

The background features a large, abstract, wavy shape in shades of green, resembling a stylized wave or a flowing ribbon. It starts from the left, curves upwards and then downwards, and ends on the right side. The color transitions from a lighter green on the left to a darker green on the right. A solid dark green horizontal bar is positioned at the very bottom of the image.

Introduction to Unit Test

JUnit

- 자바 애플리케이션 개발에 사용되는 단위 테스트 도구
- 에릭 감마와 켄트 벅이 최초 개발
- 자바 개발 영역에서 단위 테스트를 위한 실질적인 표준 프레임워크로 발전

라이브러리 설치

```
<dependency>  
  <groupId>junit</groupId>  
  <artifactId>junit</artifactId>  
  <version>4.12</version>  
  <scope>test</scope>  
</dependency>
```

단위 테스트 도구

```
<dependency>  
  <groupId>org.hamcrest</groupId>  
  <artifactId>hamcrest-core</artifactId>  
  <version>2.2</version>  
  <scope>test</scope>  
</dependency>
```

테스트 매칭 도구

```
<dependency>  
  <groupId>org.testng</groupId>  
  <artifactId>testng</artifactId>  
  <version>7.6.0</version>  
  <scope>test</scope>  
</dependency>
```

단위 테스트 도구

Junit의 기본 구조

■ 3.x

- TestCase 상속
- 테스트 메서드의 이름은 test로 시작
- 테스트 메서드의 반환 형식은 void
- 테스트 메서드의 전달인자는 없음

■ 4.x

- Annotation으로 Test 설정
- 상속과 테스트 메서드에 대한 이름 제약 없음
- 메서드 반환 형식과 전달인자 규칙은 동일

주요 테스트 어노테이션

Annotation	설명
@BeforeClass	테스트 클래스의 테스트가 시작되기 전에 한 번 실행되는 메서드
@Before	각 테스트 메서드 전에 실행되는 메서드
@Test	테스트 메서드
@After	각 테스트 메서드 후에 실행되는 메서드.
@AfterClass	테스트 클래스의 테스트가 종료된 후에 한 번 실행되는 메서드
@Test(timeout)	테스트가 일정 시간을 초과하면 실패하게 만드는 속성
@Test(expected)	발생 예상되는 예외를 지정하는 속성

Assert 클래스

- JUnit은 테스트의 성공 실패를 판단하는 데 사용되는 공통 메서드 제공
- JUnit 3.x 버전까지는 TestCase 클래스에서 제공하는 assertXXX 메서드를 상속 받아서 사용
- JUnit 4.x 버전부터 assertXXX 메서드를 static 형태로 정의한 Assert 클래스 제공

주요 assert 메서드

메서드	설명
<code>assertArrayEquals</code>	두 개의 1차원 배열 값이 동일한 경우 성공 반환
<code>assertEquals</code>	두 객체의 값이 같을 경우 성공 반환 (vs <code>assertSame</code>)
<code>assertFalse</code>	<code>boolean</code> 변수의 값이 <code>false</code> 인 경우 성공 반환
<code>assertNotEquals</code>	두 객체의 값이 같지 않을 경우 성공 반환 (vs <code>assertNotSame</code>)
<code>assertNotNull</code>	변수가 <code>null</code> 이 아닌 경우 성공 반환
<code>assertNotSame</code>	두 객체의 참조가 같지 않을 경우 성공 반환 (vs <code>assertNotEquals</code>)
<code>assertSame</code>	두 객체의 참조가 같을 경우 성공 반환 (vs <code>assertEquals</code>)
<code>assertThat</code>	<code>assert</code> 메서드를 보완하기 위해 만든 메서드 (4.4 or above)
<code>assertTrue</code>	<code>boolean</code> 변수의 값이 <code>true</code> 인 경우 성공 반환
<code>fail</code>	테스트 실패 처리. 테스트 케이스 내부의 코드에서 오류가 발생하거나 정지가 필요할 때 사용

JUnit Suite

- 여러 개의 JUnit 단위 테스트를 모아서 일괄 테스트를 수행할 때 사용
- `@BeforeClass`, `@AfterClass` Annotation 사용

테스트 케이스 만들기

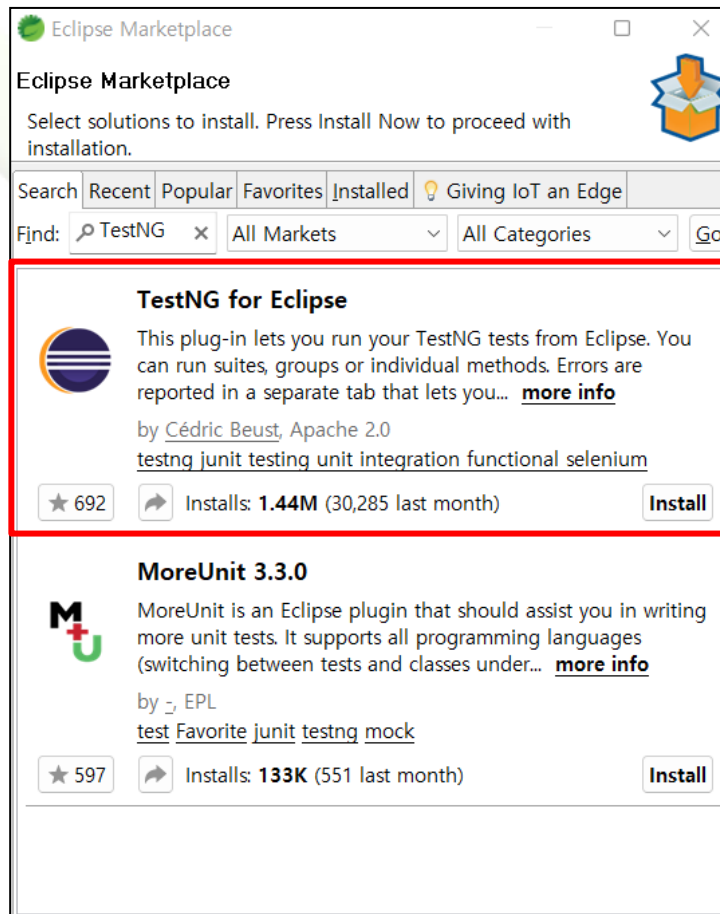
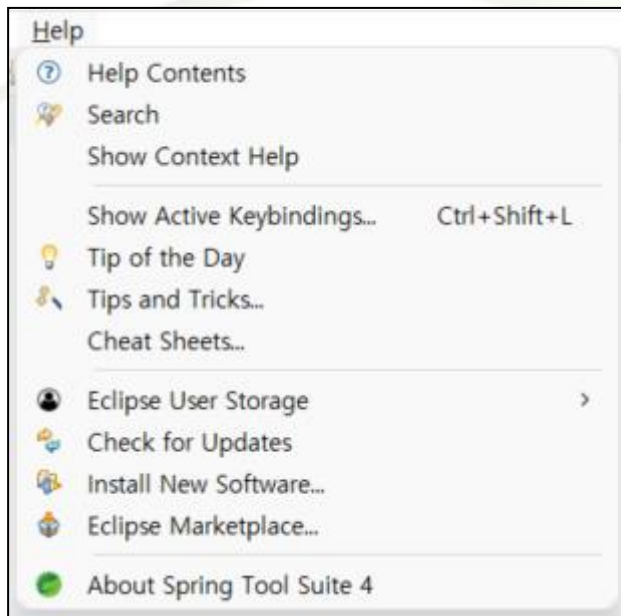
- 테스트 케이스
 - @Test Annotation이 적용된 public 메서드 정의
- 테스트 전후에 호출되는 메서드 정의
 - 테스트 메서드 호출할 때 매번 호출
 - » @Before Annotation 적용 메서드 → 리소스 준비
 - » @After Annotation 적용 메서드 → 리소스 해제
 - 한 번만 호출
 - » @BeforeClass Annotation 적용 메서드 → 리소스 준비
 - » @After Annotation 적용 메서드 → 리소스 해제

TestNG

- JUnit과 NUnit의 개념을 기반으로 개발된 자바 테스트 프레임워크
- 주요 기능
 - 다양한 Annotation 제공
 - Multi-Thread 기반으로 안전한 테스트 실행 가능
 - 유연한 테스트 설정 → XML 기반 설정 및 전달인자로 조정 가능
 - 데이터 기반 테스트 지원 (@DataProvider Annotation)
 - 파라미터 지원
 - 강력한 실행 모델 지원
 - 다양한 도구와 플러그인 지원 (eclipse, IntelliJ IDEA, Maven 등)
 - 로깅을 위해 JDK의 기본 기능 사용 (추가 의존성 없음)
 - 애플리케이션 서버 기반 테스트를 위한 방법 제공

TestNG 설치

■ Eclipse Plugin 설치



TestNG Annotation

Annotation	설명
@BeforeTest	테스트가 실행되기 전에 실행되는 메서드
@BeforeSuite	테스트 스위트가 실행되기 전에 실행되는 메서드
@BeforeClass	테스트 클래스의 테스트가 시작되기 전에 실행되는 메서드
@BeforeMethod	각 테스트 메서드 전에 실행되는 메서드
@Test	테스트 메서드.
@AfterMethod	각 테스트 메서드 후에 실행되는 메서드.
@AfterClass	테스트 클래스의 테스트 후에 실행되는 메서드
@AfterSuite	테스트 스위트가 실행된 후 호출되는 메서드
@AfterTest	테스트가 실행된 후 호출되는 메서드

TestNG 테스트 설정 파일

■ 설정 파일 구조

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="test-suite-name" verbose="1">
  <groups>
    <run>
      <exclude name="group-name" />
      <include name="group-name" />
    </run>
  </groups>
  <test name="test-name">
    <classes>
      <class name="class-name" />
    </classes>
  </test>
  <test name="test-name">
    <packages>
      <package name="package-name" />
    </packages>
  </test>
</suite>
```

TestNG 테스트 설정 파일

■ 설정 파일 구성 요소

구성 요소	설명
<test>	테스트 정의 요소. name 속성으로 테스트 이름 지정
<classes>	테스트를 수행할 클래스 목록
<class>	classes 하위 요소. 수행할 테스트 클래스를 name 속성에 지정
<packages>	테스트를 수행할 패키지 목록
<package>	packages 하위 요소. 수행할 패키지를 name 속성으로 지정
<groups>	테스트 대상 클래스 또는 패키지의 테스트를 수행할 때 수행하거나 수행하지 않을 그룹 관리
<run>	group의 하위 요소. 실행하거나 실행하지 않을 항목 지정
<exclude>	run의 하위 요소. 실행하지 않을 그룹 지정
<include>	run의 하위 요소. 실행할 그룹 지정

Spring MVC 단위 테스트

- 라이브러리 추가

```
<dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-test</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
<dependency>
    <groupId>org.mockito</groupId>
    <artifactId>mockito-all</artifactId>
    <version>1.10.19</version>
    <scope>test</scope>
</dependency>
```

스프링 MVC 테스트

▪ Controller 테스트

```
@RunWith(SpringJUnit4ClassRunner.class)
@WebAppConfiguration()
@ContextConfiguration(locations = {
    "file:src/main/webapp/WEB-INF/spring/root-context.xml",
    "file:src/main/webapp/WEB-INF/spring/appServlet/servlet-context.xml"
})

public class AccountControllerTest {

    private MockMvc mockMvc;

    @Autowired
    private WebApplicationContext wac;

    @Before
    public void setup() {
        mockMvc = MockMvcBuilders.webAppContextSetup(wac).build();
    }

}
```


스프링 MVC 테스트

▪ Controller 테스트

```
public class AccountControllerTest {  
  
    @Test  
    public void loginTest() throws Exception {  
        mockMvc.perform(post("/account/login.action")  
                        .param("memberId", "iamuserone")  
                        .param("passwd", "992299222"))  
                .andExpect(status().is3xxRedirection())  
                .andDo(print());  
    }  
  
    @Test  
    public void registerTest() throws Exception {  
        assertTrue(true);  
        mockMvc.perform(post("/account/register.action")  
                        .param("memberId", "iamuserabc")  
                        .param("passwd", "99229922")  
                        .param("email", "iamuserabc@example.com"))  
                .andExpect(status().is3xxRedirection())  
                .andDo(print());  
    }  
}
```