SICP 단어 모음

노트북: 컴퓨터 프로그램의 구조와 해석
만든 날짜: 2018-02-02 오후 2:52 **수정한 날짜:** 2018-02-02 오후 8:13
작성자: haidn1994@gmail.com

스터디에서 짚고 넘어가야할 단어모음

프로그래밍(또는 컴퓨터 과학: 이하 CS)을 공부할때에는 용어의 뜻을 정확하게 아는 것이 중요한데 그 이유는 다음과 같다.

- 그 용어를 얼핏 봐서는 무슨 말을 하는 것인지 추측하기 어렵다. 따라서 다음과 같은 문제가 생긴다.
  - 대표적인 예가 "다이나믹 프로그래밍"이다.
    CS는 태어난지 얼마 안된 학문이라서 용어 체계가 제대로 확립되지 않고 폭발적으로 성장한 까닭에 이렇게 잘못 지어진 이름이나, 읽는 법은 같지만 맥락에 따라서 여러 뜻으로 해석되는 용어도 수두룩하다.
  - 게다가 CS가 다른 학문과는 다른 점은 그런 다른 분야들에 비해 추상성이 높다는 점이다. 따라서 용어의 뜻을 명확하게 학습하지 않으면 더 추상성이 높은 용어를 받아들이기 힘들어진다.
- 뜻을 잘못 이해하거나 어렴풋이 이해하는 경우가 많다.
  - 제일 골치아픈 경우다. 차라리 이해를 못한다면 용어의 정확한 정의를 찾아다니려는 노력이라도 할 텐데, 뜻을 오해해서 받아들이고 더 높은 개념을 이 오개념으로 이해하려고 드니 더 높은 수준의 공부가 안되거나 엄청난 비효율이 초래된다.

그리고 그 용어를 학습하는 데에도 다음 방법으로 하면 그냥 학습하는 것보다 더 깊은 이해를 할 수 있다.

- 먼저 해당 용어에 보편적인 뜻이 있는 지 알아보고, 있다면 그 뜻도 학습한다.
- 그리고 CS내부에서 그 용어가 어떤 뜻을 가지는 지 알아야 된다. 꼭 명확한 뜻풀이로 학습해야 한다.
  - 그리고 '세그먼트'처럼 IT에서도 그 맥락(context)에 따라서 여러가지 뜻을 가지는 용어는 더더욱 조심히 학습하도록 한다.

- 자료(data)
- 정보(information)
- 파일(file)
- 자료형(data type)
- 객체(object)
- 메시지(message)
- 프로그램(program)
- 프로그래밍 언어(programming language)
- 프로세스(process)
- 바인드(bind)
- 프로시저(procedure)
- 변수(variable)
- 상수(constant)
- 환경(environment: 292p, 335p)
- 스코프(scope)
- 블록 구조(block structure)

단어	뜻	비고
자료(data)	1. 자료는 사람, 물건, 조건, 상황 등을 묘사한 기본적인 사실들의 집합을 말하여, 일반적으로 가공되기 전의 상태를 말한다. 2. 자료는 수, 영상, 단어 등의 형태로 된 의미 단위이다. 보통 연구나 조사 등의 바탕이 되는 재료를 말하며, 자료를 의미있게 정리하면 정보가 된다. 3. 자료는 의미를 가지고 기록될 수 있는 어떤 것을 뜻한다. 정리하면 다음과 같다. • 자료는 사람, 물건, 조건, 상황등을 묘사한 기본적인 사실들의 집합으로 의미를 가지며, 기록될 수 있는 단위다. 이런 자료를 의미있게 정리 또는 가공하면 정보가 된다.	1. 통계학 - 이훈영 저 2. 한국어 위키백과 - 자료 3. 컴퓨터공학CAMP PPT 8차시
정보(information)	1. 요즘에는 컴퓨터의 정보 처리를 기반으로 한 정보(데이터)가 많이 대두된다. 정보의 원래 뜻에	1. 한국어 위키백과 - 정보 2. 영어 위키백과 - Information

	따라, 정보와 자료(데이터)를 구별하고, 정보를 "뜻을 가지는 자료"라고 생각하는 의견도 있지만, 이러한 분야에서는 전체적으로 정보의 뜻을 가지고 문제삼는 경우는 별로 없으므로 특별히 정보와 자료는 구별하지 않는다. 구분하자면, 데이터를 모아 둔 것이 자료라면 자료를 특정한 목적의 의사결정을 위해 가공한 형태를 정보라고 할 수 있다. ...(중략)... 결국 정보란 일정한 의도를 가지고 정리해 놓은 자료의 집합이며, 정보가 되기 위해서는 이용자, 즉 어떤 목적을 갖는 사람이 있어야 하고 자료가 처리되어야 한다. 그리고 정보는 이용자를 위하여 일정한 규칙에 따라서 재배열, 요약, 삭제하는 행위를 거쳐야 한다. 2. 정보는 뭔가를 알리는 것이다. 다시 말해서, 어떤 종류의 질문에 대한 대답이다. 3. 정보는 의사결정에 도움이 되도록 요약되거나 도표로 표시된 것으로써 효과적인 의사 결정을 위해 가공된 형태의 자료를 말한다. 정리하면 다음과 같다. • 정보는 의사결정 및 가치판단(다시 말해 "해석")에 도움이 되도록 정리되거나 가공된 자료의 집합이다. 즉, 정보는 그 정보의 이용자에게 뭔가를 알릴 수 있어야 한다.	3. 통계학 - 이훈영 저
파일(file in computer)	1. 파일은 컴퓨터가 다루는 정보(주의:information은 data가 아니다.)의 기본 단위다. 컴퓨터 파일의 구현은 비트 스트림으로 이루어진다. 2. 주의점: 중요한 것은 파일이라는 것이 임의의 정보를 포함하는 객체라는 사실이다.	1. TCP/IP 완벽 가이드 - Charles M. Kozierok 지음 - 강유, 김진혁, 민병호, 박선재 옮김 2. 이하동문
자료형(data type, type)	• **자료형**이란 우선 구성원(member)이 될 수 있는 자료들의 집합이라 간주할 수 있다. 그러나 정수형 변수 x가 값만 갖는다는 것은 별 의미가 없다. 변수가 갖는 값을 가지고 가감승제등 적당한 연산을 수행할 수 있어야 한다. 그래서 **자료형**이란 객체(objects)의 집합과 이 객체들의 실체(instance)들을 생성(create), 작성(bulit-up), 소멸(destroy), 수정(modify), 분해(pick apart)하는 연산들의 집합을 의미한다.	• 이해가 어렵다면 수학적 객체(mathmatical object)를 예시로 들면 이해에 큰 도움이 된다. • 예를 들어, 정수 체계를 생각해보자. ◦ 어떠한 정수도 1로 나누어 떨어지는 값을 가진다. ◦ 정수에서 가장 기본이 되는 연산은 가감승제다. ◦ 가감승제를 통해서 정수의 값을 변경할 수 있다. (사실 새로운 값을 얻는 것이고, 수학에서 값을 변경한다거나 상태를 바꾼다는 것은 불가능하다. 하지만 일단 넘어가자.) ◦ 정수는 머릿속 어딘가에 생각함으로써 기억 또는 저장하거나, 물리적인 형태로 기록 또는 저장할 수 있다. 펜으로 적거나, 컴퓨터 파일 형태로 저장하거나...
객체(object)	• 주의: 두 정의 모두 CS에서 사용하는 정의다. 1. **객체**는 기억장소와 이 기억장소의 값을 변경할 수 있는 연산의 집합 2. ... 뭔가를 읽으려면 어딘가 읽을 곳이 필요하다. 즉, 읽을 내용을 컴퓨터 메모리의 어딘가에 저장해야 한다. 그런 '**(기억)장소**'를 바로 **객체**(object)라고 한다. **객체**란 저장하는 정보의 종류를 가리키는 **자료형**(data type)과 연계된 메모리의 한 영역이다.	
메시지(message in computer)	1. 통신을 위해 만들어진 파일을 메시지라고 부른다. 2. (프로그래밍 언어에서) 객체가 자기 자신이나 또는 다른 객체에게 연산들 중 하나의 수행을 요청하는 것을 뜻한다.	1. TCP/IP 완벽 가이드 - Charles M. Kozierok 지음 - 강유, 김진혁, 민병호, 박선재 옮김 2. ISO/IEC 2382-2015를 참고함

프로그램(program)	1. 프로그램은 실행 가능한 파일이다.
	    1. 바이너리 형태로 저장된 프로그램은 OS가 바로 실행할 수 있다.
	    2. 사람이 읽을 수 있는 코드(code)형태로 저장된 프로그램은 컴파일 후 바이너리 형태로 만들어 실행시키거나, 인터프리터 등에게 이 파일을 실행하도록 명령하는 형태로 실행할 수 있다.
	2. 1의 정의에 비추어 보면, 다음과 같이 정의할 수 있다.
	프로그램은 실행 시에 프로세스를 어떻게 만들지에 대한 광범위한 정보를 담고 있는 파일이다.
		1. 리눅스 API의 모든 것 - 마이클 커리스크 지음 - 김기주, 김영주, 우정은, 지영민, 채원석, 황진호 옮김

프로그래밍 언어(programming language)	...프로그래밍 언어에 대한 정의로 "인간이 컴퓨터로 수행하고자 하는 바를 컴퓨터에게 전달하기 위하 표현법"이라 말하곤 한다. 그러나 이 정의는 약간 부적절한 면이 있다.
	1940년 이전의 컴퓨터에서는 전선 연결(hard-wired) 방법으로 프로그램을 수행했다. 즉, 필요한 작업을 수행하기 위해서 컴퓨터 내부 스위치를 세팅한다. 이러한 방법으로 필요한 계산을 수행하도록 지시할 수 있다고 해서 전선 연결하는 방법을 프로그래밍 언어를 사용한 프로그래밍이라고 하기는 어렵다.
	...(중략)...
	실제로 프로그래머들은 고급 수준의 추상화를 수행함으로써, 프로그램을 간략히 작성하고, 이해하기 쉬우며, 기계들 사이에 이식하기 쉽다는 것을 알게 되었다. 모든 프로그래밍 언어에서 배정, 반복, 선택과 같은 구조는 항상 사용되며 **특정 기계와 무관**하다.(human-readable 특성) 이러한 구조는 표준화된 형식으로 표현되어 기계가 실행할 수 있도록 번역될 수 있다.
	...(중략)...
	따라서 프로그램이 계속 기계 독립적이 되어 가는데도 불구하고 현대의 대부분의 프로그래밍 언어 들은 아직도 폰 노이만 구조를 그대로 반영하고 있다. 그러나 병렬처리기를 비롯한 새로운 컴퓨터 구조의 개발로 인하여 프로그래밍 언어로 하여금 특정 모델이나 특정 기계에 기반을 두기 보다는 계산과 처리 일반을 표현만 할 수 있으면 된다는 것에 경각심을 가지게 하였다. 그래서 프로그래밍 언어에 대한 정의를 다음과 같이 좀 더 정교하게 내리고자 한다.
	
	• **정의**: 프로그래밍 언어는 기계가 읽을 수 있고, 사람이 읽을 수 있는 형식으로 계산을 서술하기 위한 표기 체계이다.
	    ◦ 이 정의에서 나타난 세 가지 주요 개념은 다음과 같다.
	        ▪ **계산(computation)**
	        계산은 튜링 머신이라는 수학적 개념을 가지고 형식적으로 정의할 수 있다...
	        ▪ **기계가 읽을 수 있는(machine-readable)**
	        언어를 기계가 읽을 수 있도록 하려면, 효율적인 번역이 가능하도록 단순한 구조를 가지고 있어야 한다. 이것은 특정 기계에 종속된 특징이 아니라, 번역의 정확성과 복잡성을 염두에 두고 정확히 명시할 수 있는 일반적인 명세이다...
	        ▪ **사람이 읽을 수 있는(human-readable)**
	        기계가 읽을 수 있다는 것과는 달리 이것은 더욱 부정확한 표현이다. 사람이 읽을 수 있게 하기 위해서는 프로그래밍 언어가 컴퓨터의 행위를 읽기 쉽도록 추상화 시키는 방법을 제공해야 한다. 더구나 기계적 특징에 무지한 사람조차도 충분히 이해할 수 있도록 추상성을 제공해야 한다...
		• 출처: 프로그래밍 언어 개념 - 원유헌 저

프로세스	1. 프로세스는 실행중인 프로그램을 뜻한다.	1. 리눅스 API의 모든 것 - 마이클 커리스크

(process)	2. 프로세스는 프로그램을 실행하기 위해 자원이 할당된, OS(커널이)가 정의한 추상적인 존재다.	지음 - 김기주, 김영주, 우정은, 지영민, 채원석, 황진호 옮김 2. 이하동문
프로시저 (procedure)	다음 웹문서를 참고하라. 1. http://www.terms.co.kr/procedure.htm 2. http://www.terms.co.kr/routine.htm 3. http://www.terms.co.kr/function.htm - 2번 뜻 참고	
바인딩 (binding)	• 바인딩이란?: 이름에 어떤 속성을 관련시키는 일련의 작업 즉, 프로그램의 기본 단위(이름/식별자)에 이 단위가 택할 수 있는 여러 속성 중에서(자료형, 등과 같은 여러 보일러플레이트) 일부를 선정하여 결정하는 작업 • 자세한 것은 웹문서를 참고하라: http://slidesplayer.org/slide/11304432/#	
변수(variable)	1. 변수란 실행시간에 저장된 값이 변경될 수 있는 객체를 의미한다. 그리고 변수의 속성은 다음과 같다. 1. 이름/식별자(name/identifier): 제한된 길이의 영문자/숫자로 구성된 변수의 한 요소(component) 2. 형/타입 속성: 자료형(data type) 참고 3. 주소(참조)(address(reference)): 변수와 관련된 첫 번째 메모리 셀을 가리키는 식별자(id) 4. 값(value): 선언한 타입(type)에서 결정 가능한(후보) 값중 하나 5. 영역(scope): 변수의 이름이 이해되고 참조될 수 있는 프로그램의 부분 6. 수명(lifetime): 실행 시간에 그 이름에 관련된 값을 보유할 기억 장소가 배정되어 있는 시간 2. (느슨한 정의): 명명된 객체를 의미한다.	• http://slidesplayer.org/slide/11304432/#
상수(constant)	1. 상수란 실행시간에 저장된 값이 변경될 수 없는 객체를 의미한다. 1. 주의: 값이 배정될 수 없다는 의미는 아니다. **변경**이 불가능한 것이다. 2. 상수는 식별자로 주어지며 프로그램 수행 중 결합된 값이 결코 변하지 않는다. 선언문 또는 묵시적 선언으로 생성된다. 상수 허용시 고려사항은 다음과 같다. 1. 단순 자료형 또는 구조 자료형(record, array) 2. 상수 값 표현: 번역 시간이나 실행 시간에 편가되는 수식 가능 여부 3. 상수 값 배정 시간(정적: 프로그램 실행 전 한 번 or 동적: 상수가 정의된 블록을 시작할 때마다 매번) 4. 미리 정의된 상수 제공 여부	• 다음 웹문서를 참고하자: http://slidesplayer.org/slide/11304432/#
선언 (declarations)	1. 실행시 사용될 자료의 속성을 언어의 번역기에게 알려주는 프로그램 문장 **자료의 속성**: 자료형, 크기, 이름, 생성 시기, 소멸 시기, 참조하기 위한 첨자등	
환경 (environment)	1. 환경이란 지역 단위로 묶여진 장소와 관련된 모든 식별자들을 언급한 용어로서, 지역 변수는 물론 진입점과 비지역 변수들을 접근하기 위한 정보들도 포함한다. 1. **바탕 환경**은 맨 바깥쪽에 있는 환경으로, 어떠한 블록 안쪽에서나 블록이 하나도 없는 곳에서나 그 식별자들에 접근할 수 있다.	

	2. 지역 환경은 블록이 하나 이상있는 곳에 만들어지는 환경으로 블록 **안쪽 정의**(internal definition)와 바탕 환경, **2종류의 식별자**에 접근할 수 있다.	
영역(scope)	1. 영역(scope)이란 식별자의 영역을 의미하며, 프로그램에서 사용되는 식별자(변수, 상수, 레이블, 자료형, 부프로그램의 이름)가 이름의 효력을 나타낼 수 있는 범위를 의미한다. 2. 하지만 원래 영역(scope)란 좀 더 넓은 의미로 사용되는 선언 영역을 의미하며 국제 규격은 프로그램에서 어떤 선언이 유효한 부분이라고 정의한다.	
블록 구조(block structure)	...(중략)... Algol60은 복합문(compound statement) 개념을 이용하여 매우 우아한 영역의 개념을 도입하였다. 이 복합문은 **일련의 문장 그룹(프로지저/함수가 정의된 곳을)을 단일 문장으로 취급(프로시저/함수 호출한다.)**할 수 있어 매우 유용하게 사용하게 된다. 일련의 문장 집합을 하나의 단위로 표시하기 위해 begin-end로 묶어 복합문으로 사용했으며, begin-end 사이에 변수, 부프로그램, 레이블들을 선언할 수 있도록 했다. 복합문의 begin-end라는 예약어로 묶인 부분만이 자신의 영역이 되며, Algol60에서는 이 복합문을 블록(block)이라 불렀다. 이러한 블록 구조(block structure)는 영역의 정의를 위해 자주 사용되는데, Algol60은 이 begin-end 블록 내에서 선언된 항목들의 이름은 그 블록 안에서만 사용 가능한 **지역 식별자(local indentifier)**로 정의된다. 블록이란 선언을 가질 수 있어 자체로 새로운 프로그램 환경을 설정할 수 있는 특별한 언어 구조를 갖는 프로그램 단편이다. C, C++, Java 역시 복합문(두 개의 중괄호로 묶인 일련의 문장)에 선언문을 선언할 수 있어 Algol60과 같은 블록을 가지며, Pascal은 주 프로그램(main()?)과 부프로그램(프로시저, 함수)이 블록이다. ...(후략)...	