

공부에 도움되는 자료들 - 첫번째 스터디

노트북: 컴퓨터 프로그램의 구조와 해석
만든 날짜: 2018-02-01 오전 11:49
작성자: haidn1994@gmail.com
URL: <http://code.i-harness.com/ko/q/3b4af9>

수정한 날짜: 2018-02-02 오후 8:07

공부에 도움되는 자료들 - 첫번째 스터디

재귀 함수에 대한 이해

함수형 프로그래밍에서 함수, 특히 재귀함수는 빼놓고 생각할 수 없기 때문에 반드시 공부해야 한다. 특히 놓치지 말아야 할 것은 함수란 무엇(What)인가에 대한 이해도 중요하지만(특히 함수형 프로그래밍에서 함수란 무엇인지에 대한 이해가 필요), 함수란 어떻게(How) 구성되는 지에 대한 이해도 중요하다고 생각한다. 전자는 스터디를 진행하면서 차근차근 알아가고 후자는 필요한 자료를 충분하게 준비했으니 하나씩 공부하기를 바란다.

- 다음 시간까지 해올 것
 - 컴퓨터공학CAMP 1강 18P ~ 22P의 내용을 학습하고 재귀함수의 간략한 원리와 재귀함수 작성에 필요한 2가지 요소를 외워올 것
 - 초기 조건(맨 아래로 내려갔을 때 계산해줄 것)
 - 이는 기저 사례(base case)라는 말로도 표현할 수 있다.
 - 반복 조건(맨 아래가 아닌 수에서 어떻게 예전 값과 같이 계산하는지)
 - 특히 18P의 2번째 더 읽어보기는 꼭 읽어보면 좋을 것 같다.
- 함수가 무엇(What)인지에 대한 이해는 SICP스터디를 진행하면서 차근차근 알아갈 것이다. 하지만 함수가 어떻게(How) 구성되는지에 대한 것은 스스로 공부하자. 여기 좋은 자료를 제공할 것이다. 난이도 순서로 배치했고, 궁금할 때 한번씩 읽어보자.
 - 컴퓨터공학CAMP 1강 18장 - 더 읽어보기: <http://rvulib.tistory.com/318>
 - 프로그래밍 원리: chapter13_about_function.pdf - 한세경 저(helpful_res에 있음)
 - 어셈블리어(Assembly Language)와 C 그리고 호출 규약(Calling Convention): <http://kkamagui.springnote.com/pages/339546>
 - 64비트 멀티코어 OS 원리와 구조 - 한승훈 141P ~151P: 5.2.2 스택 초기화와 함수 구현(조만간 올릴 예정)

기본적으로 알아야 할 점화식과 알고리즘들

알고리즘 과목을 공부하다보면 꼭 나오는 단골 점화식과 알고리즘이 있다.

- 팩토리얼(계승)
- 피보나치 수열
- 최대공약수(GCD: Common Greatest Divisor) 구하는 유클리드 알고리즘
- 조합(Combination) 점화식 등등...

이 녀석들은 너무나 자주 등장하므로 그냥 외워버리는 편이 좋다. JS코드와 Scheme코드를 제공하도록 하겠다. 그러므로 꼭! 외워오도록 하자. 추가적으로 점화식도 제공하도록 하겠다. 수학적 정의가 궁금하다면 써보도록 하자.

JS 코드

```
// Indentation Style은 K&R을 따르도록 한다.

// 정수 판정을 할 때 왜 정제법으로 이를 판정할까? 잘 생각해보자.
function isInt(n)
{
    return n%1 === 0;
}

// 팩토리얼(계승)
function facto(n)
{
    // n이 1보다 작은지? n이 정수가 맞는지? 두가지 검사가 필요하다.
```

```

    if(n < 1 || isInt(n)) {
        console.log("invalid data");
        return -1;
    }

    return (n === 1) ? 1
           : facto(n - 1)*n
}

// 피보나치 수열의 점화식 표현
function fibo(n)
{
    if(n < 0 || isInt(n)) {
        console.log("invalid data");
        return -1;
    }

    return (n === 0) ? 0 :
           (n === 1) ? 1 : fibo(n - 2) + fibo(n - 1);
}

// GCD: 유클리드 알고리즘
function gcd(b, s)
{
    if((b > 0 || s > 0) || (isInt(b) || isInt(s))) {
        console.log("invalid data");
        return -1;
    }

    return (s === 0) ? b
           : gcd(s, b%s);
}

// 조합(중복조합)의 점화식 표현
// 보통 이항계수(binomial coefficient)의 점화식 표현이라고 부른다.
// 편의상 함수이름은 Combination의 C로 하겠다.
function C(n, r)
{
    if((n > 0 || r > 0) || (isInt(n) || isInt(r))) {
        console.log("invalid data");
        return -1;
    }

    return ((n === r) || r === 0) ? 1
           : C(n - 1, r - 1) + C(n - 1, r);
}

```

Scheme 코드

```

# Lisp는 아직 익숙하지 못해서 방어적 프로그래밍은 빼도록 하겠다.

(define (inc n)
  (+ n 1))

(define (dec n)
  (- n 1))

# 팩토리얼(계승)

(define (facto n)
  (* (facto (dec n)) n))

# 피보나치 수열

(define (fibo n)
  (cond (= n 0) 0
        (= n 1) 1
        (else (+ (fibo (dec n)) (fibo (dec (dec n)))))))

```

```
# GCD: 유클리드 알고리즘
```

```
(define (gcd b s)
  (if (= s 0)
      b
      (gcd s (remainder b s))))
```

```
# 조합 점화식
```

```
(define (C n r)
  (if (or (= n r) (= r 0))
      1
      (+ (C (dec n) (dec r))
         (C (dec n) r))))
```

프로그래밍 언어론

SICP는 함수형 프로그래밍(물론 프로그래밍을 공부하는데에도 좋다.)을 공부하는 데 좋은 교재다. 하지만 상당히 불친절한 교재이기도 해서, 다른 참고자료 없이 이 책만 들이파면 도대체 무슨 소리를 하는지 알기 어려운 경우도 상당히 많다.

SICP를 차근차근 읽다보면 가장 많이 겹치는 점이 있는 전공과목은 바로 프로그래밍 언어론이다. 따라서 참고도서를 하나 잡고, SICP를 읽어 나가면서 용어 해설을 하도록 하겠다.

SICP 앞쪽을 공부하다 보면 모름지기 좋은 프로그래밍 언어가 갖추어야 할 세가지 요소에 대해서 말하는데, 그 내용은 다음과 같다.

- 기본 식(primitive expression) - 언어에서 가장 단순한 것을 나타낸다.
 - 수와 산술 연산이 기본 데이터고 기본 프로시저다.
- 엮어내는 수단(means of combination) - 간단한 것을 모아 복잡한 것(compound element)로 만든다.
 - 엮은식을 겹쳐 쓰는 것이 여러 연산을 한데 묶는 수단이 된다.
- 요약하는 수단(means of abstraction) - 복잡한 것에 이름을 붙여 하나로 다룰 수 있다.
 - 이름과 값을 짝지어 정의한 것이 모자라나마 요약하는(간추리는) 수단이 된다.

프로그래밍 언어론에서 "기본식"과 "엮어내는 수단"에 대한 개념을 나타내는 용어는 잘 모르겠다. 하지만 "요약하는 수단"은 프로그래밍 언어론에서 바인딩이라는 개념으로 소개된다. 그러므로 다음 시간까지 바인딩에 대해서 공부하고, 그 외 몇가지 중요한 용어들도 공부해보면 좋겠다.

다음 시간까지 공부 해올 주제는 다음과 같다.

꼭 공부했으면 하는 주제

- 바인딩(D.W.Barron 표기법도 공부하고, 변수의 4요소에 대해서 알아보자.)의 개념에 대해서 알아보자.(참고자료를 읽어보면 알 수 있을)
 - 여기에 상수 표현을 위한 Barron 표기법과 상수의 3요소도 알아보면 좋다.
 - 이 그래프에 더붙어서 바인딩에 대한 설명도 공부하면 좋을 것 같다.
 - 참고자료: 프로그래밍 언어론: 변수(variable), 영역(scope), 수명(lifetime) <http://slidesplayer.org/slide/11304432/#>

꼭 공부하지 않아도 좋지만 유용한 자료

- 참고도서에서 제공하는 프로그래밍 언어 분야 용어의 국제 규격과 국가 규격을 참고하여 관련 용어중에서 필요한 것을 선정하여 공부하자.
알면 좋은 용어 리스트는 다음과 같다. 하단에서 언급하는 ISO/IEC 2382-2015뿐만 아니라 다른 출처도 사용해서 좋은 정의를 찾고, 학습해보자.
 - (어휘 토큰:lexical token, 어휘 요소:lexical element, 어휘 단위:lexical unit)
 - 언어 구성자(language construct)
 - 식별자(identifier)
 - 미리 정의된 식별자(predefined identifier)
 - 예약어(reserved word)
 - 중복정의/과적하다.(to overload)
 - 영역, 선언 영역(scope, scope of a declaration)
 - 정적(static)
 - 동적(dynamic)
 - 매개변수(parameter)
 - 인수(argument)

- (모듈:module, 단위 프로그램:program unit)
- 부프로그램(subprogram)
- (프로시저:procedure, 서브루틴:subroutine)
- 함수(function in programming language)
- 태스크(task in programming language)
- 참고로 참고도서에서 부록으로 제공하는 "프로그래밍 언어 분야 용어의 국제 규격과 국가 규격"은 ISO/IEC 2382-15를 뜻한다. 하지만 최근 ISO/IEC 2382-2015로 개정되었으며, 해당 웹문서에 대한 링크는 다음과 같다. <https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:ed-1:v1:en>
- 참고문헌
 - 서적: 프로그램 언어 개념 - 원유형 - 정역사
 - 웹문서: ISO/IEC 2382-15: <https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:ed-1:v1:en>

함수형 프로그래밍

SICP를 공부하는 목적중 하나는 함수형 프로그래밍을 공부하는 것이다. 하지만 프로그래밍 언어론을 공부할 때와 마찬가지로 SICP는 훌륭한 교재임이 틀림없지만, 불친절한 교재이기도 하다. 따라서 이를 공부할 때는 SICP만 들이파는 것보다는 참고서적을 몇 개 잡고, 참고하고 실습하면서 "이게 이런 뜻이었군!"하는 경험을 하는 것이 중요하다고 생각한다.

역시 몇가지 참고도서들과 파일, 웹문서를 알려주겠다.

- 참고도서
 - 함수형 자바스크립트 프로그래밍 - 유인동 저
 - 하스켈로 배우는 함수형 프로그래밍 chapter 0(파일 첨부) - 오카와 노리유키 저 - 정인식 역
 - 하스켈을 배우자는 뜻이 아니라, 0장에서 명령형 언어와 함수형 언어의 차이점과 함수형 언어를 사용해서 얻을 수 있는 이점을 상세히 설명하니 참고하자.
 - 함수형 사고 - 닐 포드 저 - 김재완 역: 안사도 된다. 필요할 때 내용을 조금씩 참고해서 스터디 자료에 포함시키겠다.
 - 함수형 자바스크립트 - 루이스 아텐시오 저 - 이일웅 역: 첨부한 웹문서에서 미리보기를 누르면 꽤 많이 볼 수 있다. 참고하라.
 - 함수형 자바스크립트 : 새롭고 올바른 자바스크립트 프로그래밍 기법 - 마이클 포거스 저 - 우정은 역: 미리보기를 눌러도 조금밖에 볼 수 없지만 저자가 매우 뛰어난 실력을 가진 사람("클로저 프로그래밍의 즐거움"을 집필함)이니 책을 사도 후회하지 않을거라 생각한다.
- 웹문서
 - 유인동씨의 인프런 강의(무료): <https://www.infllearn.com/course/%ED%95%A8%EC%88%98%ED%98%95-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D/>
 - 함수형 프로그래머가 되고 싶다고?(강사님이 알려줌) (Part 1~6: 리스트에서 찾을 것): <https://github.com/FEDevelopers/tech.description/wiki/%EB%B2%88%EC%97%AD-%EB%AC%B8%EC%84%9C>
 - 함수형 자바스크립트: http://www.hanbit.co.kr/store/books/look.php?p_code=B9124201558
 - 함수형 자바스크립트: 새롭고 올바른 자바스크립트 프로그래밍 기법 -
 - > http://www.hanbit.co.kr/store/books/look.php?p_code=B9124201558