

초격차 패키지 Online.

Start with Boot!

PART1 | 강사 소개, 사용 도구 소개

오리엔테이션 느낌으로 가볍게

PART2 | 프로젝트 셋업

자바 프로젝트 만들기, 깃헙에 올리기

PART3 | OOP 로 만든 정렬 구현체

Spring 공부를 위한 워밍업

PART4 | Spring의 적용

Spring 이 무엇을 도와주는가? 왜 써야 하는가?

PART5 | Spring Boot의 적용

왜 Spring Boot 로 시작해야 하는가?

Start with Boot!

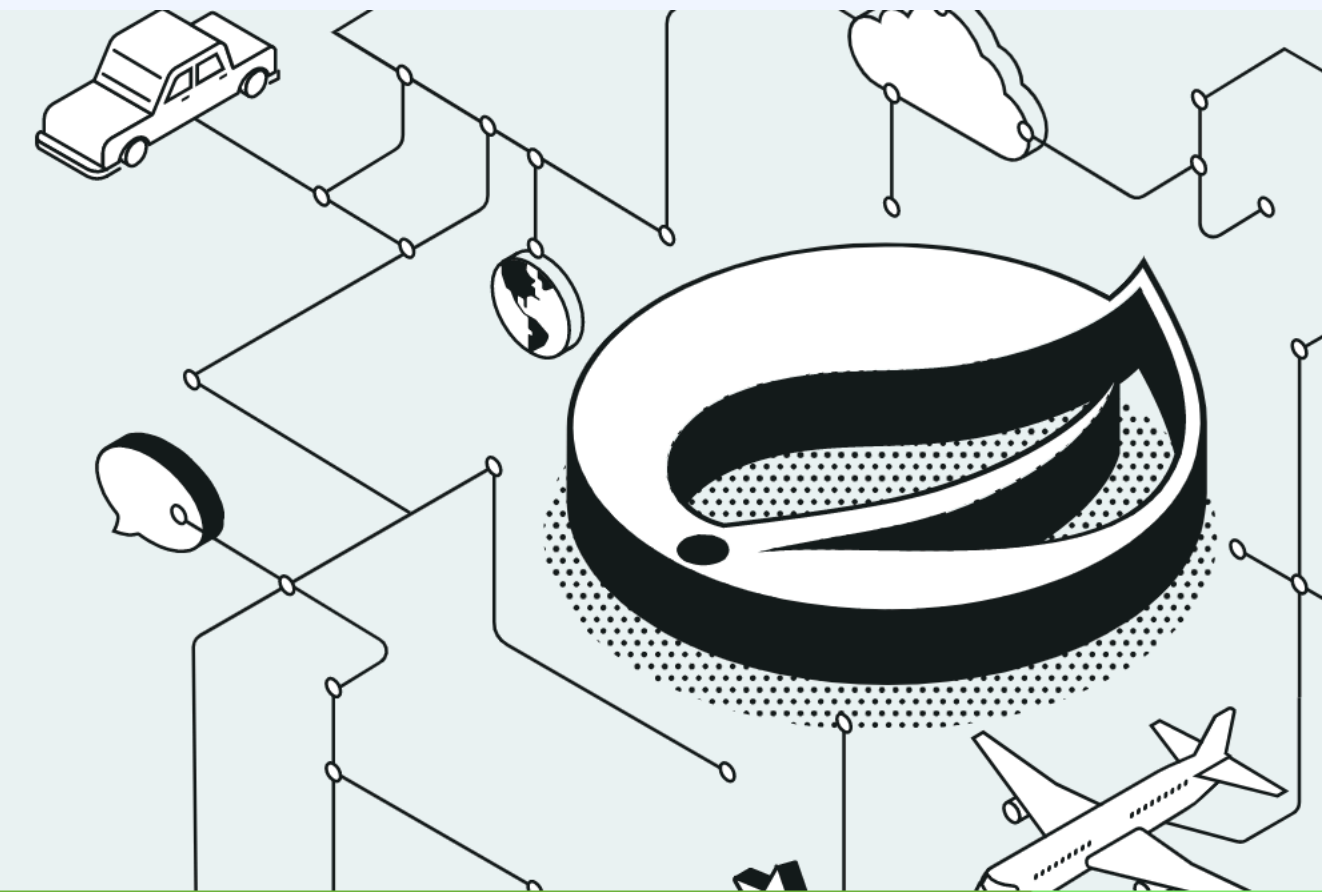
4 Spring의 적용

Spring

4. Spring의 적용

Why Spring?

Spring makes programming Java quicker, easier, and safer for everybody. Spring's focus on speed, simplicity, and productivity has made it the [world's most popular](#) Java framework.



“We use a lot of the tools that come with the Spring framework and reap the benefits of having a lot of the out of the box solutions, and not having to worry about writing a ton of additional code—so that really saves us some time and energy.”

SEAN GRAHAM, APPLICATION TRANSFORMATION LEAD, DICK'S SPORTING GOODS

Watch now

Spring

4. Spring의 적용

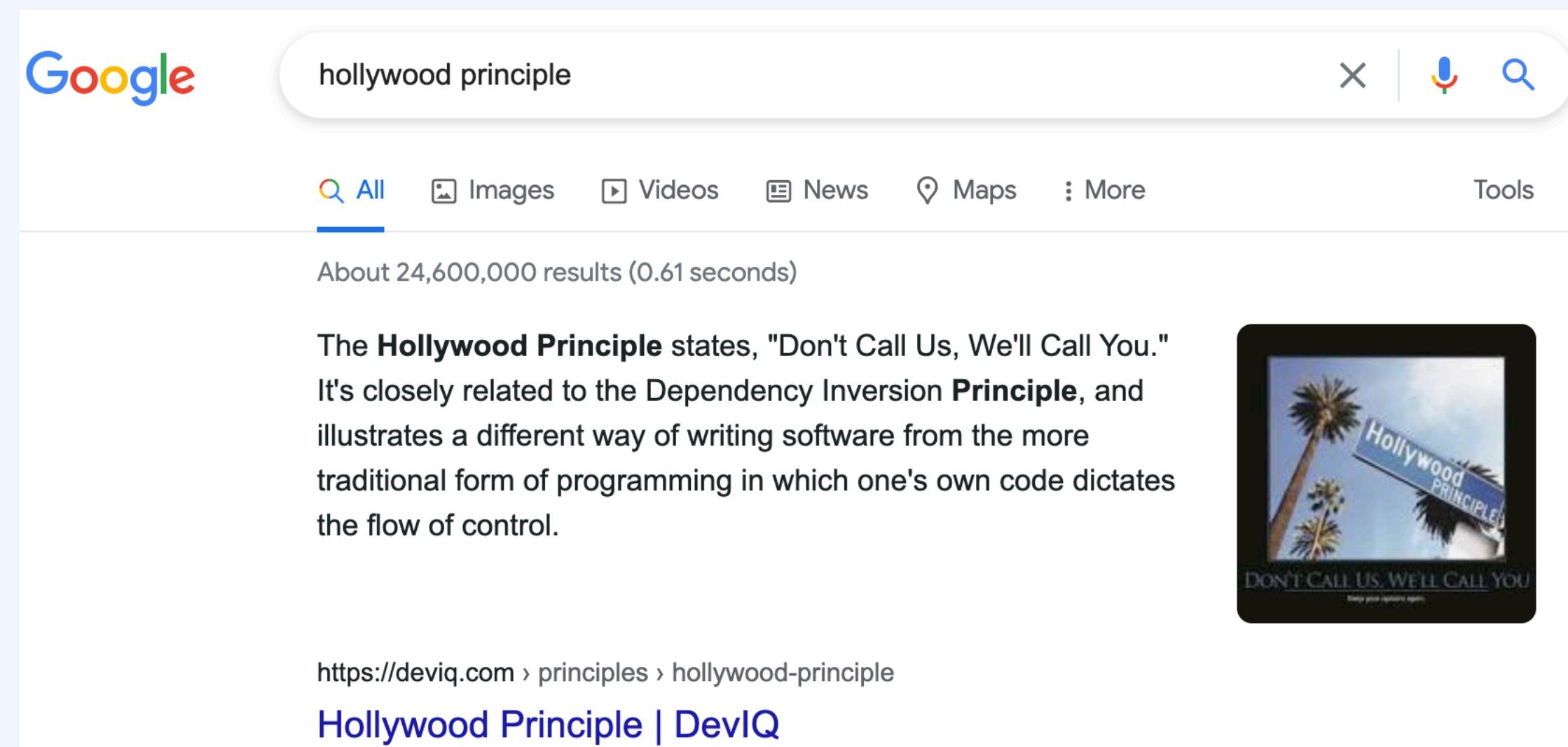
Spring 핵심 기능

1. The IoC Container
2. Resources
3. Validation, Data Binding, and Type Conversion
4. Spring Expression Language (SpEL)
5. Aspect Oriented Programming with Spring
6. Null-safety
7. Logging

1. The IoC Container

IoC (Inversion of Control)

"전통적인 제어 흐름에 비추어볼 때, 제어 흐름을 반대로 뒤집은 것" - Wikipedia



1. The IoC Container

"라이브러리를 사용할 때는 내 코드가 라이브러리 코드를 호출하지만
프레임워크를 사용할 때는 프레임워크가 내 코드를 호출합니다."

```
@SpringBootTest
class HelloServiceTest {

    @Autowired
    private HelloService helloService;

    @Test
    void test() {
        // Given
        String input = "Uno";

        // When
        String result = helloService.hello(input);

        // Then
        Assertions.assertEquals("Hello Uno!", result);
    }
}
```

프레임워크

라이브러리

2. Resource

low-level resource 에 접근할 수 있는 보다 폭넓은 기능을 제공

- UrlResource
- ClassPathResource
- FileSystemResource
- PathResource
- ServletContextResource
- InputStreamResource
- ByteArrayResource

3. Validation, Data Binding, Type Conversion

- 데이터의 검증
- 데이터를 인식하고 자료형에 할당
- 데이터 자료형의 변환

```
public class PersonForm {

    @NotNull
    @Size(min=2, max=30)
    private String name;

    @NotNull
    @Min(18)
    private Integer age;

    ....

}
```

```
@GetMapping("/api/employees/{id}")
@ResponseBody
public String getEmployeesById(@PathVariable String id) {
    return "ID: " + id;
}
```

```
package org.springframework.core.convert.support;

final class StringToInteger implements Converter<String, Integer> {

    public Integer convert(String source) {
        return Integer.valueOf(source);
    }

}
```


4. SpEL

4. Spring의 적용

- 스프링 애플리케이션의 런타임에 다양한 데이터에 접근하기 위한 언어
- JSP Unified EL 과 유사하지만 스프링에 특화되어 더 다양한 기능을 제공

```
@Value("${value.from.file}")  
private String valueFromFile;
```

5. Aspect Oriented Programming with Spring

- AOP: 관점 지향 프로그래밍 - 공통 기능을 개발자의 코드 밖에서 필요한 시점에 적용 가능
- AOP 를 적극적으로 사용하고 지원하는 프레임워크
- Proxy, Aspect, Join Point, Advice, Pointcut, Weaving
- CGLib, AspectJ
- AOP 를 사용하지 않아도, 심지어 몰라도 여전히 프레임워크를 사용 가능

비즈니스 로직 뿐인데, 트랜잭션을 보장하는 작업이 일어나고 있음

```
@Transactional
public boolean update(int score) {
    return scoreService.updateScores(score);
}
```

6. Null-safety

4. Spring의 적용

null-safe 한 코드를 작성할 수 있는 다양한 방법을 지원

- @Nullable
- @NonNull
- @NonNullApi
- @NonNullFields

7. Logging

4. Spring의 적용

별도의 외부 설정 없이 로깅 구현체 사용 가능

- SLF4J + Logback
- Log4j 2
- JUL (java.util.logging)

그 밖에...

4. Spring의 적용

웹 애플리케이션을 만드는데 유용한 여러 지원들

- Testing
- Data Access
- Web Servlet
- Web Reactive
- Integration: REST endpoints, email, scheduling, cache, ...

Spring makes it easy to create Java enterprise applications. It provides everything you need to embrace the Java language in an enterprise environment, with support for Groovy and Kotlin as alternative languages on the JVM, and with the flexibility to create many kinds of architectures depending on an application's needs.

스프링은 엔터프라이즈 애플리케이션을 만드는데 필요한
거의 모든 요소를 지원하는 프레임워크

Reference

4.

Spring의 적용

- <https://github.com/djkeh/fastcampus-spring-practice>