# 테스트와 로깅, 빌드

하지만 프로그래밍에서는 계산을 위한 산술 연산자 뿐만 아니라 훨씬 다양한 역할의 연산자들이 사용됩니다. 값의 크고 작음을 비교하기 위한 비교 연산자, 변수에 값을 대입하는 데 사용하는 대입 연산자, 산술 연산자와 대입 연산자가 합쳐진 증감 연산자, 참 거짓의 집합 연산 결과를 처리하는 논리 연산자 등은 모두 프로그래밍에서 연산자의 범주에 포함되는 것들입니다.

모든 연산자를 숙지하고 있을 필요는 없습니다. 하지만 연산자는 대부분 컴퓨터 언어에서 공통일 뿐만 아니라 코딩 기초 개념을 이해하기 위해 필수적인 요소이므로, 잘 알아두면 다른 언어를 배울 때 그만큼 시간과 노력을 줄일 수 있습니다.

## 스프링 부트 테스트

가장 먼저 배워볼 것은 대입 연산자입니다. 사실 이 연산자는 여러분이 이미 알고 있을텐데요, 바로 변수에 값을 대입할 때 사용한 '='입니다. ‘연산자 오른쪽에 있는 값을 왼쪽의 변수에 대입한다’라는 의미를 가지는 연산자로, 수학 시간에 사용하는 등호('=')와 모양이 같지만 의미가 전혀 다르므로 사용에 주의해야 한다고 설명한 기억이 나는군요.

### 스프링 부트에서 테스트하기

대입 연산자의 사용방법을 다시 한번 확인해 봅시다. 오른쪽의 값을 왼쪽에 대입하는 것이 원칙입니다.

객체 개념을 지원하지 않는 언어에서는 속성과 기능을 함께 정의할 수 없습니다. 속성은 속성끼리만 묶을 수 있고 기능은 기능끼리만 묶을 수 있을 뿐, 속성은 기능을 포함할 수 없으며 기능 또한 속성을 소유할 수 없습니다. 따라서 이런 류의 언어에서는 항상 속성과 기능을 나누어 정의해야 합니다. 속성은 변수로, 기능은 함수로 말입니다. 반면 객체를 다루는 언어에서는 속성과 기능을 함께 다룰 수 있습니다.

#### 산술 연산자

소제목은 워드 스타일에서 [제목 4]를 클릭합니다(대제목과 마찬가지로 스타일에서 [제목 4]를 클릭합니다).

산술 연산자는 대입 연산자와 함께 프로그래밍에서 가장 많이 사용되는 연산자입니다. 값을 더하거나 빼고, 곱하거나 나누고 제곱을 계산하는 등 수학 계산을 처리하는 데 사용하죠. 가장 잘 알려진 산술 연산자로는 사칙 연산자가 있습니다. 아마도 다들 익숙할 겁니다. 초등학교 이후로 많이 사용해 왔을 테니까요.

### 부트에서 테스트하기

숫자와 문자열을 + 연산자로 연결하는 경우에는 어떻게 될까요? 프로그래밍에서는 데이터 타입이 다른 경우 원칙적으로 연산이 불가능합니다. 다만 한 쪽 데이터를 다른 쪽의 데이터 타입으로 바꿀 수 있다면 해당 타입으로 연산이 처리됩니다. 바꿀 수 없다면요? 오류죠.

BoardServiceImpl.java

01 : package com.rubypaper.service;

02 :

03 : import java.util.ArrayList;

04 : import java.util.Date;

05 : import java.util.List;

06 :

07 : import org.springframework.stereotype.Service;

08 :

09 : import com.rubypaper.domain.BoardVO;

10 :

11 : @Service

12 : public class BoardServiceImpl implements BoardService {

13 :

14 : public String hello(String name) {

15 : return "Hello : " + name;

16 : }

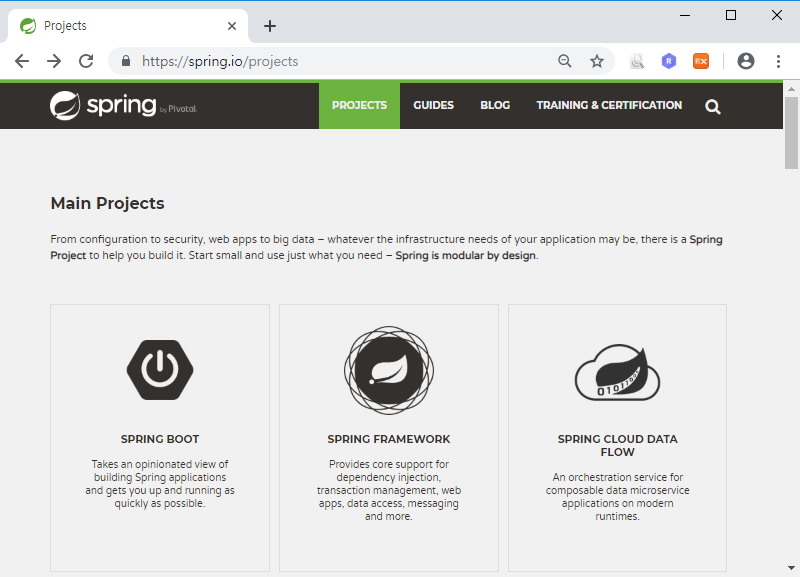
17 :

18 : public BoardVO getBoard() {

19 : BoardVO board = new BoardVO();

20 : board.setSeq(1);

숫자 1과 2, 그리고 문자 "3"의 더하기 연산은 순서에 따라 다른 결과를 만들어 냅니다. 1행과 2행은 앞에서부터 순서대로 처리되어 "33"이라는 결과를 만들지만, 마지막 3행은 괄호로 인해 뒤쪽부터 연산이 수행됩니다. 우선 숫자 2와 문자열 3이 결합하여 문자열 "23"이 만들어지고, 여기에 다시 숫자 1이 결합하면서 "123"이 되죠. 이처럼 데이터 형이 섞여 있을 경우 연산 순서에 따라 결과가 달라지기 때문에 주의해야 합니다



[그림 3-1] 스프링 프레임워크 홈페이지

<주의>

/my-app/dist/index.html은 빌드(ng build)의 결과물로 실제 배포 시에는 서버로 이관된다. ng serve 명령어에 의해 내장 개발 서버를 사용하여 로컬 환경에서 프로젝트를 실행(preview)하는 경우, Angular CLI 내부적으로 빌드를 자동 수행하므로 빌드(ng build)를 별도 실행하여 /my-app/dist/index.html을 생성할 필요는 없다. 자동으로 빌드되어 가상 드라이브에 저장되어 있는 index.html을 내장 개발 서버가 로드한다고 이해하면 된다.

</주의>

<노트>

2016년 08월 1.0.0-beta.11부터 Angular의 빌드 시스템은 SystemJS에서 Webpack으로 변경되었다.[[1]](#footnote-1)

</노트>

private async void startButton\_Click(object sender, RoutedEventArgs e)

{

await DoSomethingAsync();

}

### 목록

본문 중 여러 항목을 열거해야 할 경우 스타일에서 [목록 표시]를 클릭합니다.

* 라이브러리 : 프로그래밍할 때 자주 사용하는 기능을…
* 실행 환경 : 프로그램을 실제로 동작시키는 환경…
* commando : 프로그래밍할 때 자주 사용하는 기능을 모아…
* Rambo : 프로그램을 실제로 동작시키는 환경을 뜻…

1. [↑](#footnote-ref-1)