

# 컴퓨터 과학의 역사와 철학 그리고 핵심원리들

## - 식사하는 철학자

- ✓ 컴퓨터 과학의 전체 그림을 그려보기[복기]
- ✓ 0과 1에서 컴퓨터가 탄생하기 까지[복기]
- ✓ 컴퓨터와 우리와의 대화[복기]
- ✓ 프로그래밍의 핵심개념 (변수, 제어문, 함수, 프로그래밍 패러다임)
- ✓ Github의 포크와 브랜치 실습과 이해
- ✓ Github에 HTML 페이지 퍼블리싱 하기
- ✓ DNS 원리 이해하고 도메인 구매
- ✓ CSS 코드의 재사용 연습하기
- ✓ UI 라이브러리 사용하기 (Semantic UI)

## 배운 것들에 대하여 생각해보기(5주간)

컴퓨터과학의 전체 역사  
최초의 컴퓨터가 탄생하기까지의 생각  
HTML(hyper Text)  
모르는 태그를 찾는법  
CSS 기본 구조  
CSS 선택자  
CSS Box모델  
CSS 레이아웃

Github  
Git 기본명령  
협업하는 법

Atom 에디터

세계 최초의 프로그래머

찰스베베지 해석기관 – 미분기(입력/연산/출력/저장/제어)  
과의 만남  
시적인 과학자 에이다 러브레이스(Ada Lovelace)

Loop  
Goto  
If  
의 개념을 최초로 생각해냄

프래그래밍이란 무엇일까요?

어떤 목표를 이루기 위한 사전에 계획  
예) TV 편성표



변수란?

값을 담는 그릇이다.

```
if (true) { var x = 5; } console.log(x); // 5
```

```
if (true) { let y = 5; } console.log(y); // ReferenceError: y is not defined
```

변수에 대하여 적어보세요?

[https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Values,\\_variables,\\_and\\_literals#%EB%B3%80%EC%88%98\\_%EB%B2%94%EC%9C%84](https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Values,_variables,_and_literals#%EB%B3%80%EC%88%98_%EB%B2%94%EC%9C%84)

## 비교 연산자와 참과 거짓

참과 거짓의 상태를 만드는 것

```
console.log(Boolean(''));  
console.log(Boolean('0'));  
console.log(Boolean(0));  
console.log(Boolean(1));  
console.log(Boolean([]));  
console.log(Boolean(undefined));  
console.log(!!undefined);
```

비교 연산자는 무엇이고, 어떤 일을 할 수 있을까요?

<https://www.javascripture.com/Boolean>

## 조건문 제어문

상태에 따라 다른 길을 갈 수 있다.

```
if (condition_1) {  
    statement_1;  
} else if (condition_2) {  
    statement_2;  
} else if (condition_n) {  
    statement_n;  
} else {  
    statement_last;  
}
```

조건문 할 수 있는 것.

<https://opentutorials.org/course/3085/18800>

[https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Control\\_flow\\_and\\_error\\_handling](https://developer.mozilla.org/ko/docs/Web/JavaScript/Guide/Control_flow_and_error_handling)

## 함수

재사용 가능한 코드를 만들기 위해 코드  
함수는 매개변수를 입력받고 값을 반환.

```
function square(number) {  
    return number * number;  
}
```

코드를 재 사용하는 이유

## 최초의 컴파일러를 만든 사람 – 그레이스 호퍼(Grace Brewster Murray Hopper)

디버깅의 대중화  
컴파일러 구현

“나에게 프로그래밍은 실용적 기술 그 이상의 의미가 있다.  
프로그래밍은 지식의 근본을 탐구하는 대규모 프로젝트다.”

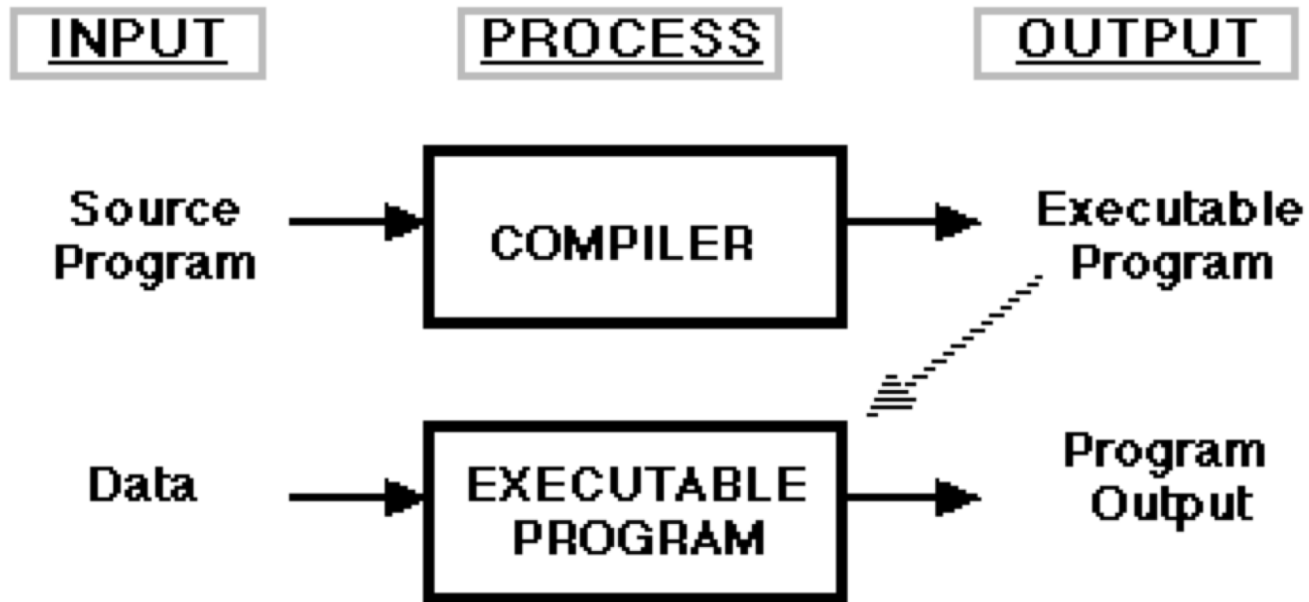
여러운 말을 쉬운 말로 번역해보세요. 그게 컴파일러다.





## 컴파일러

컴파일러는 프로그램을 만드는 프로그램



<https://www.cs.uaf.edu/~cs631/node4.html>

## C언어와 발명과 유닉스 시스템 개발

### Hello World – 데니스 리치 (Dennis MacAlistair Ritchie)

모든 기계에서 동일하게 동작하는 컴퓨터 언어.

C언어를 개발하고 C언어로 유닉스를 다시 제작.

유닉스는 현대 운영체제의 모형.

운영체제란?

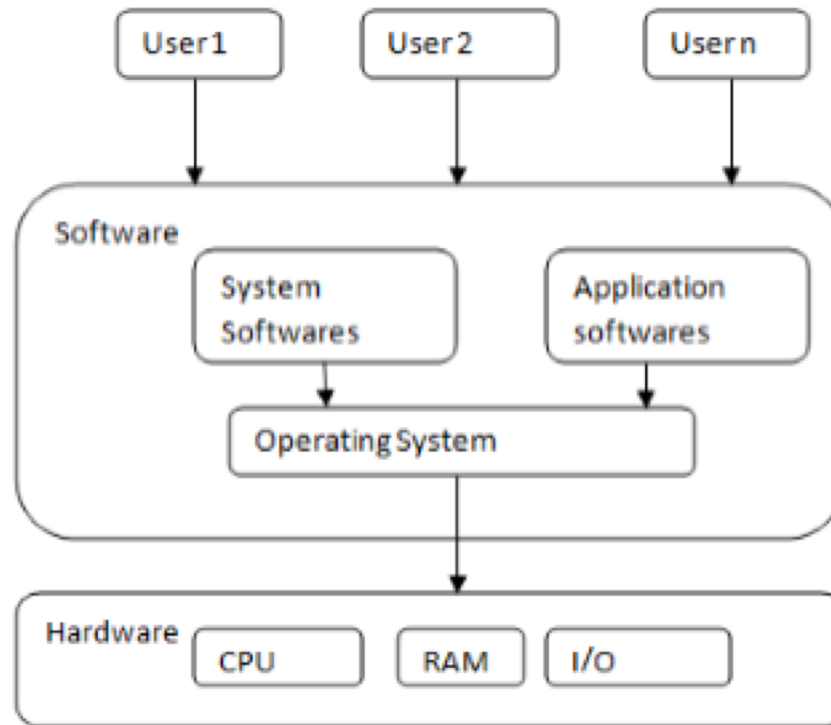
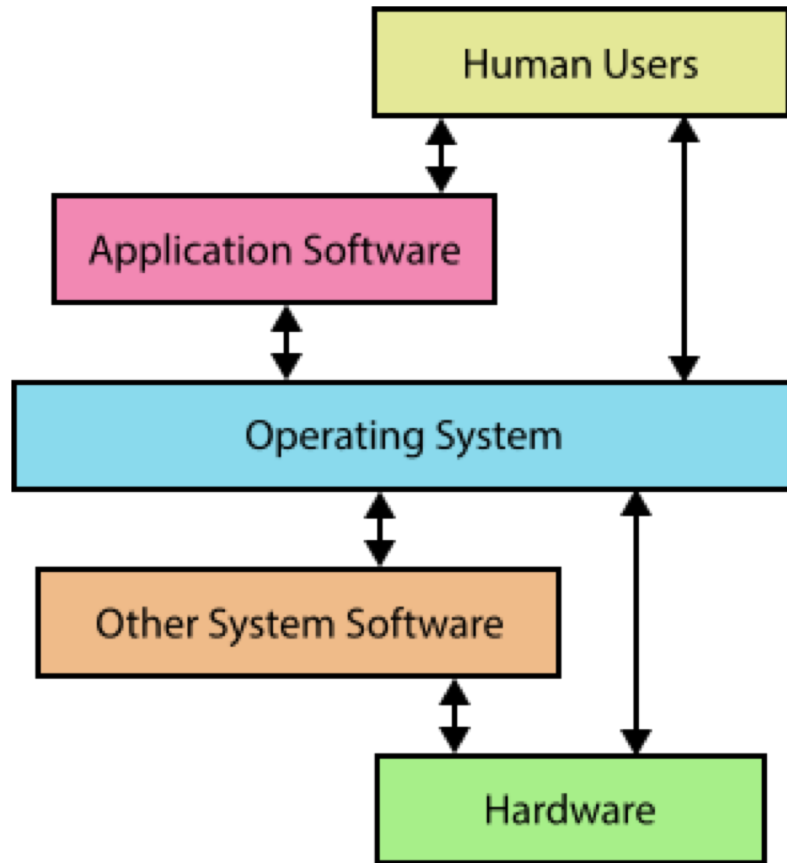


<http://it.donga.com/20992/>

<https://joone.net/2017/08/29/10-%EB%8D%B0%EB%8B%88%EC%8A%A4-%EB%A6%AC%EC%B9%98%EC%9D%98-%EC%9D%BC%EC%83%9D/>

[https://ko.wikipedia.org/wiki/%EB%8D%B0%EB%8B%88%EC%8A%A4\\_%EB%A6%AC%EC%B9%98](https://ko.wikipedia.org/wiki/%EB%8D%B0%EB%8B%88%EC%8A%A4_%EB%A6%AC%EC%B9%98)

## 운영체제의 기능



프로세스 관리  
CPU 스케줄링  
메모리 관리  
파일 관리  
입출력장치 관리  
다중프로그래밍 시스템

[https://commons.wikimedia.org/wiki/File:Role\\_of\\_an\\_Operating\\_System.svg](https://commons.wikimedia.org/wiki/File:Role_of_an_Operating_System.svg)

<http://www.ques10.com/p/4418/explain-objectives-functions-of-os-1/>

## 깃과 리눅스를 개발한 리눅스 토발츠

### 리눅스 토발츠 (Tim Berners-Lee)

리눅스 커널 개발-1991년,  
리눅스 커널 점유율: 퍼블릭 클라우드 90%, 스마트폰 82%, 슈퍼컴퓨터 99%  
리눅스 1400여 개 회사, 15,637명의 개발자 리눅스 커널 개발 동참

오픈소스가 소프트웨어 개발분야에서 성공했던 이유.  
사람들은 왜 오픈소스로 개발을 할까요?



<https://www.linuxfoundation.org/2017-linux-kernel-report-landing-page/?SSAID=389818>

[https://ko.wikipedia.org/wiki/%EB%A6%AC%EB%88%84%EC%8A%A4\\_%ED%86%A0%EB%A5%B4%EB%B0%9C%EC%8A%A4](https://ko.wikipedia.org/wiki/%EB%A6%AC%EB%88%84%EC%8A%A4_%ED%86%A0%EB%A5%B4%EB%B0%9C%EC%8A%A4)

# 리눅스의 기본 철학 영상

[https://www.ted.com/talks/linus\\_torvalds\\_the\\_mind\\_behind\\_linux/transcript?utm\\_campaign=&awesm=on.ted.com\\_sJsS&utm\\_content=addthis-custom&source=twitter&utm\\_source=direct-on.ted.com&utm\\_medium=on.ted.com-twitter&language=ko#t-1207394](https://www.ted.com/talks/linus_torvalds_the_mind_behind_linux/transcript?utm_campaign=&awesm=on.ted.com_sJsS&utm_content=addthis-custom&source=twitter&utm_source=direct-on.ted.com&utm_medium=on.ted.com-twitter&language=ko#t-1207394)