
컴퓨터 과학 101

구글리지 켄타

강사 소개

박용

- 구구컬리지 대표(교육 비영리단체)
- 전) AUD 개발 본부장(인공지능 서비스 총괄)
- 전) 삼성전자 무선사업부 시스템 소프트웨어 개발

목차

컴퓨터는 무엇으로 구성되어 있을까?

컴퓨터는 어떻게 소통할까?

프로그래밍 언어란?

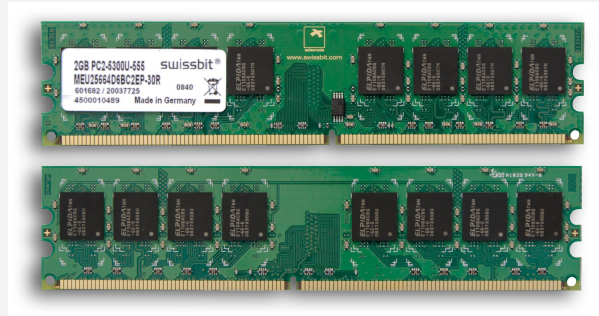
프로그래밍 개발 방법론

프로그래머의 역량

다양한 프로그래머 직군

무엇이든 물어보세요

컴퓨터는 무엇으로 구성되어 있을까?



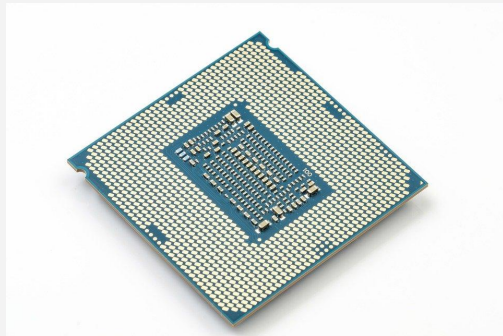
CPU

중앙처리장치 - 연산

클럭 속도 - 높을 수록 빠른 연산

코어 수 - 많을 수록 빠른 연산

인텔 5.2GHz, 16 cores



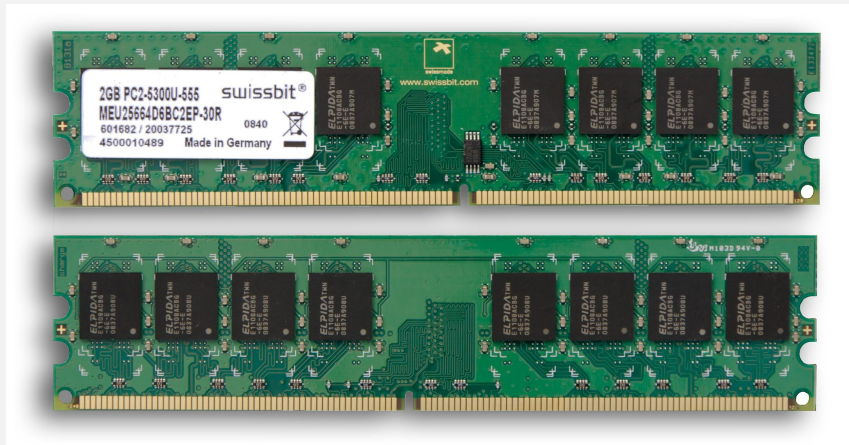
RAM(메모리)

주기억장치

용량 - 8GB, 16GB

프로그램이 실행되는 곳

컴퓨터가 꺼지면 데이터가 사라짐



SSD, HDD

보조기억장치

데이터를 저장하는 곳

HDD는 물리적 방식으로 데이터를 저장하기 때문에 느림



보조장치

입력과 출력 도구



GPU

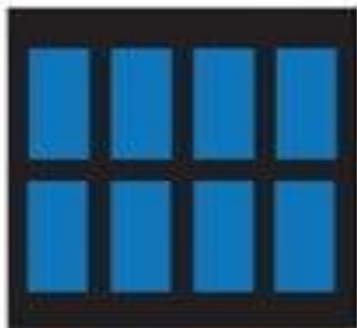
그래픽 처리 장치, 그래픽 연산에 특화된 장치

단순한 대량 작업에 특화되어 있음

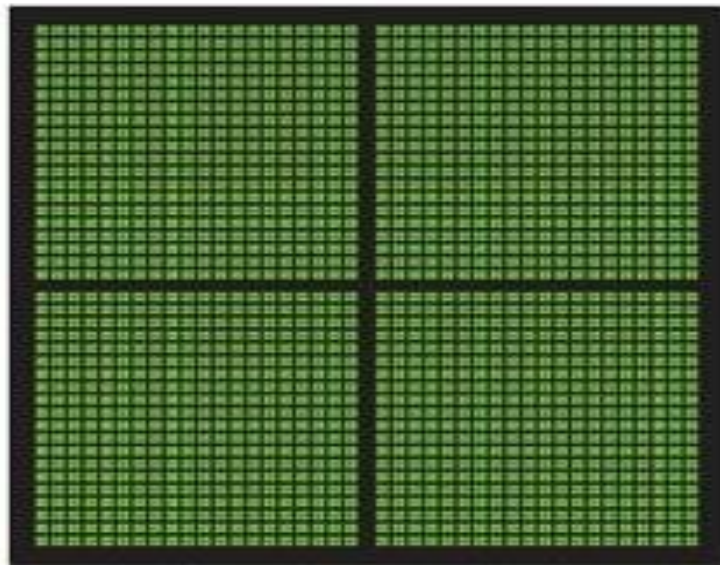
대용량 데이터 처리 -> 인공지능, 빅데이터, 블록체인

예) RTX3060 - 3584개 코어

CPU 와 GPU 비교



CPU
MULTIPLE CORES



GPU
THOUSANDS OF CORES

컴퓨터와 어떻게 소통할까?

```

40
41 $(function){cards();});
42 $(window).on('resize', function(){cards();});
43 function cards(){
44     var width = $(window).width();
45     if(width < 750){
46         cardssmallscreen();
47     }else{
48         cardsbigscreen();
49     }
50 }
51 function cardssmallscreen(){
52     var cards = $('<div>card</div>').length;
    var height = 0;
    card2 = 2;
    i = 1; i<=cards; i++){
        i = $('<div>card</div>').length;
    }

```


기타

```
[ ~ ]$ hexdump -C binary_file_example
00000000  41 00 41 00 00 00 42 00  42 00 00 00 43 00 43 00  |A.A...B.B...C.C.|
00000010  00 00 44 00 44 00 00 00  45 00 45 00 00 00 46 00  |..D.D...E.E...F.|
00000020  46 00 00 00 47 00 47 00  00 00 48 00 48 00 00 00  |F...G.G...H.H...|
00000030  49 00 49 00 00 00 4a 00  4a 00 00 00 4b 00 4b 00  |I.I...J.J...K.K.|
00000040  00 00 4c 00 4c 00 00 00  4d 00 4d 00 00 00 4e 00  |..L.L...M.M...N.|
00000050  4e 00 00 00 4f 00 4f 00  00 00 50 00 50 00 00 00  |N...O.O...P.P...|
00000060  51 00 51 00 00 00 52 00  52 00 00 00 53 00 53 00  |Q.Q...R.R...S.S.|
00000070  00 00 54 00 54 00 00 00  55 00 55 00 00 00 56 00  |..T.T...U.U...V.|
00000080  56 00 00 00 57 00 57 00  00 00 58 00 58 00 00 00  |V...W.W...X.X...|
00000090  59 00 59 00 00 00 5a 00  5a 00 00 00
0000009c  |Y.Y...Z.Z...|
```

이진수

숫자를 0과 1로 표현하는 숫자체계

0, 1, 10, 11, 100, 101, 110 (0, 1, 2, 3, 4, 5)

반도체와 이진수와 컴퓨터

반도체는 도체와 부도체의 성질을 둘 다 가짐

전기가 흐를 때 도체 $\rightarrow 1$

전기가 흐르지 않을 때 부도체 $\rightarrow 0$

반도체와 이진수와 컴퓨터

반도체가 두개 있다고 가정하면 4개의 명령을 내릴 수 있음

00 은 모니터를 끄

01 은 모니터를 켜

10 은 스피커를 끄

11은 스피커를 켜

다양한 프로그래밍 언어

파이썬, 자바스크립트, C, C++, C#, Go,
Visual Basic, B, R, Rust ...

지금도 프로그래밍 언어는 계속 나오는 중...

무엇을 써야 할까?, 무엇을 배워야 할까?

왜 이렇게 인기가 많은 걸까?

프로그래밍 언어마다 특징이 있음, 장단점이 존재함

C - 기계에 대한 직접 접근이 가능함, 실행이 빠름, 임베디드 또는 시스템 소프트웨어를 만들 때 많이 사용함

자바, C#, C++ - OOP를 기반으로 제작된 프로그래밍 언어, 게임, 응용소프트웨어

파이썬 - 스크립트 언어이고 익히기 쉽고 벡터 방식의 데이터 형태를 다룰 때 많은 라이브러리를 기본 연산으로 가지고 있음, 통계, 인공지능 등에 사용

자바스크립트 - 웹페이지 인터랙션 기능을 제작할 때 사용하는 언어

그래서 우리는 어떤 프로그래밍 언어를 배워야 하죠?

프로그램 종류

운영체제와 응용프로그램

운영체제: 시스템을 관리하고 응용프로그램을 실행하는 프로그램

- 윈도우, 맥OS, 리눅스

응용프로그램: 브라우저, 엑셀 등

- 운영체제에 종속됨

프로그래밍 개발 방법

전통적인 개발 프로세스

기능 명세

제품 기능에 대한 설명과 요구사항 분석과 작성

개발

각 기능 개발 및 구현

테스트

통합, 부하 테스트

배포 및 런칭

시장 또는 클라이언트에게 배포 및 런칭

문제점

사용자가 정말 원하는 제품을 모른다

개발자가 만든 제품이 사용자가 원했던 제품이 아니다

다 만들었더니 둘 다 ...

결국 아무도 쓰지 않는 제품을 열심히 만들게 됨

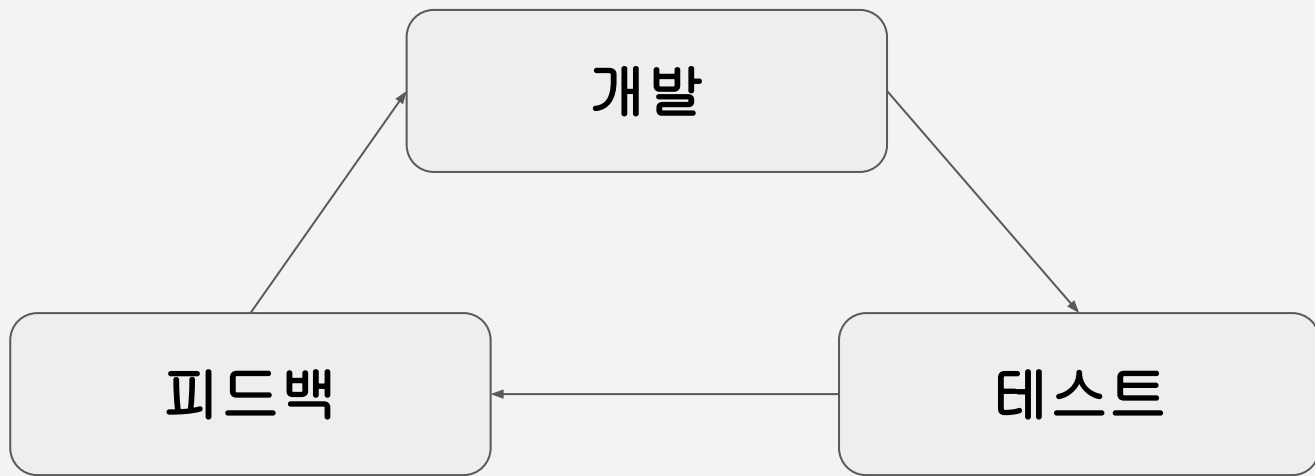
점진적인 개발

모든 기능을 한 번에 구현하는 것이 아닌

점진적으로 기능을 개발하는 방법

- 사용자 회원가입, 로그인 기능 구현 후 클라이언트 피드백 및 테스트를 진행함
- 좋아요 기능을 구현 후 클라이언트 피드백 및 테스트를 진행

점진적인 개발



점진적인 개발

사용자 참여를 통해 기능 단위로 피드백을 받기 때문에 원하는 제품을 만들 수가 있음

- 소통에 대한 비용이 올라감
- 기능, 요구사항 수정사항이 많음
- 애자일, 스크럼 등등

테스트 주도 개발(TDD)

개발방법론

기능을 개발하고 테스트를 진행하는 전통적인 방법과 다르게

테스트 코드를 먼저 만들고 테스트를 통과하는 코드를 작성하는 방법

TDD 예시

나누기를 개발한다고 했을 때

나누기 기능을 실행했을 때
통과 또는 실패를 표시하는
테스트 코드를 우선 개발
예) 0으로 나눴을 때 에러가
나오는 지 검사

테스트 코드를
통과할 수 있게 구현

TDD 장점

테스트가 개발과 함께 이루어 지기 때문에 언제나 테스트가 끝난
기능을 확보할 수 있음

테스트 자동화

코드 변경에 대한 개발자 스트레스를 줄여줌

Pair Programming

두 명의 프로그래머가 한조를 이루어서 번갈아가면서 프로그래밍하는 방법(짝 프로그래밍)

- 프로그래밍에 대한 정보 공유
- 팀간의 프로그래밍 실력 및 격차를 줄여줌

프로그래머의 역량

어떤 코드가 좋은 코드일까?

기준

정확성 - 목표한 대로 기능하는가

성능 - 소스코드의 크기, 메모리, CPU의 사용량, 실행속도 등

가독성 - 다른 사람이 보기에 읽기 쉬운 코드인가

유지보수 - 변경과 확장이 용이한가

현업에서 능력

소통과 협업

팀단위의 프로그램 개발

소통과 협업은 매우 중요한 능력

- 출을 지키는 코드 작성(코드 컨벤션)
- 가독성 있는 코드 작성
- 변경과 확장성 있는 코드
- 코드리뷰

어떻게 실력을 늘릴까?

어떤 실력을 늘릴 것 인가?

다양한 종류의 개발을 하고 싶은가?

프로그래밍 작성 실력을 늘릴 것인가?

새로운 프로그래밍 언어를 배울까?

알고 있는 프로그래밍 언어에 대해 깊게 알아갈까?

커뮤니티, 토이 프로젝트

커뮤니티를 최대한 이용

- 최신 내용은 책이나 영상으로 얻을 확률이 높음
- 시니어 레벨의 내용은 경험자를 통해서!
- 이직 제안도 많이 받음

토이 프로젝트

- 프로그래밍은 이론이 아닌 실전
- 실제 프로그래밍으로 제품을 만들어봐야함!
- 실 배포와 운영을 해보면 더욱 더 좋음
- 운 좋으면 CEO가 됨

오픈소스

제품의 코드가 공개되어 누구나 코드를 볼 수 있고 수정할 수 있음

유명한 오픈소스 프로젝트 : 리눅스, 안드로이드, 아파치, 마리아 DB
등등

코드가 공개되어 있기 때문에 유명한 프로젝트의 경우, 작성 규칙
또는 최적화 방법 등 배울 것이 많음

프로그래머 직군

어플리케이션에 따른 직군

웹 어플리케이션 - 웹에서 동작하는 어플리케이션(웹사이트)

응용 어플리케이션 - 운영체제에서 동작하는 어플리케이션(엑셀, 워드)

임베디드 소프트웨어 - 기기위에서 동작하는 소프트웨어(운영체제 없이), IoT 등

앱 어플리케이션 - 모바일에서 동작하는 어플리케이션

인공지능, 블록체인, 게임 등

다른 직군들

DBA, 데이터 엔지니어

테스터

인프라 관리

기타 IT 용어 설명

클라우드

로컬에서 프로그램을 설치하고 실행하는 것이 아닌

온라인에서 프로그램을 실행하는 것

애저, 팀즈, 오피스365, 구글 워크스페이스, AWS 등

- 데이터 공유와 협업이 편리 해짐
- 시간 단위로 사용비용을 내기 때문에 가격에 대한 장점
- 리소스 관리에 대한 부담감이 적음
- IaaS, PaaS, SaaS

무엇이든 물어보세요

조별로 연구 및 발표자료를 작성

키워드

컴파일러

웹 프론트
엔드

코드 리뷰

웹 백엔드

구글
애널리틱스

버전 관리

UI, UX

IaaS

데이터베이스
종류

웹 어플리케이션
구성요소

IDE

스크립트
언어란?