

사전 학습

소프트웨어 개발 프로세스 이해하기 1
파이썬 (python), 자바 (java)

프로그래밍 언어의 종류

	스크립트 언어	컴파일 언어
특 징	<ul style="list-style-type: none">• 소스 코드를 컴파일(Compile)하지 않고도 실행할 수 있는 프로그래밍 언어• 유닉스의 셸(Shell) 스크립트, 윈도우의 배치(Batch) 스크립트처럼 일괄 처리 작업에 사용• 일반적으로 스크립트 언어는 빠른 성능이 필요하지 않은 웹 애플리케이션을 만드는 데 이용되어 왔으며, 크게 서버 사이드 스크립트(server-side-script)와 클라이언트 사이드 스크립트(client-side-script)로 나눌 수 있다.	<ul style="list-style-type: none">• 작성한 소스를 컴파일러를 통해 기계어 코드로 변경해야 사용자가 실행할 수 있는 실행 파일이 얻어지는 프로그래밍 언어
종 류	<ul style="list-style-type: none">• 서버 사이드 스크립트 : Python, ASP, PHP, JSP, PERL, RUBY 등• 클라이언트 사이드 스크립트 – javascript, Vbscript, JScript	<ul style="list-style-type: none">• Java, C, C++, C#, Objective-C, swift, small 등

파이선 개발환경 구축용 개발도구 이해

- 파이선 개발환경 구축 순서

1. Python download > 설치

2. IDE (Integrated Development Environment : 통합 개발 환경) 툴 선택해서 다운로드 > 설치

- 파이참 (pycharm) : 실무에서 주로 사용
- VisualStudio Code : 텍스트 편집기
- JupyterNote : 아나콘다 설치시 자동 설치됨
- Thony 등 : IoT 코드 편집기용으로 주로 사용

파이선으로 만들 수 있는 프로그램 종류

- 파이썬으로 할 수 있는 일

- 시스템 유틸리티 제작
- GUI 프로그래밍
- C/C++와의 결합
- 웹 프로그래밍
- 수치 연산 프로그래밍
- 데이터베이스 프로그래밍
- 데이터 분석, 사물 인터넷

- 파이썬으로 할 수 없는 일

- 시스템과 밀접한 프로그래밍 영역
- 모바일 프로그래밍

개발 구현과 디버깅 이해

- 파이선에서 코드를 실행하는 방법
 1. 작성한 코드 파일에서 바로 실행
 2. `_main` 함수를 통해 실행

테스트와 배포 이해

1. 개발툴 실행
2. 새 코드 파일 준비 : new > 파일이름.py
3. 코드 작성 > 저장
4. 디버깅 : break point 로 지정한 위치의 코드 실행 확인
5. 코드 실행 테스트
6. 패키징 : exe 파일 만들기

자바

- Java는 1995년부터 사용되어 온 오브젝트 지향적인 프로그래밍(Object-Oriented Programming, OOP) 언어입니다. 개발자는 Java를 사용하여 애플리케이션이 속한 도메인 경계 내에서 작동하는 애플리케이션을 프로그래밍합니다.
- Java와 JavaScript는 같은 시기에 각기 다른 회사에서 개발된 서로 다른 프로그래밍 언어이므로 둘을 혼동해서는 안 됩니다. Java 개발을 통해 브라우저나 서버의 베어 메탈 머신, 가상 머신 또는 컨테이너에서 실행할 수 있는 애플리케이션이 생성됩니다. JavaScript는 브라우저에서만 실행됩니다.

자바 개발환경 구축

1. JDK (Java Development toolKit) 다운로드 > 설치
 - ORACLE JDK : 기업용 유료
 - OPEN JDK : 무료
2. IDE (Integrated Development Environment : 통합개발환경) 툴 zip 다운 > 압축 풀고 사용
 - Eclipse development for java : 자바 개발 전용
 - Eclipse development for java web : 자바 + 웹 개발

자바로 만들 수 있는 것

1. 애플리케이션 (application) == 솔루션(solution)
: main() 메소드로 실행되는 자바 프로그램
pc에 설치하고 나서 사용하는 프로그램
2. 웹애플리케이션
: 웹서버에서 WAS (Web Application Server) 위에서 구동되는 웹 애플리케이션
main() 이 없음
3. 애플릿 (Applet)
: 웹 브라우저에서 실행되는 자바 프로그램
html 태그 중에서 <applet> 또는 <object> 태그로 실행시킴
main() 이 없음

라이브러리, 프레임워크, 애플리케이션 프로그래밍 인터페이스(API)

- 라이브러리는 애플리케이션의 특정 기능
- 프레임워크는 애플리케이션의 구조
- API는 사용자를 위해 이 둘을 한데 묶어주는 요소
- 프레임워크에는 라이브러리, 컴파일러를 비롯해 개발 프로세스에서 사용되는 다른 프로그램이 포함될 수 있습니다. 탄탄한 프레임워크가 갖춰져 있다면, 개발자는 도메인 또는 애플리케이션에서 반복 사용되는 코드로 인한 문제를 겪지 않아도 됩니다.

라이브러리

라이브러리란 컴퓨터 프로그램에서 빈번하게 사용되는 사전에 컴파일된 루틴 또는 리소스(클래스, 템플릿, 설정 데이터 등)를 모아둔 것입니다. 태스크를 실행해야 할 때마다 이 기능에 대한 코드를 작성하는 대신, 태스크를 수행해야 할 때마다 루틴이 사용됩니다. 리소스가 필요한 경우 프로그램은 해당 리소스에 대한 라이브러리를 호출합니다.

라이브러리란 자주 사용되는 로직을 재사용하기 편리하도록 잘 정리한 일련의 코드들의 집합을 의미합니다.

- 라이브러리는 자동차의 기능을 하는 부품을 의미합니다.

예를 들어, 자동차를 굴러갈 수 있게 하는 바퀴, 어두운 밤을 환하게 비출 수 있는 헤드라이트, 비 올 때 창문을 닦아주는 와이퍼 등이 라이브러리라고 할 수 있습니다.

프레임워크

프레임워크는 라이브러리의 컬렉션이라고 할 수 있습니다. 이 둘 사이의 주된 차이점은 제어의 역전, 즉 프레임워크에는 작업을 호출할 수 있는 제어권이 있다는 것입니다. 프레임워크는 필요한 템플릿 또는 프레임을 이미 보유하고 있기 때문에, 기존 항목을 기준으로 필요한 리소스를 호출할 수 있습니다(코드의 비어있는 영역을 채워야 함).

자바 프레임워크 : 스프링 (spring) 주로 사용

Gof의 디자인 패턴으로 유명한 랄프 존슨(Ralph Johnson)은 "프레임워크란, 소프트웨어의 구체적인 부분에 해당하는 설계와 구현을 재사용이 가능하게끔 일련의 협업화된 형태로 클래스들을 제공하는 것"이라고 정의하였습니다.

프레임워크 종류

구분	종류
자바 프레임워크	Struts, Spring, 전자정부 프레임워크
QRM 프레임워크	myBatis(iBatis), Hibernate
자바스크립트 프레임워크	AngularJS, React, Polymer, Ember
프론트엔드 프레임워크	Bootstrap, Foundation, MDL

프레임워크 장단점

• 장점

1) 효율적.

- 아무것도 그려지지 않은 제로에서 코드를 일일이 짜는 것보다 시간과 비용이 훨씬 절약되며 생산성이 좋아집니다.

2) Quality 향상.

- 버그 발생 가능성을 처리해줌으로써 개발자가 반복 작업에서 실수하기 쉬운 부분을 커버해줍니다. 다수의 개발자가 사용하며 수정하다 보니 이미 검증된 코드라고 볼 수 있습니다.

3) 유지 보수 Good!

- 프레임워크를 쓰지 않고 일일이 코드를 짜 놓은 경우, 회사 입장에서 개발 담당자가 바뀌어버리면 곤란해집니다. 그러나 Framework를 사용하면 코드가 보다 체계적이어서 담당자가 바뀌더라도 위험부담을 줄일 수 있으며 유지 보수에 안정적입니다.

프레임워크 장단점

• 단점

1) 학습시간이 길다.

- 코드를 본인이 짜 놓은 것이 아니기 때문에, 프레임워크에 있는 코드를 습득하고 이해하는 데 오랜 시간이 걸립니다.

2) 제작자의 의도된 제약 사항

- 제작자가 설계한 구조를 어느 정도 유지한 채 코드에 살을 붙여나가야 합니다. 따라서 개발자는 자유롭게 유연하게 개발하는 데 한계가 있습니다.

프레임워크는 단점이 존재하지만 단점을 커버할 수 있을 만한 좋은 장점이 있습니다. 개발의 상황과 목적에 맞는 프레임워크를 잘 파악하여 선택한다면 시간과 비용을 줄이는 것은 물론이고 코드의 품질이 훌륭한 개발을 할 수 있을 겁니다.

애플리케이션 프로그래밍 인터페이스(API)

프레임워크는 애플리케이션 프로그래밍 인터페이스(API)와 유사합니다. 기술적인 측면에서 프레임워크에는 API가 포함되어 있습니다. 프레임워크는 프로그래밍의 기반인 반면, API는 프레임워크가 지원하는 요소에 대한 액세스 권한을 제공합니다.

프레임워크 개발 환경 이해

- Java™ 프레임워크는 개발자가 Java 프로그래밍 언어로 애플리케이션을 만들기 위해 사용하는 미리 작성된 코드의 본문입니다.
- Java 프레임워크는 Java 프로그래밍 언어에만 있는 독특한 요소로, 소프트웨어 애플리케이션 및 Java 프로그램을 개발하는 데 사용되는 Java 플랫폼입니다.
- Java 프레임워크는 템플릿처럼 사용할 수 있는 미리 작성된 재사용 가능 코드의 본문으로, 개발자가 애플리케이션을 개발할 때 필요에 따라 사용자 정의 코드를 입력하여 쓸 수 있습니다.
- 프레임워크는 여러 번 사용할 수 있게 제작되었으므로 개발자는 모든 것을 처음부터 수동으로 작성하지 않고도 애플리케이션을 프로그래밍할 수 있습니다.
- Java 프레임워크에는 시스템 소프트웨어와 상호 작용할 뿐만 아니라 하드웨어 기기를 처리, 입력, 관리하는 데 사용할 수 있는 사전 정의된 클래스(예: 오브젝트의 카테고리)와 함수가 포함될 수 있습니다. 이는 프레임워크의 유형, Java 개발자의 기술 수준, 목표, 선호도에 따라 달라집니다.

소프트웨어 개발 방법론 1

- **반복 개발법(Iterative Development) :**

소프트웨어를 기능 단계 단위로 개발하는 것을 강조. 한 단계에서 완벽하게 작동해야만 다음 단계로 넘어감. 문제를 조기 발견가능. 위험이 큰기능은 초반에 위험이 낮은 기능은 후반에 일정을 잡으므로써 시간에 쫓겨 급하게 처리하게 되는 단점을 해결 가능.

- **스크럼(Scrum) :**

정기적으로 정해진 시간 동안 회의를 하며, 스크럼에서 정한 목표에 맞춰 몇 주 정도는 "스프린트"단위로 일을 진행. 협업을 강조/장려 하고 조기 피드백을 달성하고 속도를 유지하는데 있어서 효율적.

소프트웨어 개발 방법론 2

- **애자일 개발 절차(Agile Development) :**

반복 개발 모델을 제대로 된 소프트웨어 개발 방법으로 확정 시켜 놓은것. 일일 스크럼, 다부서 간 팀 구성, 매 반복 단위마다 완전한 시험 사이클 수행 등의 특징이 합쳐져 있음.

- **익스트림 프로그래밍(eXtreme programming) :**

애자일 개발 절차의 변종으로, 짝 프로그래밍, 테스트 주도 개발, 현장 고객 요건, 설계 리팩토링, 철저한 코드 검토 등이 주로 강조됨

- **린 개발법(Lean Development) :**

반복 개발법과 밀접하게 연관. "크게 생각하고, 작게 행동하고, 빨리 실패하면 신속하게 배울 수 있다"

소프트웨어 개발 방법론 3

- **폭포수 개발 절차(Waterfall Development Process) :**

명세서, 설계, 코드, 단위 시험, 등 선형적으로 쭉 이어져서 진행되는 정통적인 개발 모델

- **래피드 프로토타이핑(Rapid Prototyping) :**

강점과 약점을 미리 이해할 수 있도록 개념 검증용 프로토타입을 만들어야 한다는 아이디어를 바탕으로 만들어진 방법론