

제01장

# Spring Framework

Spring

# 학습목표

1. 프레임워크에 대해서 알 수 있다.
2. 스프링 프레임워크에 대해서 알 수 있다.

```
each: function(e, t, n) {  
    o = e.length,  
    a = M(e);  
    if (n) {  
        if (a) {  
            for (; o > i; i++)  
                if (r = t.apply(e[i], n), r ===  
        } else  
            for (i in e)  
                if (r = t.apply(e[i], n), r ===  
    } else if (a) {  
        for (; o > i; i++)  
            if (r = t.call(e[i], i, e[i]))  
    } else  
        for (i in e)  
            if (r = t.call(e[i], i, e[i]))  
    return e  
},  
trim: b && !b.call("\uffff\u00a0") ?  
    return null == e ? "" : b.call(  
} : function(e) {  
    return null == e ? "" : (e + "  
},  
makeArray: function(e, t) {  
    var n = t || [];  
    return null != e && (M(Ob  
},  
isArray: function(e, t, n) {  
    var r;  
    if (t) {  
        if (n) return m.c  
        for (n = t.length  
            if (n in t  
    }  
}
```

# 목차

1. 프레임워크
2. 스프링 프레임워크

```

each: function(e, t, n) {
  var r, i = 0,
      o = e.length,
      a = M(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r === !1) break
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break
  } else if (a) {
    for (; o > i; i++)
      if (r = t.call(e[i], e[i]), r === !1) break
  } else
    for (i in e)
      if (r = t.call(e[i], e[i]), r === !1) break;
  return e
},
trim: b && !b.call("\uffff\u00a0") ? function(e) {
  return null == e ? "" : b.call(e)
} : function(e) {
  return null == e ? "" : (e + "").replace(C, "")
},
makeArray: function(e, t) {
  var n = t || [];
  return null != e && (M(Object(e)) ? x.merge(n, "string"
),
isArray: function(e, t, n) {
  var r;
  if (t) {
    if (n) return m.call(t, e, n);
    for (r = t.length, r = r ? 0 > n ? Math.max(0, r + n
      if (n in t && t[n] === e) return n
  }
}

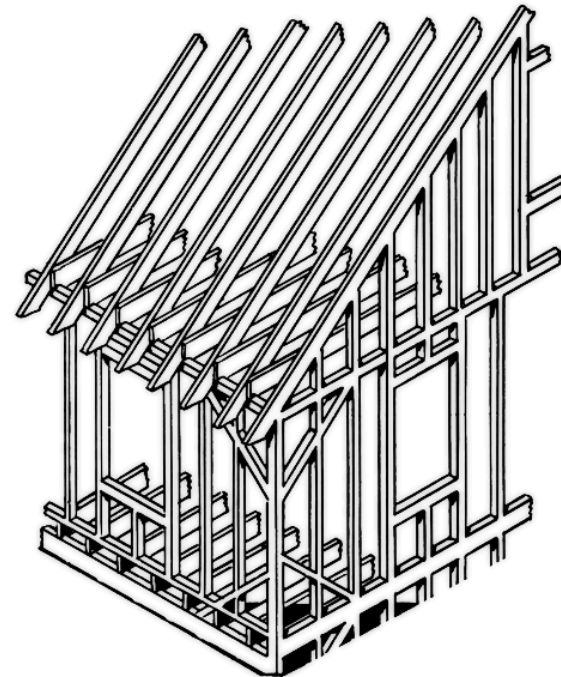
```

# 1. 프레임워크

# Framework 개념

## ■ Framework

- Framework를 한글로 번역하면 '뼈대'라고 함
- 애플리케이션의 기본 구조를 제공하는 것이 Framework임
- Framework는 애플리케이션을 개발할 때 사용할 수 있도록 미리 만들어 놓은 클래스나 인터페이스 등을 제공함
- 스트러츠, 스프링과 같은 Framework 들이 있음



기본 틀을 제공하는 Framework

# Framework 특징

## ■ Framework 특징

- 처음부터 다 만들 필요가 없다.
  - 이미 만들어진 기능을 가져다 사용하므로 효율성이 높고 품질이 보장됨
  - 개발 시간이 짧아짐
- 개발 방법이 정해져 있다.
  - 개발의 표준화로 인해서 생산성이 높아짐
  - 개발 후 유지보수 및 기능의 확장이 용이함
  - 신입 개발자도 경력 개발자처럼 세련된 코드를 작성할 수 있음

# 전자정부 표준프레임워크

## ■ 전자정부 표준프레임워크

- 공공 기관에서 개발하는 웹 애플리케이션에 대해 공통적으로 사용할 수 있는 기반 기술과 라이브러리를 제공
- 정부 관련 프로젝트 진행 시 사용하는 개발 생태계
- 개발된 컴포넌트와 모듈을 다른 프로젝트에서도 재사용 가능
- 새로운 기능의 추가나 수정이 용이함
- 공공 정보 시스템에 필수적인 보안 기능 내장
- Spring Framework 를 채택하여 사용하고 있음

<https://www.egovframe.go.kr/>

```

each: function(e, t, n) {
  var r, i = 0,
      o = e.length,
      a = M(e);
  if (n) {
    if (a) {
      for (; o > i; i++)
        if (r = t.apply(e[i], n), r === !1) break;
    } else
      for (i in e)
        if (r = t.apply(e[i], n), r === !1) break;
  } else if (a) {
    for (; o > i; i++)
      if (r = t.call(e[i], e[i]), r === !1) break;
  } else
    for (i in e)
      if (r = t.call(e[i], e[i]), r === !1) break;
  return e;
},
trim: b && !b.call("\uff\u00a0") ? function(e) {
  return null == e ? "" : b.call(e);
} : function(e) {
  return null == e ? "" : (e + "").replace(C, "");
},
makeArray: function(e, t) {
  var n = t || [];
  return null != e && (M(Object(e)) ? x.merge(n, "string"
),
isArray: function(e, t, n) {
  var r;
  if (t) {
    if (n) return m.call(t, e, n);
    for (r = t.length, r = r ? 0 > n ? Math.max(0, r + n) :
      if (n in t && t[n] === e) return n;
  }
}

```

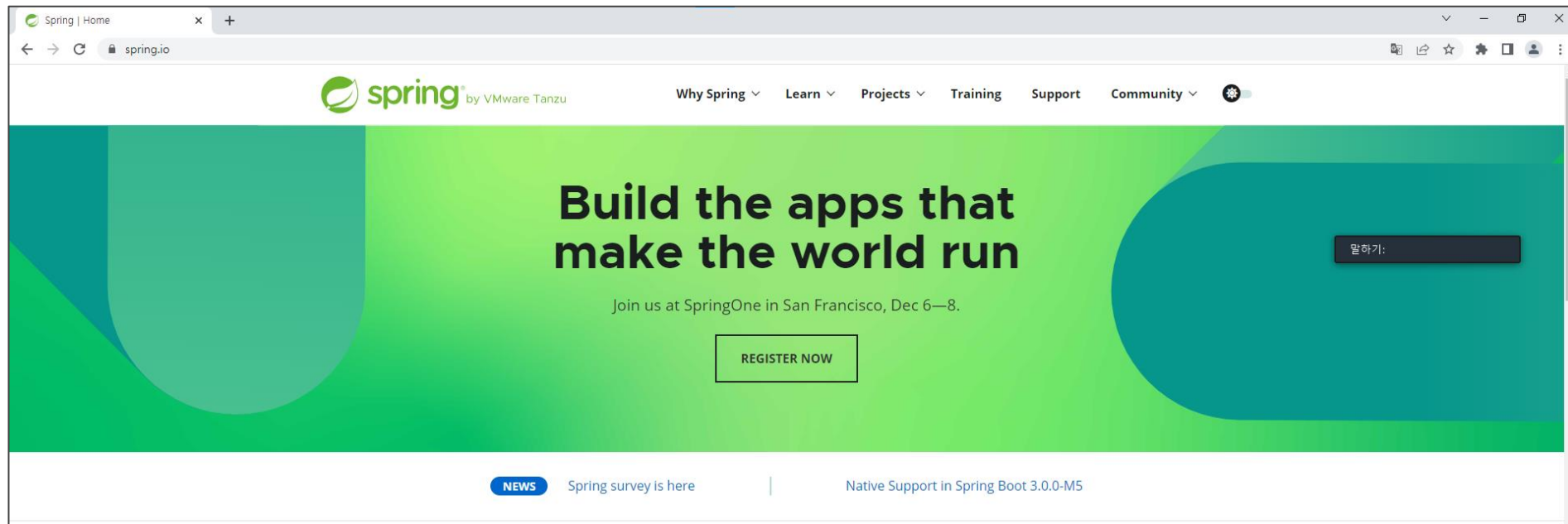
## 2. 스프링 프레임워크



# Spring Framework

## ■ Spring Framework

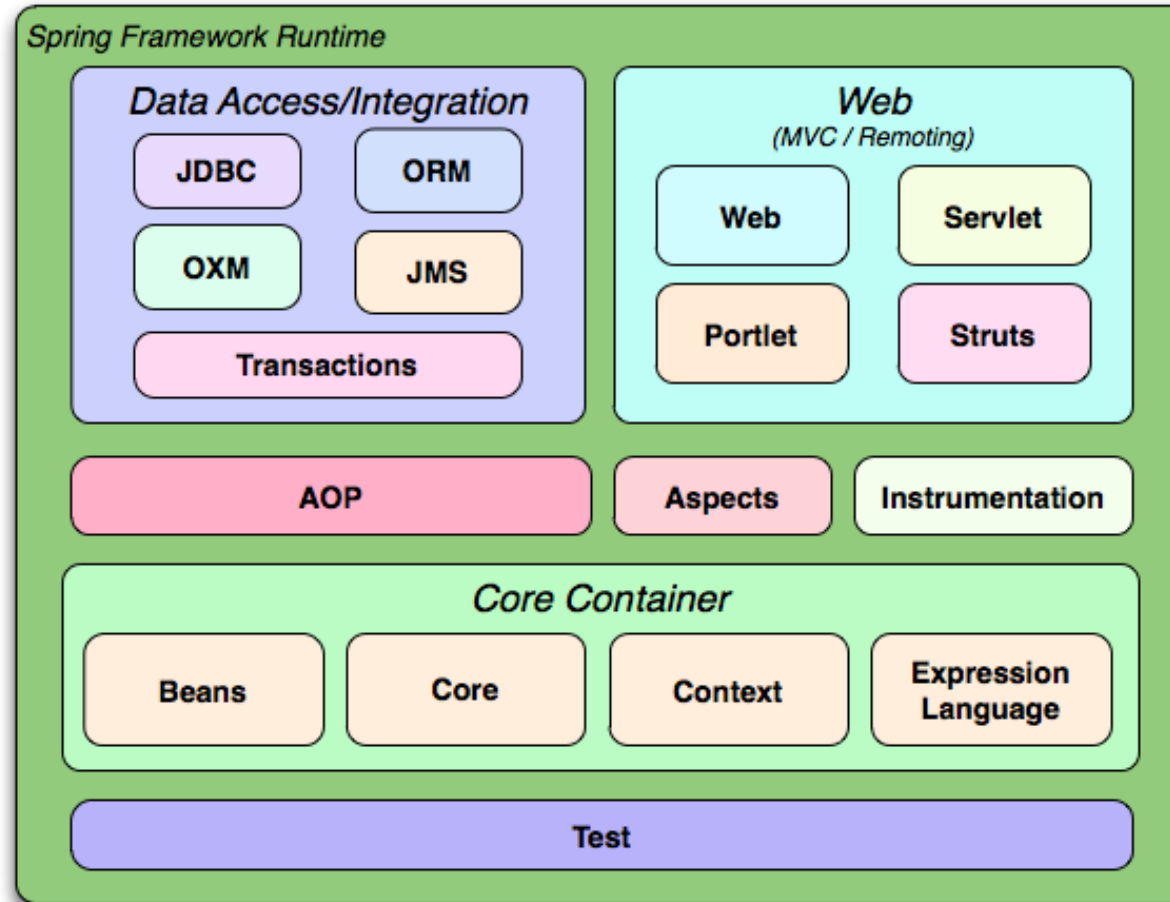
- 2004년 로드 존슨(Rod Johnson)이 만든 오픈소스 프레임워크
- 가장 널리 사용하는 프레임워크 중 하나
- 자바 기반의 엔터프라이즈 애플리케이션 개발을 담당하던 EJB(Enterprise JavaBeans)를 대체함



공식 홈페이지 <https://spring.io>

# Spring Framework 구조

- Spring Framework 구조 (출처 : <https://docs.spring.io/>)



# Spring Framework 특징

## ■ 경량 프레임워크 (Lightweight)

- EJB에 비해 가벼운 경량 프레임워크(Lightweight Framework)
- 몇 개의 모듈과 JAR 파일로 구성

## ■ 컨테이너 지원 (Container)

- 특정 객체의 생성과 관리를 담당하며 객체 운용에 필요한 다양한 기능을 제공
- Singleton, Prototype 등 여러 형태의 객체 운용 가능

## ■ 제어의 역행 (IoC)

- 객체 생성을 개발자가 대신 컨테이너가 담당
- 애플리케이션을 구성하는 객체 간 결합을 느슨한 결합(낮은 결합도)으로 유지

## ■ 관점지향 프로그래밍 (AOP)

- 비즈니스 메소드를 구성하는 공통 모듈과 핵심 모듈을 분리하여 작성
- 모든 비즈니스 메소드가 반복해서 사용하는 공통 로직을 핵심 비즈니스 로직과 구분하여 작성

# Spring Framework 모듈

## ■ Spring Framework 모듈

모듈	설명
spring-beans	스프링 컨테이너를 이용해서 객체를 생성하는 기본 기능 제공
spring-context	객체 생성, 라이프 사이클 처리, 스키마 확장과 같은 기능 제공
spring-aop	AOP 기능 제공
spring-web	REST 클라이언트, 데이터 변환 처리, 서블릿 필터, 파일 업로드 지원 등 웹 개발에 필요한 기반 기능을 제공
spring-webmvc	스프링 기반의 MVC 프레임워크 제공, 웹 애플리케이션을 개발하는데 필요한 컨트롤러, 뷰 구현을 제공
spring-websocket	스프링 MVC에서 웹 소켓 연동 처리
spring-tx	트랜잭션 처리를 위한 추상 레이어 제공
spring-jdbc	JDBC 프로그래밍을 보다 쉽게 할 수 있는 템플릿 제공
spring-orm	하이버네이트, JPA, MyBatis 등과의 연동 지원
spring-jms	JMS 서버와 메시지를 쉽게 주고 받을 수 있도록 하기 위한 템플릿, 애노테이션 제공
spring-context-support	스케줄링, 메일 발송, 캐시 연동, 벨로시티 등 부가 기능 제공

# 개발 환경 준비

## ■ Spring 개발 준비

- Language

- JDK
- JAVA\_HOME 등록 및 Path 설정

- WAS

- Apache Tomcat Server
- CATALINA\_HOME 등록 및 Path 설정

- IDE

- STS 또는 vscode

# 개발 환경 준비

## ■ STS (Spring Tool Suite)

- Spring Framework 사용을 위한 각종 라이브러리를 추가해 둔 IDE
- Eclipse 기반으로 Eclipse와 거의 동일한 개발 환경을 제공함
- STS3와 STS4 버전이 제공
  - STS3 : Spring Legacy Project 지원. 현재 더 이상 지원 되지 않음. (jdk-11 필요)
  - STS4 : Spring Boot Project 지원. (jdk-17 이상)
- 다운로드 링크
  - STS3 : <https://github.com/spring-attic/toolsuite-distribution/wiki/Spring-Tool-Suite-3>
  - STS4 : <https://spring.io/tools>