


Silicon Valley Coding

Lecture 1

시작하기전에.. wifi: svcoding, svcoding5G password: svcoding2016

- 아톰(Atom editor)을 다운로드 받으세요. (www.atom.io)
 - Atom같은것들을 코드 에디터(*code editor*)라고 합니다. Code를 작성할때 쓰지요.
- 크롬(Chrome)을 다운로드 받으시고 설치해주세요.
- 크롬(Chrome)을 기본 브라우저로 설정해주세요.
 - 크롬(Chrome) 브라우저 우상단의  아이콘 클릭 -> 설정(Setting) -> 기본브라우저 설정
- Google 계정이 없으시면 하나 만들어주세요. (gmail있으시면 계정이 있으신거예요)
 - 개발자로서 google계정은 필수입니다. 없으시면 gmail계정을 만들어주세요. (gmail.com)
- <https://goo.gl/kYpKgW> 로 가서 gmail, 이름, 전화번호를 입력하고 제출하세요.
- 위에서 입력하신 Gmail로 수업 시작전에 Slack 초대를 보내겠습니다. 수업시작전 Gmail을 열고 초대받은 링크를 클릭하여 Slack에 가입하면 됩니다. Slack은 개발자용 카카오톡 이라고 보시면 됩니다.

성공적으로 코스를 마치기위한 팁

- 모르는것이 있으면 언제든지 물어보세요 - 질문은 필수!
- 구글/네이버 검색을 통해 답을 찾아보는 것에 익숙해지세요.
- 완벽하게 코드를 이해하지 않았다면 절대 복사 붙여넣기(복붙)하지 마세요.
- 인내심을 기르세요. 개발에는 인내심도 능력입니다.(컴퓨터 던지고 싶으실정도로 답답할때가 많은게 프로그래밍 입니다)
- 괄호안 영어에 익숙해지세요! 한번 용어 정리 후 영어 용어를 사용합니다. 모르시면 복사하시고 slack에서 물어보세요.
 - 값(value), 속성(attribute), 성질???(property), 모국어??? (native language)
- 키보드 단축키에 익숙해지세요.
 - 우클릭후 복붙하는 행동은 별로 좋은 습관이 아닙니다. *cmd + c* 와 *cmd + v*를 사용해주세요. (PC인 경우 cmd 대신 ctrl입니다)
- 컴퓨터에 최소한의 창들만 열어주세요! 사용 안하는 탭은 닫아주세요. 생각보다 중요한 습관입니다.

Computer 언어(Language) 레벨(Level)이란

- 기계어 (Machine Language)

- 1과 0(이진수)으로만 쓰여지는 언어. 컴퓨터가 보고 사용하는 언어와 가장 가까운(lower-level) 언어.

- 어셈블리 언어 (Assembly Language)

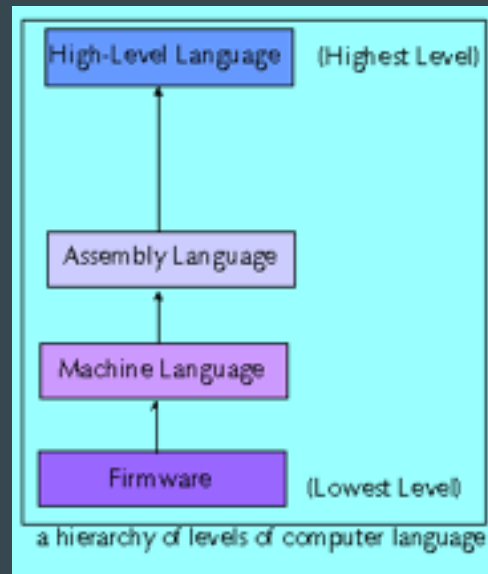
- Machine code를 조금더 읽기 쉽게 0과 1이 아닌 알파벳과 숫자를 이용한 언어.

- 컴파일러 언어(Compiler Language)

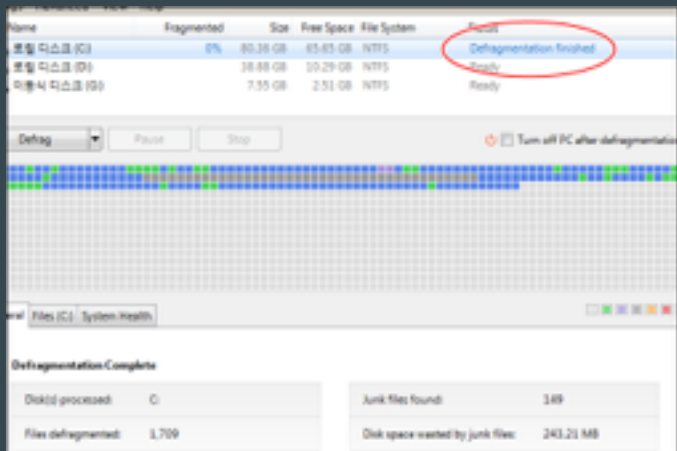
- 기계어(Machine Code)보다 멀리있는(high-level) 어셈블리 언어(assembly language)
- Machine Code로 컴파일(compile)후 구동함.
- C, C++, Go, Fortran, Pascal, Java

- 인터프리터 언어(Interpreter Language)

- 기계어보다 멀리있는 언어(high-level language)—기계어로 통역해주는 interpreter가 메모리에 구동되어 compile될 필요가 없는 언어.
- JavaScript, Python, PHP, Ruby

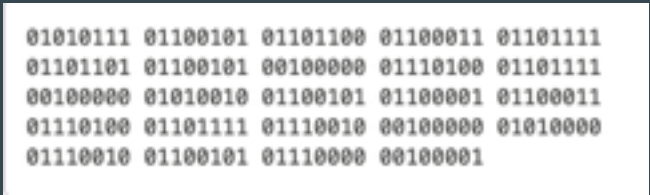


메모리_블럭(Memory Blocks),

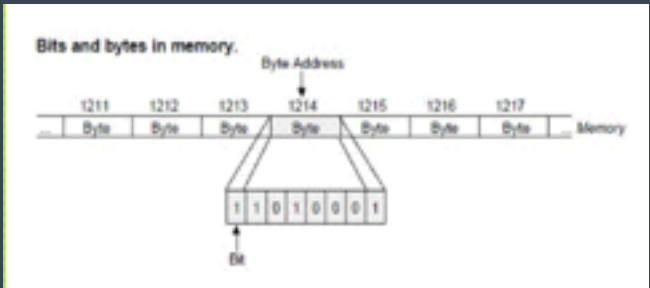
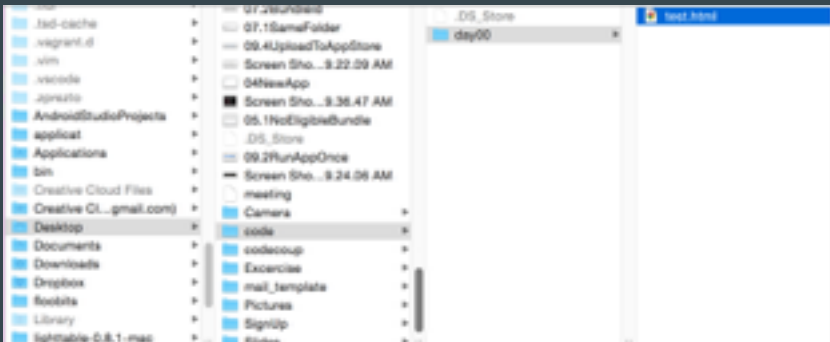


```
cd ~/Desktop
mkdir code
cd code
mkdir day00
cd day00
touch test.html
```

Interpreter가
파싱(parsing)



File System이
렌더링(rendering)



프로그래밍 이란 무엇인가?

컴퓨터가 실행할 명령어들의 모임을 만드는것.

명령어 예)

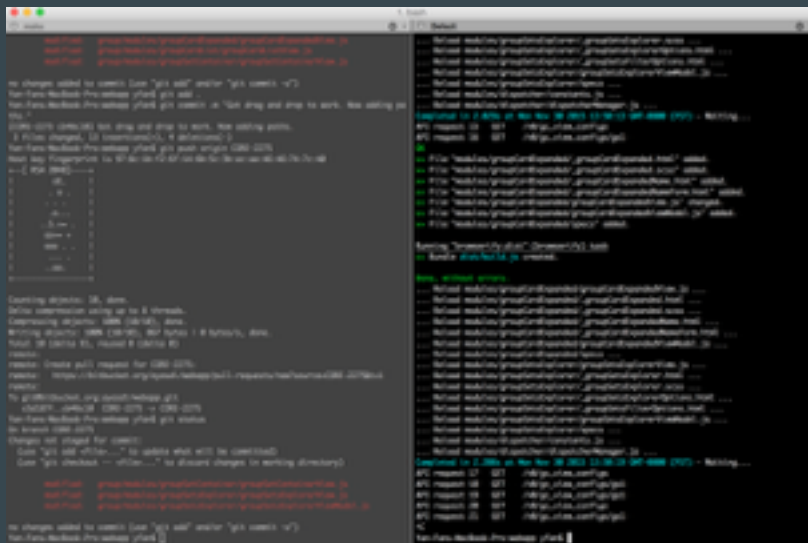
- | | |
|--|--|
| 1. 바탕화면(Desktop) 폴더로 들어가라 명령, | <code>`cd ~/Desktop`</code> |
| 2. <code>`code`</code> 폴더를 만들어라 명령, | <code>`mkdir code`</code> |
| 3. <code>`code`</code> 폴더로 들어가라 명령, | <code>`cd code`</code> |
| 4. <code>`day00`</code> 이라는 폴더를 만들어라 명령, | <code>`mkdir day00`</code> |
| 5. <code>`day00`</code> 이라는 폴더로 들어가라 명령, | <code>`cd day00`</code> |
| 6. <code>test.html</code> 이라는 파일을 만들어라 명령, | <code>`touch test.html`</code> |
| 7. 글을 <code>test.html</code> 에 쓰라 명령, | <code>`echo 이것은 글입니다 >> test.html`</code> |
| 8. Chrome브라우저에서 열라 명령, | <code>`open -a "Google Chrome" test.html`</code> |

```
1  function 명령어모음 {
2
3      cd ~/Desktop
4      mkdir code
5      cd code
6      mkdir day00
7      cd day00
8      touch test.html
9      echo 이것은 글입니다 >> test.html
10     open -a "Google Chrome" test.html
11
12 }
```

CLI vs. GUI: 컴퓨터(OS)에게 명령(command)을 내릴수 있는 2가지 방법.

CLI = command line interface

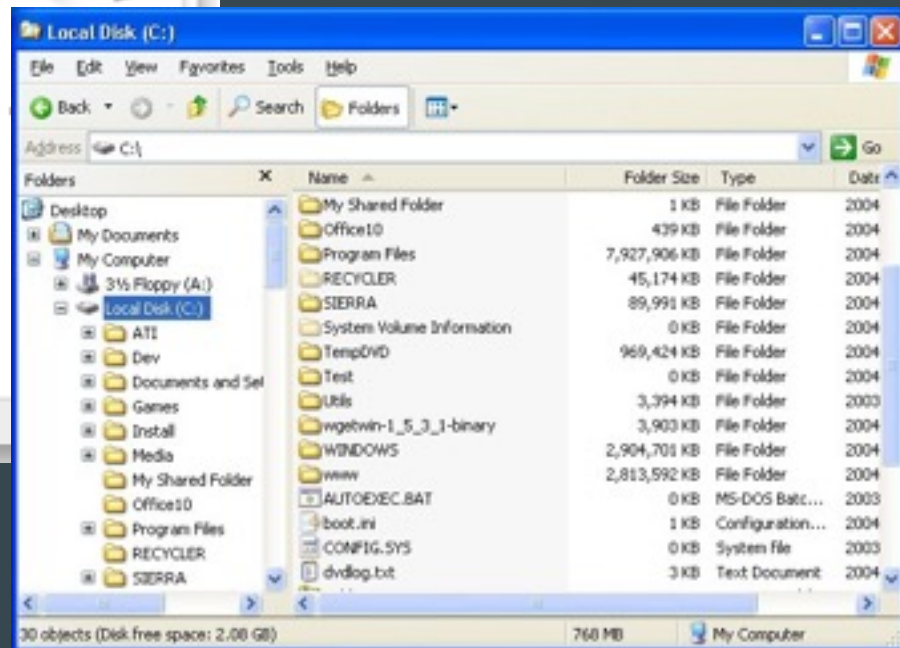
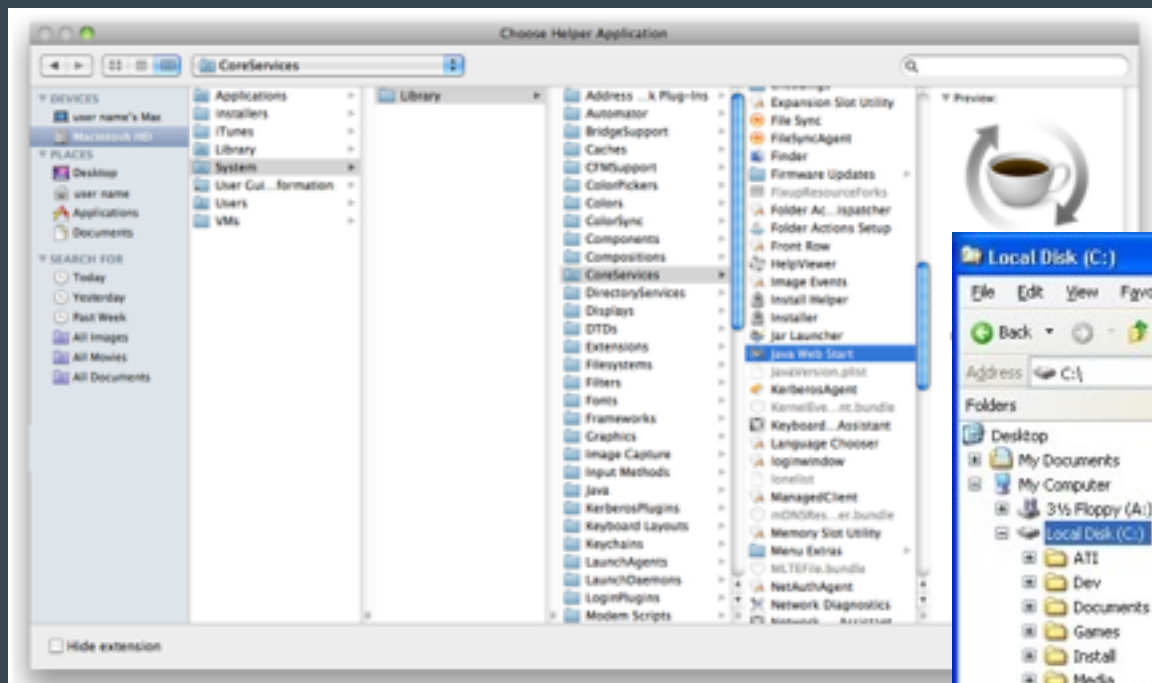
GUI = graphical user interface



CLI, GUI 발음?

스큐모픽(skeuomorphic) 아이콘이란?

파일 시스템(File Systems)



Exercise: 처음부터 읽고 exercise를 하세요.

무작정 따라하면 못배웁니다!! 항상 생각하면서 모르는것 있으시면 질문 하세요.

일부로 틀리게 써 있는 부분도 있습니다. 혼자서 “왜 안되지/없지?” 보다 바로 질문 부탁드립니다.

목표: 서식 구현법 이해

<https://svcoding0.firebaseio.com/index.html>

마크업 언어 (Markup Languages)

- 문서에서 주석을 달고, 글에 스타일을 주는 시스템/간단한 언어. (No 마우스클릭)
- 특별한 기호나 단어를 사용하여 글의 서식과 스타일을 줍니다.
- .md 확장자로 저장합니다. 예) intro_markdown.md
- Markup Language중 가장 간단한 언어인 마크다운(Markdown) language를 배우도록 하겠습니다.

해더 - 제목 텍스트

****굵은 서식 적용된 글****

이탈릭체(italic) 적용된들

- - -

Exercise:

목표: 마크다운 서식 기호 이해

저희 Slack주소.

<https://svcsample.slack.com/>

“HTML이란 무엇인가?” 섹션 전까지

HTML: 하이퍼텍스트(HyperText) Markup Language”

HTML이란:

- 태그(tag)들로 마킹되어있는 텍스트(text) 파일.
- .html 확장자(extension)로 저장됨.
- 웹브라우저의 “모국어(Native language)”
- 웹페이지의 내용과 구조(structure)를 만드는데 사용

HTML은:

- 시멘틱(Semantic) 하고
- 접근성(Accessible) 있으며
- 잘 정돈(Organized)되어 있어야 합니다.

HTML tag는 어떻게 생겼나요?

tag들은 시작(start) tag와 마무리(end) tag가 있고, 내용과 속성(attribute)을 가지고 있습니다.

```
<tagname attribute="값(value)">
  내용(content)
</tagname>
<a href="http://www.google.com" id="google" class="link">
  Google
</a>
```

a = anchor tag

href = hypertext reference (attribute의 좋은 예입니다)

attribute는 거의 대부분 선택사항(optional)입니다.

HTML 구성요소(Components)

```
<!DOCTYPE html>
<html>
  <head>
    <title>페이지 제목</title>
  </head>
  <body>
    <h1>나의 첫번째 헤딩</h1>
    <p>나의 첫번째 짧은 문장</p>
  </body>
</html>
```

```
<body></body>
```

body tag안에 들어가 있는 부분들만 브라우저에서 보통 보이는 부분들입니다.

```
<!DOCTYPE html>
```

독타입(doctype)은 tag가 아닙니다. doctype은 모든 HTML page가장 첫줄에 쓰여 있어야 브라우저가 어떠한 HTML 버전을 사용하고 있는지 해석할수 있습니다. 지금 같은 경우 HTML5를 사용하고 있습니다.

```
<html></html>
```

html tag는 페이지에서 항상 첫번째 tag입니다.

```
<head></head>
```

head tag안에는 메타(meta) 정보가 들어있습니다.
— 브라우저가 페이지를 로딩하기전에 필요한 정보들입니다.

body안에 무엇이 들어가나요?

<h1>텍스트, 해더</h1>

<h2>조금더 작은 해더</h2>

<h3>더 작은 해더</h3> ← h6 까지 tag가 있습니다.

<p>문장</p> ← p tag는 브라우저에서 정해놓은 기본 폰트 크기 (font-size)를 사용합니다.

**줄바꿈(Line breaks)
** ← closing tag가 없는 tag들도 존재합니다.

강조된(Emphasized) 텍스트 그리고

굵은(Bolded) 텍스트는 해더나 문장과는 다르게 같은 줄에서 사용할수 있습니다.

body안에 무엇이 들어가나요? 구조(Structured)를 가지고 있는것들.

**** ← 순서없는 목록(unordered lists)

**** ul안에 들어있는 목록들은 글머리기호(bullet) 스타일을 가지게 됩니다. ****

**** 목록 아이탬은 li tag안에 들어가야 합니다. ****

**** 목록 아이탬은 원하시는 만큼 만드셔도 됩니다. ****

**** ← 순서있는 목록(ordered lists)

**** ol tag안에 있는 목록 아이탬들은 번호가 매겨집니다. ****

**** 목록 아이탬은 li tag안에 들어가야 합니다. ****

<div></div> ← div tag들은 정말로 자주 사용되는 tag들입니다.

body안에 무엇이 들어가나요? 링크와 여러가지것들.

`` ← 컴퓨터 폴더안에서 그림 찾을

`` ← 링크 주소에서 그림 찾을

`Google로 가는 링크`

`현재 웹사이트의 About페이지로 가는 링크`

`<p> 첫번째 장으로 넘어가기! </p>`

`<h1 id="chapter1">첫번째 장이라고 알려주는 내부 링크</h1>`

`새창에 구글웹페이지 열기`

`<!-- 이것은 커멘트(comment)입니다. 이안에 있는 글들은 브라우저에서 보이지 않습니다. -->`

MDN은 HTML에 정보를 가장 잘설명해 놓은 곳입니다.

(그럼에도 읽기 힘든 부분이 많습니다.)

tag들 목록: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

개발자 가이드: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML>

HTML 팁

- tag들을 다 닫으세요! ((end tag가 없는 것들도 있습니다. 예시)
))
- 중첩됨(nesting)을 보이기 위해 들여쓰기(indent)를 해주세요.
- 소문자로 써주세요.
- 읽기 편하게 써주세요!

```
<ul><li><a href="http://www.google.com">Google</a></li>
<li>Facebook</li></ul>
```

NONONONONONO

```
<ul>
  <li><a href="http://www.google.com">Google</a></li>
  <li>Facebook</li>
</ul>
```

YESYESYESYESYES

Block vs. Inline элемент(Elements)

- element별로 서로 다른 element끼리 상호작용하는 방식들이 기본적으로 설정되어있습니다.
 - Block: tag들은 한줄을 전부다 차지합니다.
 - Inline: tag들은 같은줄에 섞여 있을수 있습니다.
- <div>는 특별한 tag로 무엇이든 할수있습니다. (기본적으로 block 속성을 가지고 있지만)
- , , 와 같은 테그들은 inline속성을 가지고 있습니다.
- <h1> 와 다른 heading tag들은 block속성을 가지고 있습니다. 그리고 이것들 안에 다른 block 속성 tag들을 nesting하는것은 좋은 방법 (best practice)이 아닙니다.
- 자세한 내용은 다음 학습시간에 배우겠습니다.

~~좋은 방법(best practice)~~

```
<div>
  <h1>div안에 tag들을 nesting할수 있습니다</h1>
  <div>
    <p>div안에 div를 nesting해도 됩니다</p>
  </div>
</div>
<div>
  <p>
    이런 tag들을 <em>nesting</em>하는데
    아무런 <b>문제 없습니다</b>
  </p>
</div>
```

~~정말 나쁜 방법 (Terrible practice)~~

```
<h1>
  <div>
    이렇게 쓰는것은 bad practice입니다
  </div>
  <h4>HTML쓰는데는 여러 룰이 있습니다</h4>
</h1>
<h3>그러나 쓰다보면
  <h2>저절로 알게 되는 쉬운 룰들입니다</h2>
</h3>
```

Exercise:

목표: 브라우저 입장에서의 HTML이해,
서버에 웹페이지 올리기

Sample Ends

재미 있으셨나요?

이 Sample강의는 실리콘밸리 코딩 기초 8주과정 코스에서 첫날 배우는 내용입니다.

저희 8주 기초과정과 12주 심화과정 코스에 관심이 있으시면 아래 번호로 연락주세요.

010-9178-8743

02 569 0714

www.svcoding.com