

2024년 상반기 K-디지털 트레이닝

# JS2 기본 + 팀별 활동

[KB] IT's Your Life



## 팀별 활동



응용/활용 능력 함양

## study형 과제



문제에 대한 심화학습

## 문제해결형 과제



기본기 탄탄 내재화



- 실무 관점에서의 자바 개발자 수준 기준

- 상/최상 커트라인 주요 항목

- 아래 문장의 내용을 이해하고 이를 해결할 수 있는 지식과 노하우를 가지고 있는가?
- 스프링 내부 프레임워크의 각 컴포넌트를 프로그래밍 할 수 있는가?
  - 컨트롤러, 서비스, DBCP 등을 구현하여 서버 내부 프레임워크의 주요 컴포넌트를 만들 수 있는가?
  - 리플렉션에 대한 이해가 있는가?
- 디자인 패턴에 따라 팩토리, 프록시, 프로토타입, 싱글톤 대상이 되는 컴포넌트를 분리/판별하여 설계할 수 있고, 프로젝트에 이를 구현하여 넣을 수 있는가?
- 네트워크의 주요 레이어(OSI 7계층, TCP/IP 4계층)를 이해하고 이를 웹 개발시 적용할 수 있는가?
- 서버/클라이언트/DB의 성능을 체크하여 개선안을 제시할 수 있는가?
- 서버/DB의 로그 처리할 대상을 선별하고 이를 로그로 저장하여 분석하는 방법을 알고 실행할 수 있는가?

- 실무 관점에서의 자바 개발자 수준 기준

- 중급 커트라인 주요 항목

- http모듈을 이용하여 다양한 Open API를 스프링 프로젝트에 적용하고, 만약 데이터를 json으로 가지고 온 후, 실시간 전송 및 db저장 여부를 판단할 수 있는가?
- 규칙적인 data관리시 자바로 해줄 수 있는 장치가 무엇이 있는가? (예, 스프링 스케줄러)
- spring으로 구축된 자바 서버에 다른 목적을 가진 AI서버를 구축하여 서비스를 융합할 수 있는가?(AI추천서비스, AI chatbot등)
- spring으로 구축된 자바 서버에 파이썬이나 다른 언어로 만들어진 서비스를 융합할 수 있는가? 방법은 무엇인가? 설계는 가능한가?
- 자동화 스크립트를 이해하고 이를 업무에 활용할 수 있는가? 간단한 자동화 스크립트를 생성하여 이를 팀에게 공유하고 설명할 수 있는가?

면접시 심화 질문 내용

- 실무 관점에서의 자바 개발자 수준 기준

- 기본 커트라인

- 자바의 추상클래스, 인터페이스를 차이를 구분하고 다형성에 따라 객체를 설계하여 구현할 수 있는가?
- 스프링을 이용하여 웹 사이트를 구축할 수 있는가?
- 싱글톤과 프로토타입을 구분하여 구현하고 메모리 사용의 차이를 이해하고, 그림을 그려 설명할 수 있는가?
- 클래스내의 static과 instance 변수의 메모리 사용의 차이를 이해하고, 사용목적의 차이를 설명할 수 있는가?
- 서버에서 http관련 모듈을 이용하여 다양한 Open API를 활용할 수 있는가?
- 토큰을 발급받고 (또는 발급하고) 이를 웹 프로그램시 사용할 수 있는가? 스프링이나 장고에서의 토큰 발급 사용 policy는 정책이 어떻게 되는가?



**You**

일일이 하나씩 설정하지 않고 스크립트 파일을 만들어 설정을 자동화하는 것을 부르는 말



**ChatGPT**

그것은 "자동화 스크립트" 또는 "자동화 스크립팅"이라고 불립니다. 이는 컴퓨터 프로그램이나 시스템 설정을 자동으로 조작하고 제어하기 위해 작성된 스크립트 파일을 의미합니다. 일상적인 작업이나 반복적인 작업을 간소화하고 효율성을 높이기 위해 자동화 스크립트를 사용하는 것이 일반적입니다. 이러한 스크립트는 다양한 작업에서 사용되며, 네트워크 관리, 서버 관리, 소프트웨어 배포, 데이터 처리 등 다양한 영역에서 활용될 수 있습니다.

전사적 관점 관리  
→ 개인적 관리X

AWS의 자동화 스크립팅은 AWS 리소스를 관리하고 제어하기 위해 스크립트를 사용하는 것을 의미합니다. 주로 AWS CLI(Command Line Interface)나 AWS SDK(Software Development Kit)를 활용하여 자동화 스크립트를 작성하며, 이를 통해 다양한 AWS 서비스를 프로그래밍적으로 제어할 수 있습니다.

자동화 스크립팅을 통해 할 수 있는 일은 다음과 같습니다:

1. **인프라 관리:** EC2 인스턴스, S3 버킷, RDS 데이터베이스 등 AWS의 인프라 리소스를 프로그래밍적으로 생성, 수정, 삭제할 수 있습니다.
2. **자동화된 배포:** AWS 서비스를 사용하여 애플리케이션의 배포를 자동화할 수 있습니다. 예를 들어, AWS CodeDeploy를 사용하여 애플리케이션을 자동으로 배포하고 롤백하는 스크립트를 작성할 수 있습니다.
3. **모니터링 및 알림:** AWS CloudWatch와 같은 서비스를 사용하여 리소스의 상태를 모니터링하고, 일정 조건이 충족되면 알림을 받을 수 있도록 스크립트를 작성할 수 있습니다.
4. **비용 관리:** AWS 비용을 모니터링하고 예산을 관리하기 위해 자동화된 스크립트를 작성하여 비용을 최적화할 수 있습니다.



일반적으로 Spring 프로젝트의 자동화 스크립팅은 다음과 같은 작업을 포함할 수 있습니다:

1. **의존성 관리:** Maven이나 Gradle을 사용하여 프로젝트의 의존성을 관리합니다. 이를 통해 필요한 라이브러리를 자동으로 다운로드하고 프로젝트에 추가할 수 있습니다.
2. **빌드:** 프로젝트를 빌드하여 실행 가능한 JAR 파일이나 WAR 파일을 생성합니다. 이를 통해 개발자는 소스 코드의 변경 사항을 컴파일하고 패키징할 필요 없이 빌드 프로세스를 자동으로 실행할 수 있습니다.
3. **테스트:** JUnit이나 TestNG와 같은 테스트 프레임워크를 사용하여 단위 테스트, 통합 테스트 등을 자동으로 실행합니다. 이를 통해 소프트웨어의 품질을 유지하고 변경 사항이 예상대로 작동하는지 확인할 수 있습니다.
4. **배포:** 자동화 스크립트를 사용하여 빌드된 애플리케이션을 서버에 배포합니다. CI/CD 도구를 사용하여 자동으로 배포 파이프라인을 설정하고, 코드 변경 사항이 발생할 때마다 자동으로 배포 프로세스를 실행할 수 있습니다.
5. **환경 구성:** 자동화 스크립트를 사용하여 애플리케이션의 환경 설정을 자동으로 구성합니다. 이를 통해 개발, 테스트, 운영 환경 간의 일관성을 유지하고, 설정 오류를 방지할 수 있습니다.

Spring 프로젝트의 자동화 스크립트를 작성하면 개발 및 배포 프로세스를 효율적으로 관리할 수 있으며, 개발자들은 시간과 노력을 절약하고 소프트웨어의 품질을 향상시킬 수 있습니다.



# 팀별 Q1 - 제어문

- 1 별을 10개 찍어보세요. \*\*\*\*\*
- 2 원하는 기호를 넣으세요>> @  
원하는 횟수를 넣으세요>> 10  
@@@@@@@@@@@@ (10번 @를 반복)
- 3 1)증가, 2)감소, 3)종료를 선택해주세요.>> 1  
카운트: 1  
1)증가, 2)감소, 3)종료를 선택해주세요.>> 1  
카운트: 2  
1)증가, 2)감소, 3)종료를 선택해주세요.>> 2  
카운트: 1  
1)증가, 2)감소, 3)종료를 선택해주세요.>> 3  
시스템을 종료합니다. 현재까지의 카운트는 1

127.0.0.1:5500 내용:

증가 1), 감소 2) 종료 3)을 선택해주세요.

1

취소

127.0.0.1:5500 내용:

프로그램 종료&gt;&gt; 현재까지의 카운트는 4

확인

2024년 밀레니엄 세대시군요.  
5>> 열~~심히 공부하자.  
5>> 31일까지

- 4 Date를 이용하세요!  
1) 현재 년도를 구해서,  
2000년 이상이면 밀레니엄 세대시군요.  
아니면, 밀레니엄 세대가 아니시군요.
- 2) 요일을 구해서,  
토, 일은 쉬~~~~~자 프린트  
나머지 요일은 열~~심히 공부하자 프린트
- 3) 달을 구해서  
2월은 "28일까지" 프린트  
4, 6, 9, 11월은 "30일까지" 프린트  
나머지는 "31일까지" 프린트



- 아래와 같은 좌석이 있을 때 예매 상황을 처리하려고 한다.
  - 1차원 배열 저장 공간 생성
  - 좌석상황 브라우저에 프린트
  - prompt()를 이용하여 입력을 받아 예약 처리→입력 후 예약 상황 프린트, 한 좌석당 5000원 경우

금액 프린트

A	1	2	3	4		6	7		9	10	11	12	13	14
B	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C	1	2	3	4	5	6	7	8	9	10	11	12	13	14
D	1	2	3	4	5	6	7	8	9	10	11	12	13	14
E	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G	1	2	3	4	5	6	7	8	9	10	11	12	×	×
H	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I	1	2	3	4	5	6	7	8	9	10	11	12	13	14
J	1	2	3	4	5	6	7	8	9	10	11	12	13	14
K	1	2	3	4	5	6	7				11	12	13	14

예매 좌석 번호 입력

2

입력

- 현재 예약 번호: 2, 3, 4
- 결제금액: 15000원



127.0.0.1:5500 내용:  
좌석번호 선택(1~14), 종료 0

취소 확인

127.0.0.1:5500 내용:  
좌석번호 선택(1~14), 종료 0

취소 확인

127.0.0.1:5500 내용:  
이미 예약된 자리입니다.

확인

127.0.0.1:5500 내용:  
예약 시스템을 종료합니다.

확인

예약된 좌석번호는 1 2 3  
예매 금액은 15000원







- 분석하고자 하는 사이트 팀별 선정
- 팀 구성 인원: 4명
- 분석을 모두 마친 후, 제출
  - 제출 기한: 17시 40분 이전
  - 제출 가능 형태: ppt, pdf 등의 파일 또는 notion url등의 링크
  - slack으로 제출: 담임 주강사님 DM
  - 한 명이 대표로 파일 또는 링크를 DM으로 제출
  - 파일인 경우 파일명 예시: 1조-홍길동+김길동+송길동.pdf
  - 링크 제출시 팀원이름 함께 전송
  - 링크인 경우 링크 예시: <https://www.notion.so/4cb40997cf3147d3b90ac21284603c2e?pvs=4>



### 1) 기본 분석 항목

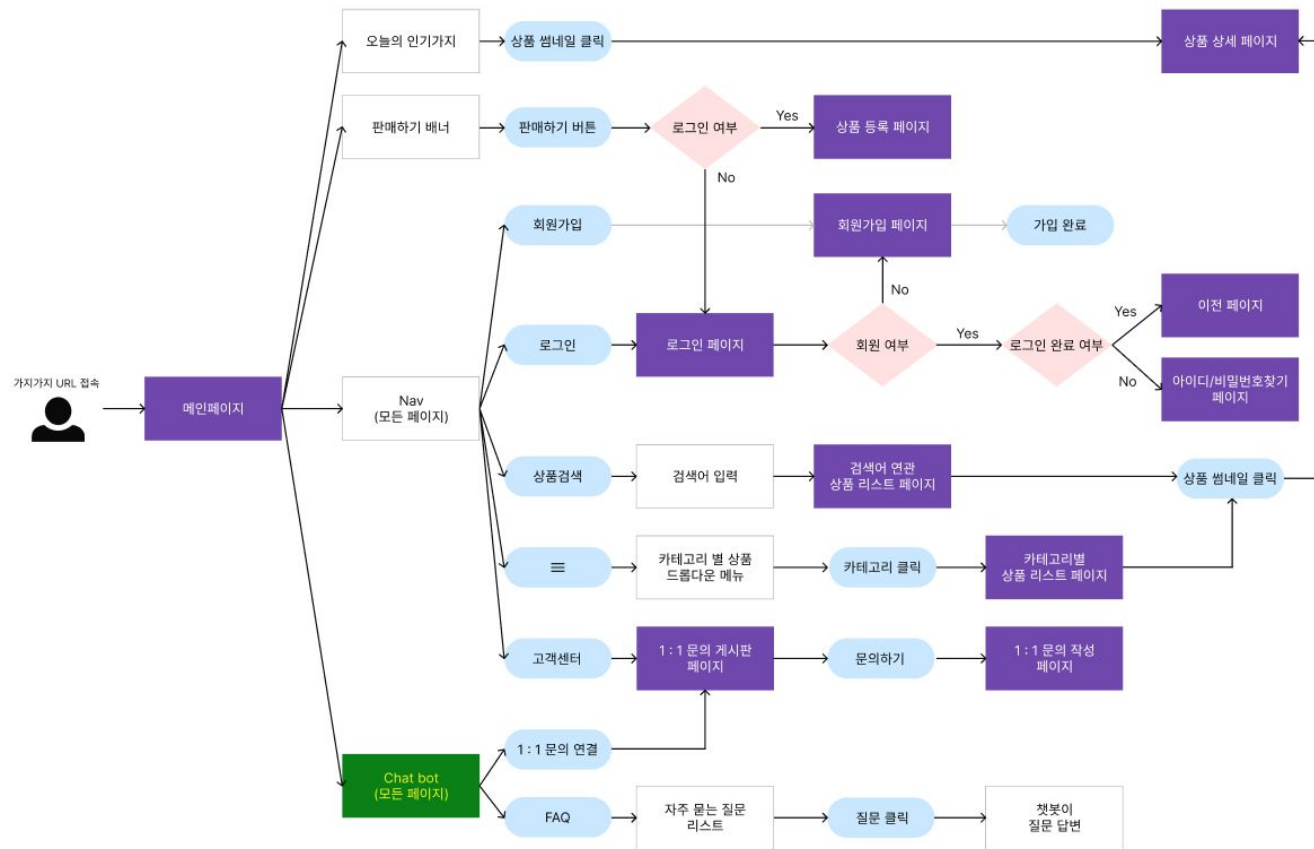
- 화면 및 코드 분석 : 메인 페이지의 기본 화면 분할 구성 분석
  - div, 시멘틱 태그 사용 여부
  - 화면 분할 포함 여부
  - DOM tree 추출
  - css, 자바 스크립트 참조 여부





## 2) 사이트의 메인부터 네비게이션 분석 → 도식화(pigma, draw.io, ppt 등 사용 가능)

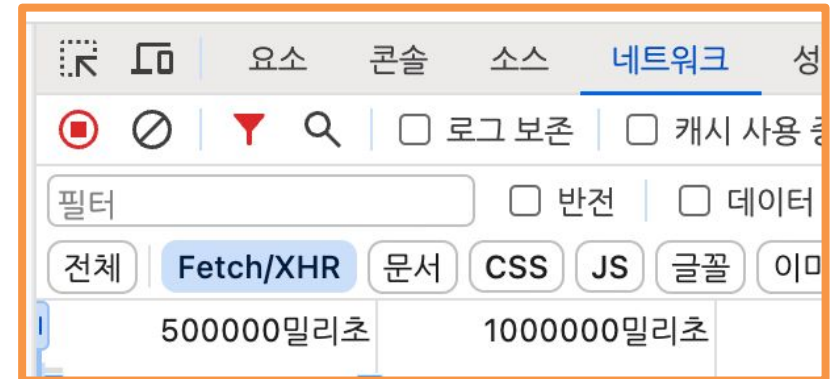
- 메인 메뉴 3개 이상
- 깊이 3이상
- 예시



### 3) 개발자 도구를 F12눌러 실행 후, 네트워크 탭을 분석

#### - 주요 분석 항목

- 동일한 하나의 서버만을 사용했는가?
- 서버 도메인이 몇 개나 되는가?
- 어떤 목적으로 서버를 여러 개로 분산했을까?(예, 미디어서버, 파일 서버 등)
- 각 항목별 시간 체크
- 비동기 통신을 포함하고 있는가?
  - 화면의 어느 부분을 구현하기 위해 비동기 통신을 사용했는가?
- 특정한 파일 2개 이상의 **Http Header**를 분석하시오.
  - 상태 코드
  - 요청 **URL** 확인 및 분석(예, 전달되는 데이터)
  - 요청 메서드
  - Content-type
  - 디스크 캐시, origin
  - 서버 종류 확인(예, nginx)
  - 기타



## 4) 개발자 도구를 F12눌러 실행 후, 애플리케이션 탭을 분석

### - 주요 분석 항목

- 쿠키 저장 값 확인
- 로컬 스토리지 확인
- 비공개 상태 토큰

The screenshot shows the Chrome DevTools Application tab with the 'Application' panel selected. The left sidebar shows the 'Storage' section expanded, with 'Cookies' selected for the domain 'https://www.coupang.com'. The main panel displays a table of cookies.

이름	값
CPUSR_RL	ArWvMpegJub7qNp2AICifwyTiSner5%2B3I%2FRCrUM8HO8UyV8d...
CSID	6591bheZUI8LTdYX61s1qQ7JfHS2nqTS~vvikNNqHfdrlYrxAPVpT9VE...
CT_LSID	GBMCbbLRWIXcsY6iRAI4ISHvkfP7eevfBtrcuZ0Lji0
CUPT	nnTIPAA1LCttht~NlnjHmM230iiZSUtNLnB5sidOFMPssuZvj4EARKpfIE...
ILOGIN	Y
MARKETID	16795512262224973034036
PCID	16795512262224973034036
_abck	409C02BD0EE6B3FD78DCC9B982D2865F~-1~YAAQRJIMF/GayliPA...
_fbp	fb.1.1679551227898.211770379
ak_bmsc	9E9AD056A406CECA8C9B79915A33FDEE~0000000000000000...
ar_debug	1
baby-isWide	small
bm_sv	0C4100A3685158317D26C729BD80E123~YAAQRJIMF+2ayliPAQAA...
bm_sz	34A12CF27FE47622B7F6E48F349E491F~YAAQRJIMF0eayliPAQAA...
cto_bundle	7wFedyV9XTicMk~JENDEDMQEPQW5Yx0PLIT~G3dDMHUI03HizMm5...

팀별 추가항목을  
더 설정해서  
분석해보세요!!



