나겠다]
- [내용유지/사라짐]

▼ K상무의 고민

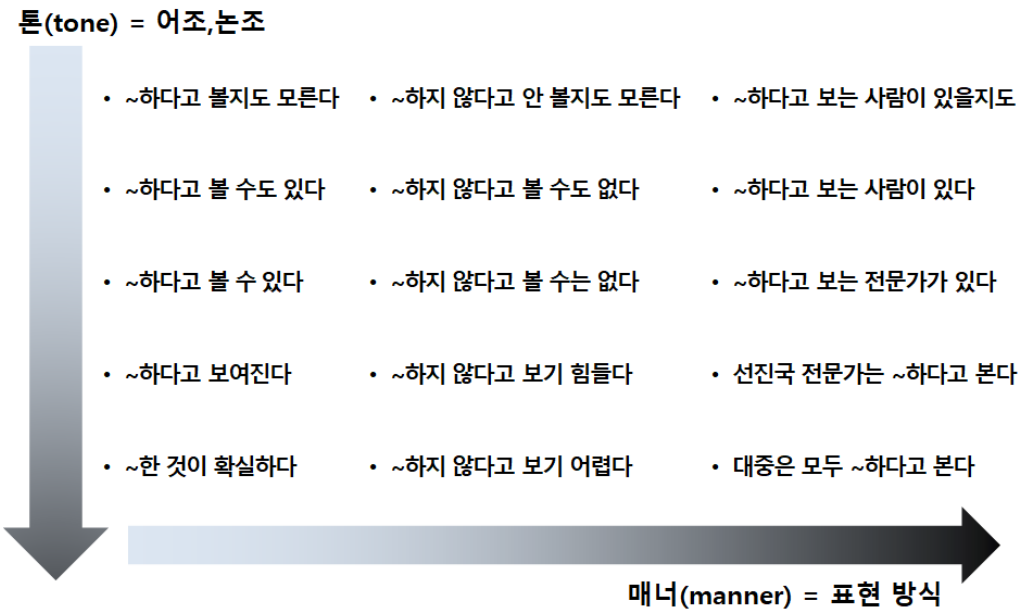


▼ K상무의 고민 해결



▼ Tone & Manner

▼ 톤과 매너란?



▼ 예문

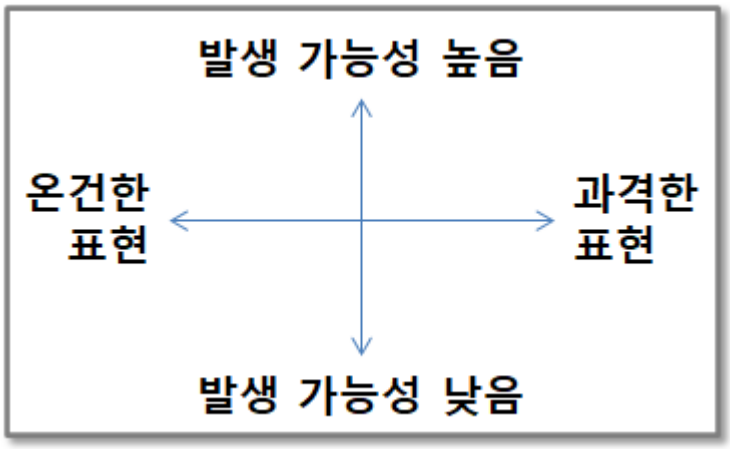
☑ 시스템 구축 방안으로 '자체 개발'**이 적절함**

- 100% → 긴요, 절박하다, 국룰이다.
- 90% → 불가피하다, 다른 대안 없다.
- 70% → 필수다
- 60% → 적절하다
- 50% →
- 40% →
- 30% → 필요하다
- 20% →
- 10% → 요구된다.
- 0% →

▼ 정확한 단어 찾는 방법

1. 포털사이트 어학사전에서 비슷한 의미나 의도를 가진 단어를 찾는다.

2. 1차원(X)이나 2차원(XY) 기준으로 단어를 배치하고 표현 방법을 달리해 본다.
3. 보고서 결론에 맞게 적절한 단어를 선택하여 쓴다.



▼ 문장을 간결하게 쓰기

▼ 군더더기 줄이기

空山木落雨蕭蕭
공 산 목 락 우 소 소

> 텅 빈 산에 나뭇잎은 떨어지고 비는 부슬부슬 내리는데 → 22글자

- [x] 빈 산 잎 지고 비는 부슬부슬 → 11글자

▼ 개조식으로 요약해서 쓰기

▼ 서술식

옛날에 착한 나무꾼이 살았어요. 나무꾼은 늙으신 부모님을 극진히 모시는 효자였지만, 무척 가난했어요. 나무꾼은 매일 산으로 나무를 하러 갔어요. 그러던 어느 날이었어요. 열심히 도끼질을 하던 나무꾼은 그만 실수로 낡은 도끼를 연못에 빠뜨리고 말았어요.
"아이고 내 도끼! 저 도끼마저 없으면 앞으로 어떻게 나무를 한단 말인가? 엉엉“ 나무꾼은 잃어버린 도끼를 생각하며 땅바닥에 주저 앉아 큰 소리로 슬피 울었어요. 그 때였어요. 연못 속에서 하얀 옷을 입은 산신령이 연기처럼 펑 나타났어요.
"너는 왜 그리 슬퍼하고 있느냐?“ "산신령, 저는 가난한 나무꾼인데 나무를 하다가 그만 도끼를 연못에 빠뜨리고 말았어요. 그 도끼가 없으면 저는 앞으로 나무를 할 수 없고 부모님을 모실 수가 없습니다.“ "음. . . 내가 그 도끼를 찾아주마.“
산신령은 다시 연못 속으로 사라졌다가 곧 도끼 하나를 들고 다시 나타났어요. "이 도끼가 네 도끼냐?“ 산신령이 들고 있는 것은 광채가 번쩍이는 은도끼였어요. "아닙니다.“ "오, 그래? 잠시만 기다리거라.“ 산신령은 다시 연못 속으로 사라졌다가 다른 도끼를 들고 나타났어요. 이번에는 더욱 광채가 찬란한 금도끼였어요. "이 도끼가 네 도끼냐?“ "아, 아닙니다, 제 도끼는 아주 낡은 쇠도끼입니다.“ 산신령은 고개를 끄덕이며 다시 도끼를 들고 연못 속으로 사라졌어요.
곧 다시 나타난 산신령의 손에는 나무꾼의 낡은 쇠도끼가 들려 있었어요. "그럼 이 낡은 도끼가 니 도끼냐?“ "네, 그게 바로 제 도끼입니다.“
나무꾼은 도끼를 되찾게 되어 너무나 기뻐하며 산신령께 꾸벅 절을 했어요. "허허, 참으로 정직한 나무꾼이로구나. 너의 정직함과 효성에 대한 상으로 이 세 개의 도끼를 모두 너에게 주마.“ 나무꾼은 뜻밖의 상을 받고 좋아서 어쩔 줄 모르며 집으로 돌아갔어요. 부모님들도 무척 기뻐하셨어요. 가난한 나무꾼은 정직한 마음 덕분에 부자가 되어 부모님과 행복하게 살게 되었어요.

▼ 개조식

- 나무꾼, 연못에 도끼 유실
- 산신령, 금은쇠도끼 제시
- 나무꾼, 쇠도끼만 수용
- 산신령, 금은도끼 포상

▼ S사 팀장의 간결성

범례: G: 산신령, W: 나무꾼

- W. 연못에 도끼 유실
- G, 금은쇠도끼 제시
- W, 쇠도끼만 수용
- G, 금은도끼 포상

▼ 예문

- "Docker, 도커의 이미지를 이용해서 컨테이너를 실행 및 중지해보자."
- https://devocean.sk.com/blog/techBoardDetail.do?ID=163125

> ☑ **블로그 목적: Docker 이미지로 컨테이너 작동 방법 공유**
>
> **우선, Docker에서 hello-world를 실행해보자.**
> 잠깐, 여기서 도커를 사용한다는 것은 무엇일까?
> 결국 이미지를 이용해서 컨테이너를 생성하고 실행한다는 것을 말함.
> 그리고, 컨테이너를 실행하려면, docker run 명령어를 사용하면 된다.
>
> **그런데, 컨테이너란 무엇일까?**
> 컨테이너는 앱 계층의 추상화를 위해서 코드와 종속성을 함께 패키징하는 것을 의미한다.
> 다시말해서, 여러 컨테이너가 동일한 머신에서 실행되고 다른 컨테이너와 OS 커널을 공유 할 수 있게되는데...
> 각각 사용자 공간에서 격리 된 프로세스로 실행되고, 컨테이너는 VM보다 적은 공간을 차지함.
> 컨테이너의 자세한 설명은 아래 웹페이지를 참고부탁드립니다.
> https://www.docker.com/resources/what-container

▼ 문단을 읽기 쉽게 쓰기

- 예문: "AWS Private NAT Gateway - Part 1"
- https://devocean.sk.com/blog/techBoardDetail.do?ID=163185

Private NAT Gateway 아키텍처 구성

<요약>
*** 사용자 VPC가 2개 서브넷 대역 존재**
*** Sub1은 연동이 안되는 대역 → IP 대역 사용 가능**
*** Sub2는 연동이 되는 대역 → Unique한 대역 사용**
*** Private NAT Gateway는 다른 VPC와 연동이 가능한 서브넷에 위치**

<상세>

사용자 VPC가 2개 서브넷 대역 존재
ZIGI-VPC1과 ZIGI-VPC2 라는 2개의 사용자 VPC가 있습니다. 2개의 사용자 VPC는 Transit Gateway를 각각 연결(Attached)되어 있습니다. 각 사용자 VPC에는 Sub1과 Sub2라는 2개의 서브넷 대역이 존재합니다.

Sub1은 연동이 안되는 대역 → IP 대역 사용 가능
Sub1은 Transit Gateway를 통해서 VPC 간에 연동이 되지 않는 대역입니다. 연동을 하지 않는 대역이기 때문에 VPC1과 VPC2에서 동일한 IP 대역 사용이 가능합니다. 물론 IP 설계를 하는 데 있어서, IP 대역에 대한 부족이 없다면 Unique한 대역을 사용하셔도 무방합니다.

Sub2는 연동이 되는 대역 → Unique한 대역 사용
Sub2는 Transit Gateway를 통해서 VPC 간에 연동을 하는 대역입니다. Transit Gateway에서 라우팅 처리를 통해서 연동을 해야하기 때문에 각 VPC 간의 Unique한 대역을 사용합니다. ZIGI-VPC1에는 Sub1 서브넷에는 Private NAT Gateway를 사용 할 가상 서버 1대가 위치합니다. Sub2 서브넷에는 오늘 구성의 핵심인 Private NAT Gateway가 위치합니다.

Private NAT Gateway는 다른 VPC와 연동이 가능한 서브넷에 위치
기존의 사설망에서 공인망으로 NAT하기 위해서 인터넷과 연동이 가능한 공인 서브넷에 NAT Gateway가 위치한 것처럼 Private NAT Gateway는 다른 VPC와 연동이 가능한 서브넷에 위치하게 됩니다. 테스트를 위해서 ZIGI-VPC2에는 ZIGI-VPC1의 가상 서버가 통신할 수 있는 가상 서버 1대가 위치합니다.
...

3. 기술 블로그, 쉽게 쉽게 쓰자!

▼ "그것이 알고 싶다 – 왜 개발자는 글을 못 쓸까?"

- https://engineering.linecorp.com/ko/blog/why-are-engineers-so-bad-at-writing/

▼ 김철수 소장의 해법

- 글쓰기가 어려운 이유는 잘 쓰려고 하니까!

▼ 주제 우선 글쓰기 → 소재 우선 글쓰기

구분	내용	종류
저著	직접 경험하고 실험한 과정이나 결과	개발기, 도입기, 적용기
술述	어떤 것을 분석하여 의미를 풀이하고 해석한 것	기술 소개, 용어 분석, 에러 해결 방법 등
편編	산만하고 복잡한 자료를 편집해 질서를 부여한 것	프로그램 설치/설정 방법, 튜토리얼, 세미나 후기, 책 리뷰
집輯	여러 사람의 견해나 흩어진 자료를 한데 모아 정리한 것	명령어 모음, 팁, OO가지 규칙

▼ 독자 중심 글쓰기 → 자기 수준으로 글쓰기

- 8살 조카에게 데이터베이스를 3문장으로 설명하기
- 또 다른 나에게 알려주기

▼ 예문: LINE 메신저 앱에 온 디바이스 머신 러닝 적용하기

- https://engineering.linecorp.com/ko/blog/on-device-machine-learning-line-app/

> ☑ 컨테이너는 앱 계층의 추상화를 위해서 코드와 종속성을 함께 패키징하는 것을 의미한다.
> 다시말해서, 여러 컨테이너가 동일한 머신에서 실행되고 다른 컨테이너와 OS 커널을 공유 할 수 있게되는데...
> 각각 사용자 공간에서 격리 된 프로세스로 실행되고, 컨테이너는 VM보다 적은 공간을 차지함.
> **컨테이너의 자세한 설명은 아래 웹페이지를 참고부탁드립니다.**
> **https://www.docker.com/resources/what-container**

▼ 주장하는 글쓰기 → 재미있게 글쓰기

▼ 1인칭 서술 → 2인칭 대화 → 3인칭 묘사

- 강남언니 1~2인칭: https://blog.gangnamunni.com/post/city-squad/
- 예문: "Docker build 가볍게 생성하기"
- https://devocean.sk.com/blog/techBoardDetail.do?ID=162976

☑ 1인칭 서술
Docker는 컨테이너 툴은 현재 가장 많이 사용되고 있는 도구이다.
컨테이너 이미지를 작성할때, 어떻게 작성하느냐에 따라서 컨테이너 빌드 속도와, 빌드 후 이미지 크기에 영향을 주며, 이러한 영향은 프로젝트 개발 라이프 사이클에서 어느정도 영향을 주게 된다.
필자의 경우 Docker Build 시 Dockerfile 을 비효율적으로 작성하는 바람에 빌드 시간이 30분이 소요된 경우도 있었다.
이런 이유는 Docker Build 레이어를 사용한다는 사실을 모른채 Dockerfile을 작성했었고, 매번 빌드시마다 의존성 파일을 다운로드하는 비효율적인 빌드를 수행했었기 때문에 발생했었다.
이번 아티클은 Golang 으로 웹 어플리케이션을 간단히 작성해보고, Docker 이미지 빌드를 수행하는 방법에 대해서 알아볼 것이다.

☑ 2인칭 대화

☑ 3인칭 묘사

▼ 스토리 + 에피소드

- 에피소드를 박스 기사(코너 속의 코너)로 구성
- 에피소드 종류: 사건사고나 인터뷰
- 강남언니: https://blog.gangnamunni.com/post/docker-compose-for-local-env

▼ 과장

겸손 그 자체

Docker는 컨테이너 툴은 현재 가장 많이 사용되고 있는 도구이다.
컨테이너 이미지를 작성할때, 어떻게 작성하느냐에 따라서 컨테이너 빌드 속도와, 빌드 후 이미지 크기에 영향을 주며, 이러한 영향은 프로젝트 개발 라이프 사이클에서 어느정도 영향을 주게 된다.
필자의 경우 Docker Build 시 Dockerfile 을 비효율적으로 작성하는 바람에 빌드 시간이 30분이 소요된 경우도 있었다.
이런 이유는 Docker Build 레이어를 사용한다는 사실을 모른채 Dockerfile을 작성했었고, 매번 빌드시마다 의존성 파일을 다운로드하는 비효율적인 빌드를 수행했었기 때문에 발생했었다.
이번 아티클은 Golang 으로 웹 어플리케이션을 간단히 작성해보고, Docker 이미지 빌드를 수행하는 방법에 대해서 알아볼 것이다.

끝.