

컴퓨팅사고와 파이썬 프로그래밍

# Ch 1. 컴퓨팅 사고와 파이썬 프로그래밍 개요



교수 김 영 탁  
영남대학교 정보통신공학과  
(Tel : +82-53-810-2497; E-mail : ytkim@yu.ac.kr)

# Outline

- ◆ 컴퓨팅 사고 (Computational Thinking)
- ◆ 컴퓨터 구조와 핵심 기능
- ◆ 컴퓨팅 사고에서의 공개 소프트웨어 패키지 활용
- ◆ 알고리즘의 표현과 유사코드 (pseudo code)
- ◆ 기본 알고리즘과 자료구조
- ◆ 파이썬 프로그래밍 환경 준비



## **컴퓨팅 사고 (Computational Thinking)**

# 컴퓨팅사고와 프로그래밍

## ◆ 인류문명의 진화

- 새로운 도구와 재질, 동력장치를 사용하여 **생산성** (*productivities*)을 향상:
  - 석기, 청동기, 철기, 증기(steam) 엔진, 전기, 산업용 로봇, **컴퓨터**, 인터넷, 분산처리, 인공지능 (AI), 빅데이터, 사물인터넷 (IoT)
- 다수 인원의 효율적인 **협동/협력** (*collaborations*)
- 부가가치 창출의 **성능** (*performance*) 개선
- **에너지 효율성** (*energy efficiencies*)



(a) 석기 망치, 창



(b) 철기 농기구



(c) 증기 (steam) 엔진



(d) 전기, 산업용로봇



(e) 컴퓨터, 인터넷, 분산처리



(f) 인공지능 (AI), 빅데이터, 사물인터넷(IoT)

## 제4차 산업혁명과 프로그래밍

### ◆ 산업혁명

산업혁명	시기	주요 변화
1차	18세기	증기 기관 (steam engine)의 발명으로 사람과 동물의 힘보다 더욱 더 큰 힘을 효율적으로 발생시킬 수 있었으며, 영국에서는 이를 활용하여 섬유생산을 획기적으로 증대시킴
2차	19세기 ~ 20세기 초반	전기 에너지 생산 및 공급 기술이 개발되고, 표준화된 모듈을 컨베이어 벨트에서 조립하는 대량 생산 기술이 개발됨
3차	20세기 후반	컴퓨터와 인터넷 기술의 개발로 과학 기술 분야의 대용량 데이터 처리가 가능하게 되었으며, 서로 다른 지역에 떨어져 있는 사람과 장치간의 협동 작업이 가능하게 되어 기술 발전이 더욱 가속화 됨
4차	2015년 ~ 현재	사물인터넷, 빅데이터, 인공지능, CPS (Cyber Physical System) 기술의 개발로 사람, 사물, 공간을 초연결, 초지능화 시킴

### ◆ 제4차 산업혁명의 핵심 기술

- 사물인터넷 (Internet of Things, IoT)
- 빅데이터 (Big Data)
- 인공지능 (Artificial Intelligence, AI), 기계학습 (Machine Learning ML), 심층학습 (Deep Learning, DL)

➔ 모두 소프트웨어가 핵심이 되는 기반 기술임 !



# 컴퓨팅 사고

## ◆ 컴퓨팅사고 (Computational Thinking)

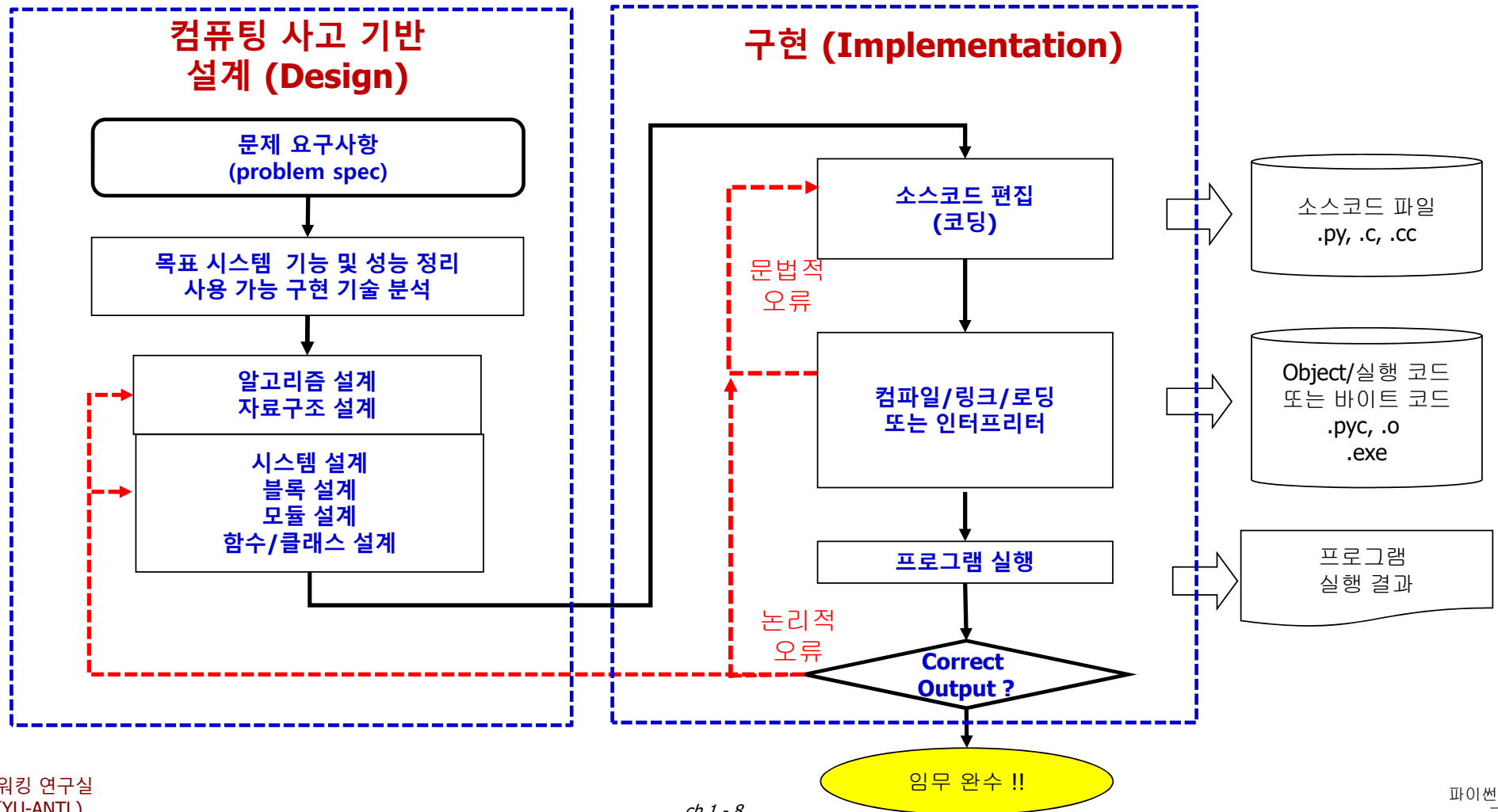
- 문제해결 방법에서 컴퓨터 장치와 프로그래밍을 활용하여 문제를 정확하고, 신속하며, 효율적으로 해결할 수 있도록 생각/사고
- 컴퓨터나 사람이 문제해결을 효과적으로 수행할 수 있도록 주어진 문제를 일반화 (generalize)/추상화 (abstract)하여 정의하고, 그 문제를 해결하는 방법을 체계적으로 정리하며, 이 문제와 유사한 다양한 변종 문제 해결에서도 사용할 수 있게 하는 사고 과정 (thinking process) 전체를 의미

# 컴퓨팅 사고

## ◆ 컴퓨터 기반의 문제해결 단계

단계	세부 실행 내용
문제 요구사항 분석	<ul style="list-style-type: none"> <li>주어진 문제의 요구사항을 분석(analyze)하고 논리적으로 타당한 요구인지 파악</li> </ul>
주어진 큰 문제를 작은 문제로 분해	<ul style="list-style-type: none"> <li>주어진 큰 문제를 작은 문제/작업 단위로 분해 (decompose)</li> </ul>
데이터 수집 분석, 규칙과 패턴 파악	<ul style="list-style-type: none"> <li>데이터를 수집하여 분석하고, 문제에 포함된 규칙과 패턴을 파악</li> </ul>
모형화(modeling)/추상화(abstraction)	<ul style="list-style-type: none"> <li>규칙과 패턴을 기반으로 모형화/추상화</li> <li>특징 추출과 일반화 (generalization)</li> </ul>
알고리즘 구상 및 설계	<ul style="list-style-type: none"> <li>작은 단위 문제들에 대하여 이미 개발되어 있는 컴퓨터 기반 해결 방법을 활용하여 문제를 해결할 수 있도록 문제 해결의 단계적 과정을 알고리즘으로 구성하고, 순서도나 유사코드로 표현</li> </ul>
프로그램으로 구현 (코딩)	<ul style="list-style-type: none"> <li>구성된 알고리즘을 프로그램으로 구현하고 컴퓨터를 사용하여 모의시험 수행</li> </ul>
실행 결과 확인, 분석	<ul style="list-style-type: none"> <li>모의시험 수행 결과를 측정 및 분석하여 당초 주어진 문제를 해결하는 방안이 되었는지 확인</li> </ul>
알고리즘의 개선	<ul style="list-style-type: none"> <li>모의시험 수행 결과 분석에 따라 알고리즘을 개선</li> <li>알고리즘 구상 및 설계를 보완하며, 구현 및 실험을 다시 실행</li> </ul>

# 컴퓨팅 사고를 기반으로 한 프로그램 설계 및 구현 절차

























# 컴퓨팅 사고와 프로그래밍 교육

## ◆ 주요 프로그래밍 언어와 특성

프로그래밍 언어	특성
C	<ul style="list-style-type: none"> <li>하드웨어를 직접 제어하는 성능이 좋아 많은 운영체제 (operating system, OS)의 구현에 사용됨</li> <li>절차 위주의 프로그래밍 구성, 컴파일러 사용</li> </ul>
C++	<ul style="list-style-type: none"> <li>객체 지향형 (object-oriented)의 클래스와 객체 (object)를 사용하며, C의 기능도 함께 포함</li> <li>컴파일러 사용</li> </ul>
C#	<ul style="list-style-type: none"> <li>.NET 환경에 최적화되어 있으며 컴포넌트 기반의 소프트웨어 개발 패러다임을 지원하는 컴포넌트 지향 프로그래밍 언어</li> <li>단순한 콘솔 응용 프로그램부터 인터넷 및 분산환경 시스템, 모바일 시스템을 위한 응용 프로그램, 게임과 같은 멀티미디어 콘텐츠에 이르기까지 다양한 소프트웨어를 쉽게 제작할 수 있도록 설계된 범용 프로그래밍 언어</li> <li>C++의 객체지향성과 자바의 분산환경 처리에 적합한 다중성을 지님</li> </ul>
Java	<ul style="list-style-type: none"> <li>하드웨어 플랫폼에 상관없이 사용할 수 있다는 장점이 있음</li> <li>객체지향형의 클래스와 객체 가능 사용</li> <li>인터프리터 또는 컴파일러 사용</li> <li>안드로이드 앱이 Java 기반이며, 웹 기반의 업무 시스템에서 Java가 많이 사용됨</li> </ul>
Java Script	<ul style="list-style-type: none"> <li>Java 스크립트 프로그래밍</li> </ul>
Python	<ul style="list-style-type: none"> <li>객체 지향형 클래스와 객체 사용가능</li> <li>인터프리터 사용</li> </ul>
Visual Basic	<ul style="list-style-type: none"> <li>영상 처리 관련 전문 프로그래밍</li> </ul>
Go	<ul style="list-style-type: none"> <li>2009년 Google이 개발한 프로그래밍언어. Garbage collection 기능 제공, concurrent (동시처리/병행처리) 지원</li> </ul>
Swift	<ul style="list-style-type: none"> <li>애플의 iOS와 macOS를 위한 프로그래밍 언어</li> <li>기존의 애플 운영체제용 언어인 오브젝티브-C와 함께 공존하도록 개발. 오브젝티브-C와 마찬가지로 LLVM으로 빌드되고 같은 런타임을 공유. 클로저, 다중 리턴 타입, 네임스페이스, 제네릭스, 타입 유추 등 오브젝티브-C에는 없었으나 현대 프로그래밍 언어가 갖고 있는 기능을 많이 포함시켰으며 코드 내부에서 C나 오브젝티브-C 코드를 섞어서 프로그래밍하거나 스크립트 언어처럼 실시간으로 상호작용하며 프로그래밍 할 수 있음</li> </ul>

# 2022년도 프로그래밍언어 순위

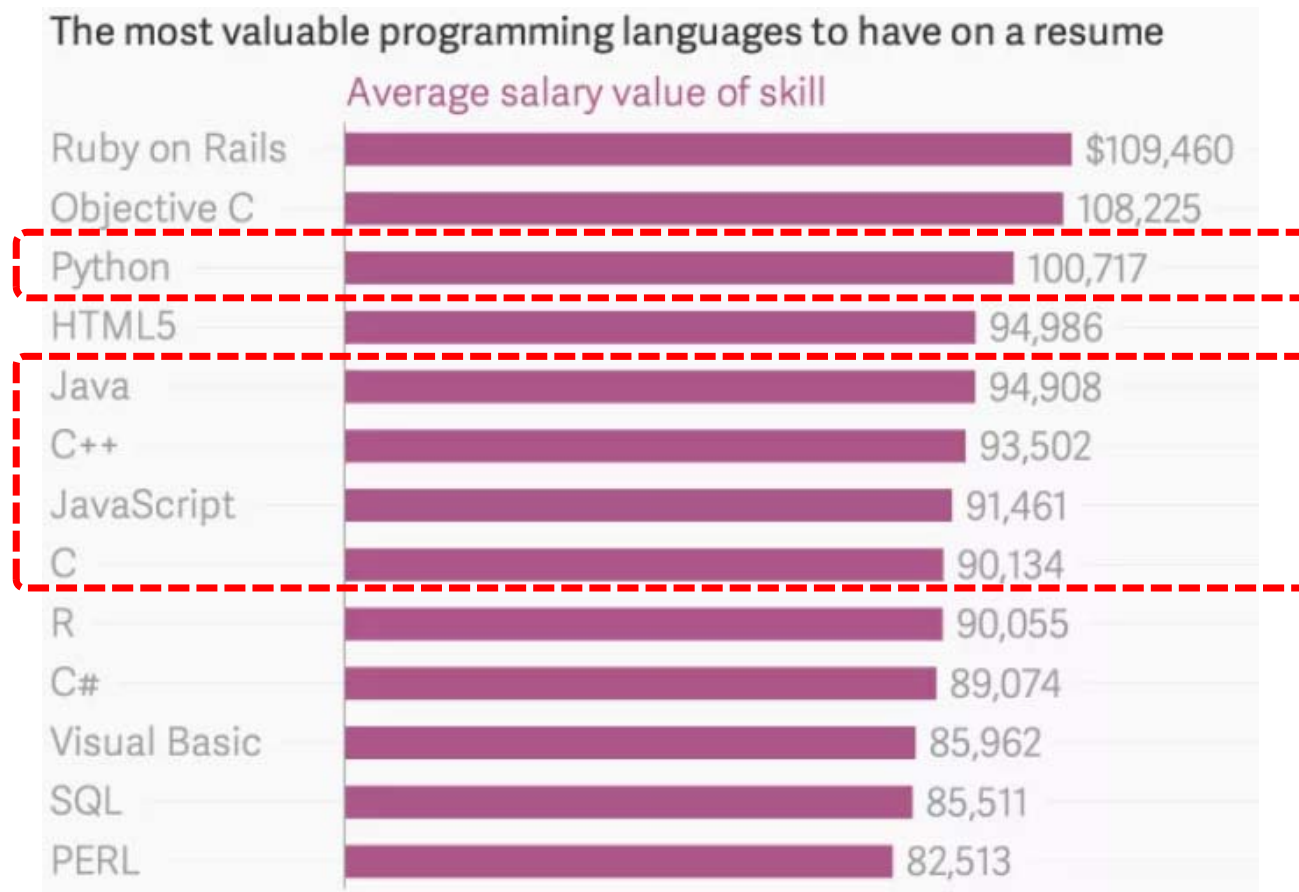
## ◆ TIOBE Index for Feb. 2022 (<https://www.tiobe.com/tiobe-index/>)

Feb 2022	Feb 2021	Change	Programming Language	Ratings	Change
1	3	▲	 Python	15.33%	+4.47%
2	1	▼	 C	14.08%	-2.26%
3	2	▼	 Java	12.13%	+0.84%
4	4		 C++	8.01%	+1.13%
5	5		 C#	5.37%	+0.93%
6	6		 Visual Basic	5.23%	+0.90%
7	7		 JavaScript	1.83%	-0.45%
8	8		 PHP	1.79%	+0.04%
9	10	▲	 Assembly language	1.60%	-0.06%
10	9	▼	 SQL	1.55%	-0.18%
11	13	▲	 Go	1.23%	-0.05%
12	15	▲	 Swift	1.18%	+0.04%
13	11	▼	 R	1.11%	-0.45%
14	16	▲	 MATLAB	1.03%	-0.03%
15	17	▲	 Delphi/Object Pascal	0.90%	-0.12%
16	14	▼	 Ruby	0.89%	-0.35%
17	18	▲	 Classic Visual Basic	0.83%	-0.18%
18	20	▲	 Objective-C	0.81%	-0.08%
19	19		 Perl	0.79%	-0.13%
20	12	▼	 Groovy	0.74%	-0.76%



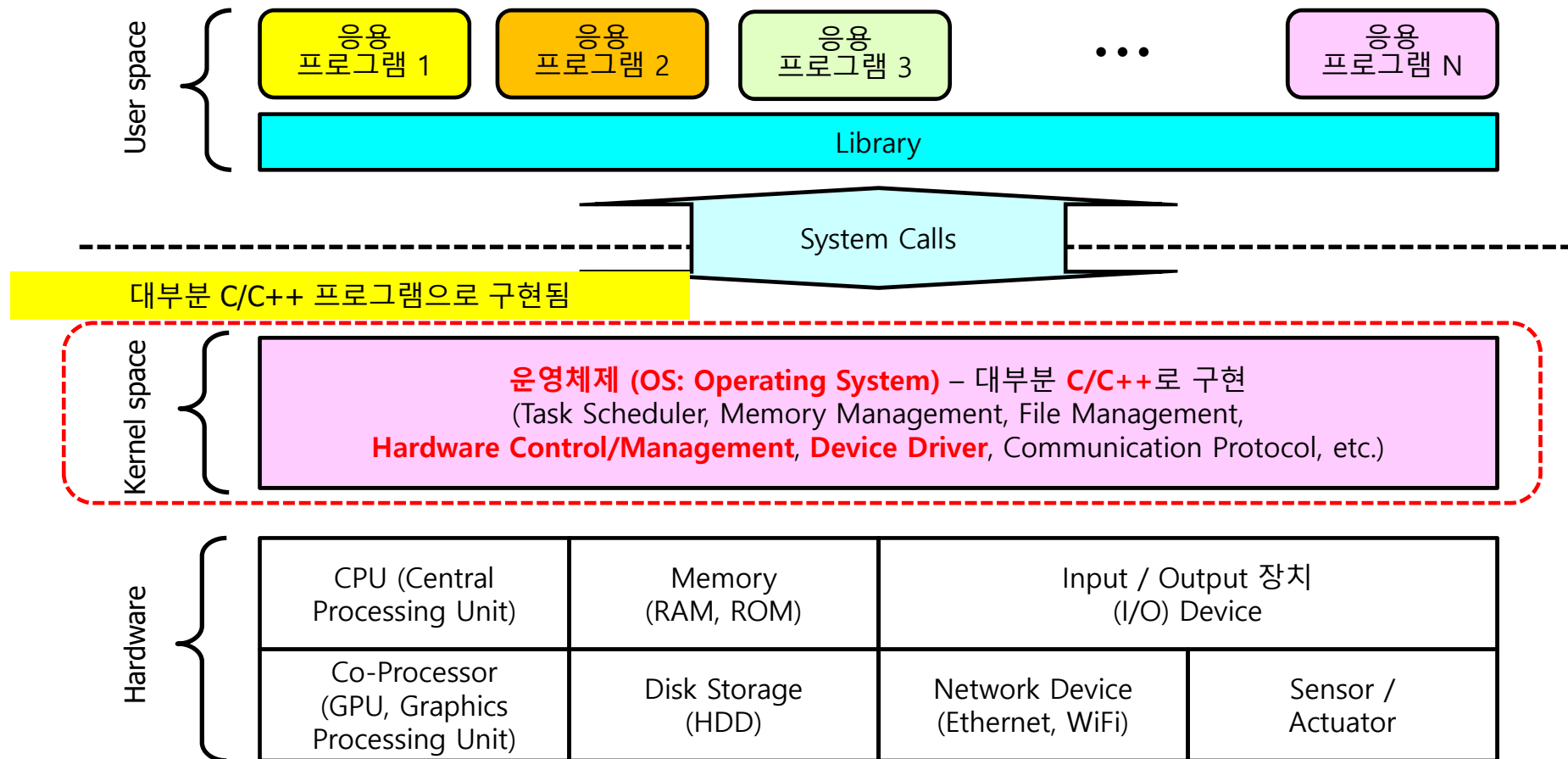
## 프로그래밍언어별 연봉

◆ <http://techneedle.com/archives/19283>



# 컴퓨터 구조와 핵심 기능

# 컴퓨터 기본 구조 – 하드웨어, 운영 체제 및 응용 소프트웨어



# 컴퓨터에서 사용되는 데이터 표현 단위

## ◆ 비트 (bit)와 바이트 (byte)

데이터 표현 단위	비트 수	바이트 수	데이터 값 범위	설명
비트 (bit)	1		0 또는 1을 표시	On/Off 상태 표현
바이트 (byte)	8	1	0 ~ 255	최대 256가지의 값을 구분하여 표현

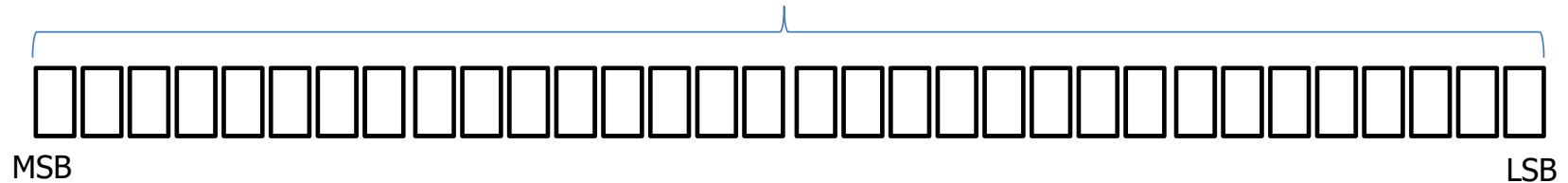
## ◆ 기본 숫자 자료형

기본 자료형	바이트 수	데이터 값 범위	주요 용도
문자 (character)	1	0 ~ 255	ASCII 문자 표현
	2	0 ~ 65535	unicode 문자 표현
정수 (integer)	1	0 ~ 255	부호 (sign)를 사용하지 않는 양수 범위의 정수
		-128 ~ +127	부호 (sign)를 사용하는 정수
	2	0 ~ 65535	양수 범위의 정수
		-32768 ~ +32767	부호 (sign)를 사용하는 정수
	4	0 ~ 4294967295	부호 (sign)를 사용하지 않는 양수 범위의 정수
		-2147483648 ~ +2147483647	부호 (sign)를 사용하는 정수
실수 (float) (참고: IEEE 754)	4	$3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$	부호 (sign): 1 비트 지수 (exponent): 8비트 가수(유효자리): 23비트
	8	$1.79 \times 10^{-308} \sim 1.79 \times 10^{308}$	부호 (sign): 1 비트 지수 (exponent): 11비트 가수(유효자리): 52비트

# 정수 (integer)의 비트 단위 표현

(a) unsigned integer

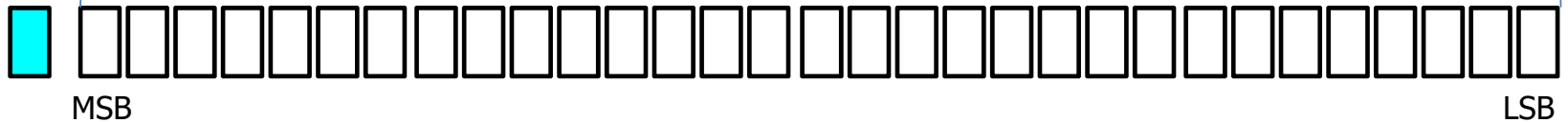
32-비트로 정수 표현



(b) signed integer

31-비트로 정수 표현

sign bit



## ◆ 8-bit 정수의 2의 보수 (2's complement) 표현

정수 (integer)의 음수 (negative number)를 2의 보수 (2's complement)로 표현하는 방법:

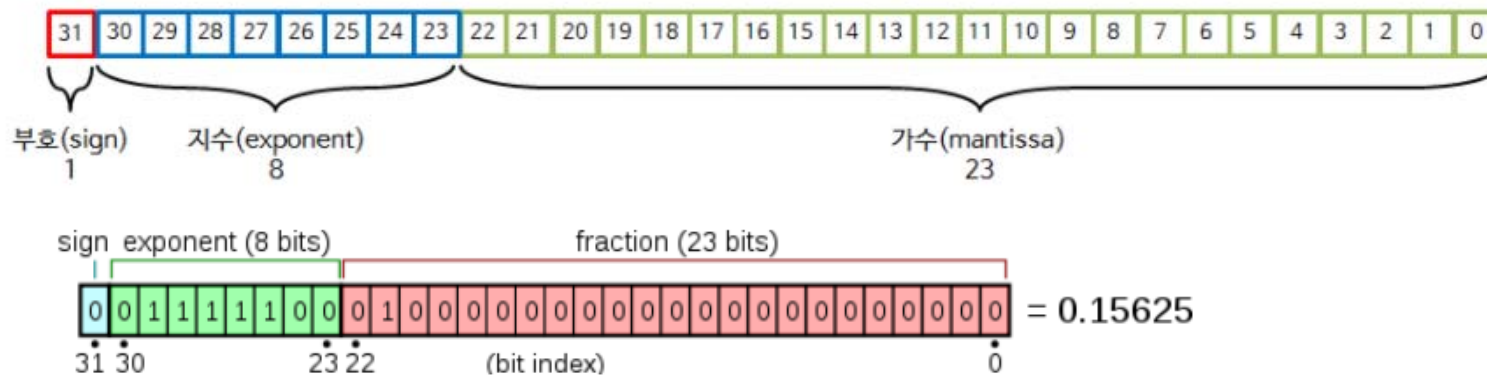
- (1) 해당 정수의 비트 표현에서 0을 1로, 1을 0으로 변환 (1's complement)
- (2) 결과에 1을 더함



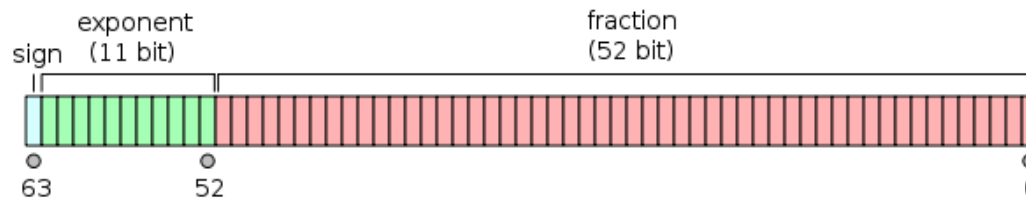


## 실수 (float)의 이진수 비트 단위 표현

### ◆ IEEE 754 float 표준 - 4 bytes (32 bit)



### ◆ IEEE 754 double 표준 - 8 bytes (64 bit)



## 정수 (integer) 데이터의 표현

### ◆ 10진수 (decimal), 2진수 (binary), 8진수 (octal)

10진수 (decimal)	2진수 (binary)	16진수 (hexadecimal)	8진수 (octal)
0	0000	0x0	00
1	0001	0x1	01
2	0010	0x2	02
3	0011	0x3	03
4	0100	0x4	04
5	0101	0x5	05
6	0110	0x6	06
7	0111	0x7	07
8	1000	0x8	10
9	1001	0x9	11
10	1010	0xA	12
11	1011	0xB	13
12	1100	0xC	14
13	1101	0xD	15
14	1110	0xE	16
15	1111	0xF	17

# 컴퓨터에서 사용되는 문자 자료형

## ◆ ASCII (American Standard Code for Information Interchange)

	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
0x00	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0x10	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0x20	SP	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0x30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0x60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

## 프로그램 실행과 데이터 처리

### ◆ 컴퓨터 프로그램의 실행 – 데이터 입력, 데이터 처리, 결과 출력 및 활용

컴퓨터 프로그램 예	데이터 입력	데이터 처리	결과 출력 및 활용
스마트 온실의 온도/습도 제어	온도 센서 데이터 습도 센서 데이터	온도와 습도 데이터를 기준값과 비교, 통계 분석	액추에이터 (히터, 냉방기)의 구동
스마트 자율 주행	GPS 위치 데이터 차선 영상 데이터 도로 표지판 영상 데이터 근접 센서 데이터	차선 및 경로 유지 분석, 근접 차량 및 장애물 분석	속도 및 방향 제어
네비게이션의 최단 경로 탐색	목표 지점 GPS 위치 데이터	현재 위치에서 목표 지점까지의 최단 경로 탐색	목표지점까지의 경로 제어
스마트 단말 장치의 예측 문자열 제시	문자 입력	현재까지 입력된 문자열에 가능성이 높은 문자열들의 검색	현재까지 입력된 문자열에 가장 확률이 높은 상위 문자열 제시

## 컴퓨팅 사고에서의 공개 소프트웨어 패키지 활용

# 공개 소프트웨어 패키지의 활용

## ◆ Open Source Software Package

- 프로그램 소스코드가 공개되어 있어  
교육이나 연구에서는 무료로 사용할 수 있는 소프트웨어

## ◆ 공개 소프트웨어의 예

- 운영체제: Linux
- 프로그래밍 언어: Python
- 과학기술 분야 연산: NumPy, SciPy, Matplotlib
- 빅데이터 통계 분석 및 예측: R, Pandas
- 영상처리: OpenCV
- 인공지능 (AI): TensorFlow, Keras



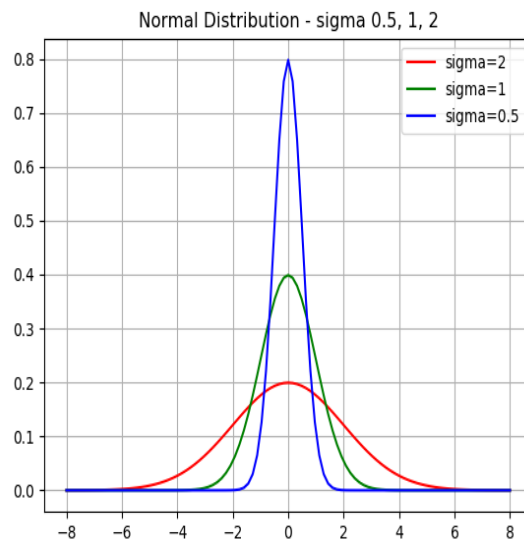
# 파이썬의 확장 패키지

파이썬 확장 패키지	설 명
NumPy	<a href="http://www.numpy.org/">http://www.numpy.org/</a> 파이썬 프로그램의 과학 기술 계산을 위하여 필요한 확장 패키지이며, 다음 기능들을 포함: - 우수한 성능의 다차원 배열 - 다양한 함수, 유니버설 함수 - C/C++ 및 Fortran 프로그램 모듈을 결합할 수 있는 기능 - 선형대수, 푸리에 변환, 랜덤 숫자 생성 및 응용 기능
SciPy	<a href="https://www.scipy.org/">https://www.scipy.org/</a> SciPy ("사이 파이"로 발음)는 파이썬 기반의 수학, 과학, 공학 분야 프로그램을 오픈 소스 소프트웨어로 설계, 구현 및 활용할 수 있는 환경 제공. NumPy, SciPy, Matplotlib, IPython, Sympy, pandas 등의 사용 환경 제공
Matplotlib	<a href="https://matplotlib.org/">https://matplotlib.org/</a> Matplotlib은 문서 및 서적 출판에서 사용가능한 품질의 2차원 그래프 및 도형 출력 라이브러리이며, 전문 인쇄용 출력 포맷 지원. Matplotlib은 파이썬 스크립트, IPython 셸, Jupyter 노트북, 웹 응용 서버 등에서 사용할 수 있으며, 다수의 그래픽 사용자 접속 툴 킷에서 사용가능.
OpenCV	<a href="https://opencv.org/">https://opencv.org/</a> OpenCV는 실시간 영상 처리가 가능하도록 계산 능력을 강화시킨 C/C++ 라이브러리이며, 멀티 코어 병렬 처리 기능을 활용함. OpenCL 기능을 사용하여 다양한 신호처리 가속 하드웨어 모듈을 활용할 수 있게 함. OpenVS는 BSD 라이선스로 제공되며 교육용 및 상업용에서 무료로 사용 가능. C++, 파이썬, Java 접속 기능이 제공되며, Windows, Linux, Mac OS, iOS 및 Android 환경에서 사용 가능.
R	통계 분석에 많이 사용되는 확장 패키지
Pandas	데이터 분석 확장 패키지 <a href="https://pandas.pydata.org/pandas-docs/stable/">https://pandas.pydata.org/pandas-docs/stable/</a>
TensorFlow	인공지능 분야에 많이 사용되는 확장 패키지
Keras	파이썬으로 구현된 딥러닝 라이브러리이며, TensorFlow위에서 딥러닝 모델을 쉽게 구성할 수 있게 함

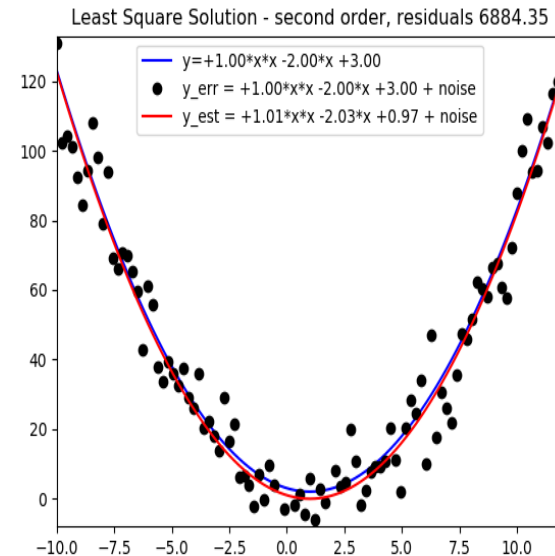


# 기초 수학 함수와 그래프 표현

## ◆ NumPy 패키지, Matplotlib 패키지



(a) 정규분포 함수 (normal distribution)



(b) 최소 자승해 2차 방정식



## 알고리즘의 표현과 유사코드

# 알고리즘이란?

## ◆ 알고리즘 (Algorithm)

- 컴퓨터 프로그램으로 문제를 해결하는 절차와 방법에 대한 정리
- 큰 문제를 작은 문제로 나누어 해결하는 절차도 포함됨
- 문제해결에 포함되는 절차와 방법을 가능한 한 세부적으로 표현하여야 함
- 순서도 (flow chart)나 유사코드 (pseudo code)로 표현하여 쉽게 수정 및 편집할 수 있게 함
- 실제 프로그램 소스코드 작성 이전에 알고리즘과 자료구조에 대한 확실한 구상이 있어야 함

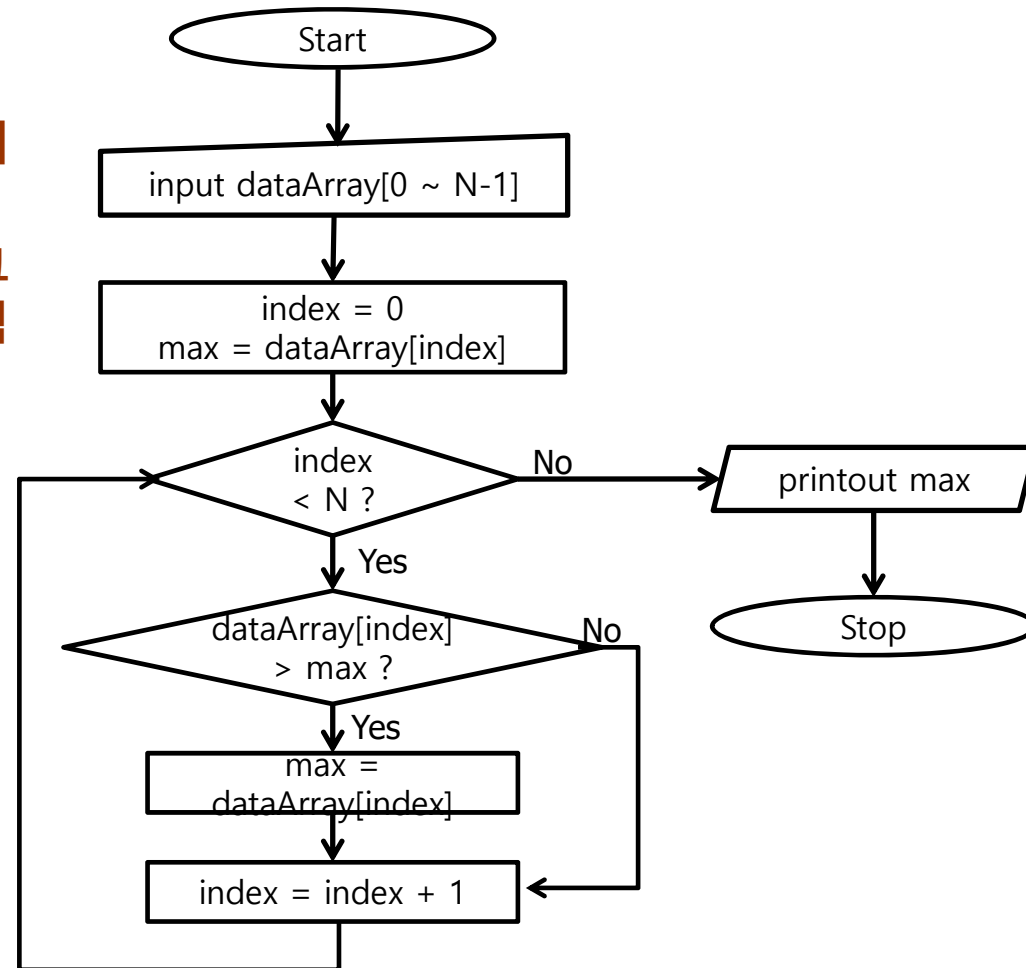
# 알고리즘의 설계

## ◆ 알고리즘의 설계

- 프로그램/소프트웨어 개발에서 가장 핵심적인 부분
- 어떤 절차와 단계를 밟아서 어떤 순서로 작업을 처리할 것인지를 설계
- 순서도 (flow chart) 또는 유사 코드/의사코드 (pseudo code)로 표현
- 알고리즘은 프로그래밍 언어와는 무관
- 알고리즘은 원하는 결과를 얻기 위하여 밟아야 하는 단계에 집중적으로 초점을 맞추는 것



## 순서도 (flow chart)

- ◆ 프로그램에서의 논리순서 또는 작업순서 등을 그래픽으로 표현하기 위한 형식
- ◆ 알고리즘이 복잡하면 그래픽으로 표현하기가 힘들어진다.



# 순서도에서 사용되는 기호와 의미

## ◆ Flow Chart Symbols

순서도 기호	의미	순서도 기호	의미
	단말(단자): 알고리즘의 시작과 끝을 표시		순서 흐름선 (flow line)
	데이터의 입력과 출력		판단, 분기
	데이터의 수동입력		지연 (delay)
	준비		논리접합, 가산접합
	데이터 처리		정렬, 분류
	미리 준비된 처리 (서브 루틴)		대조
	문서, 서류		다른 페이지 연결자
	디스플레이		페이지 내 연결자

# 유사코드/의사코드 (Pseudo Code)

## ◆ 유사코드/의사코드 (Pseudo Code)

- 프로그래밍언어가 가지고 있는 모든 문법적인 상세한 내용들을 다 포함하는 것이 아니라 기본적인 골격만 이해할 수 있도록 소스코드와 비슷하지만 간략화된 표현 방법

## ◆ 유사코드 사용의 장점

- 프로그램의 구조와 실행 상세 내용을 잘 표현할 수 있으며, 프로그램 소스코드 작성 단계 이전에 **설계 문서**로서 작성
- 프로그램의 실행 순서를 조리있게 **설명**할 수 있고, 프로그램 구조와 실행 내용을 검토할 수 있음
- 프로그램의 실행 내용과 절차를 procedure(), sub\_procedure() 등으로 **그룹화**하여 **체계적/계층적으로 설계**할 수 있음
- 프로그램의 구조와 실행 내용의 설계 문서를 일반 문서 작성기로 **쉽게 변경 및 보완**할 수 있음

## 유사코드 작성 예

### ◆ 유사코드 작성 예 – Draw\_Rectangle()

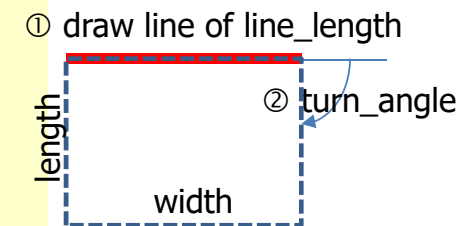
#### Procedure Draw\_Rectangle()

```
1.  /* input argument: None */
2.  /* output result: Rectangle drawn on canvas */

3.  width = 200
4.  length = 100

5.  draw_line(width)
6.  turn_right(90)
7.  draw_line(length)
8.  turn_right(90)
9.  draw_line(width)
10. turn_right(90)
11. draw_line(length)
12. turn_right(90)

13. end procedure /* end of procedure Draw_Rectangle() */
```



# 유사코드 (pseudo code) 작성법

## ◆ Pseudo code 작성에서 많이 사용되는 단어들

- 입력: input, read, obtain, get
- 출력: output, print, display, show
- 계산: compute, calculate, determine
- 초기화: initialize, set, init
- 요소를 추가: add, insert, increment
- 선형적으로 증가할 때: linear\_progress, sequence
- 반복: while, for, loop
- 조건문: if-then-else, if-elif, else
- 마지막에 조건문이 있는 반복문: repeat - until
- IF-THEN-ELSE 대신 조건 분기처리: case
- 부울 : TRUE, FALSE
- 그 외 : return, begin, end, exception



# 유사코드 (pseudo code) 작성법

## ◆ Pseudo code 작성법

- 유사코드 작성에서는 C, C++, Java, Python 등의 프로그래밍 언어 문법을 정확하게 준수하지 않아도 허용되나, 국제적으로 많이 통용되는 용어와 표현을 사용하는 것이 권장됨
  - 국제 학술 논문
  - 국제 협력 작업
- 시스템 소프트웨어의 핵심구조와 핵심 내용을 잘 표현할 수 있게 작성
- 조건문, 반복문은 if, while 다음에 조건식을 표시
- 복잡한 조건식 표현을 위하여 괄호()를 사용
- 조건문, 반복문 등이 수행되는 블록 단위 표현
  - ':' 다음 줄에 들여쓰기로 구분 - Python에서 사용
  - 또는, 중괄호 {}로 표현 - C, C++, Java에서 사용
- 프로그램 실행을 체계적으로 구분하여 procedure와 sub\_procedure의 계층적인 구조로 작성
- 각 procedure에서 실행되는 내용을 이해할 수 있는 이름을 사용
  - draw\_line(), turn\_right()
  - draw\_rectangle()
  - print(), input()

## 유사코드 작성 예

### ◆ 유사코드의 예 – DrawStar()

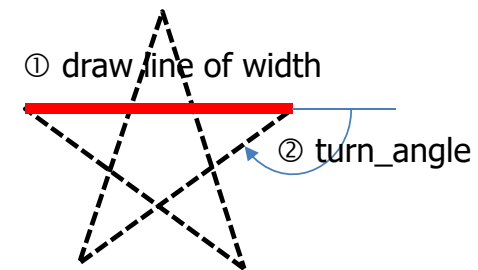
#### Procedure Draw\_Star()

```
1.  /* input argument: None */
2.  /* output result: Star drawn on canvas */

3.  width = 200
4.  turn_angle = (360 * 2) / 5
5.  count = 0
6.  move_to_start_position()

7.  while (count < 5) :
8.      draw_line(width)
9.      turn_right(turn_angle)
10.     count = count + 1
```

```
11.end procedure /* end of procedure Draw_Star() */
```



## 유사코드 (Pseudo Code) 작성

### ◆ Find\_Max()

- N개의 원소를 가진 배열에서 가장 큰 값 (최대값)을 찾아 출력

```
Procedure Find_Max(dataArray, N)  
1. /* input argument: dataArray[0..N-1]; */  
2. /*   array of data with N elements   */  
3. /* output result: the maximum of the given data array */  
4. index = 0 /* define and initialize index */  
5. max = dataArray[index]  
6. while (index < N):  
7.     if (dataArray[index] > max):  
8.         max = dataArray[index]  
9.     index = index + 1  
10. printout max  
11. end procedure /* end of Algorithm Find_Max() */
```

## 유사코드를 사용한 알고리즘의 변경

### ◆ Find\_MaxMinAvg()

- N개의 원소를 가진 배열에서 최대값, **최소값**, **평균값**을 출력

```
Procedure Find_MaxMinAvg(dataArray, N)
    /* input argument: dataArray[0..N-1]; */
    1. /* array of data with N elements */
    2. /* output result: the maximum, minimum and average values */
    /* of the given data array */
    3. index = 0 /* define and initialize index */
    4. sum = avg = 0.0
    5. max = min = dataArray[index]
    6. while (index < N):
    7.     if (dataArray[index] > max):
    8.         max = dataArray[index]
    9.     if (dataArray[index] < min):
    10.        min = dataArray[index]
    11.    sum = sum + dataArray[index]
    12.    index = index + 1
    13. avg = sum / N
    14. printout max, min, avg
    15. end procedure /* end of Algorithm Find_MaxMinAvg() */
```

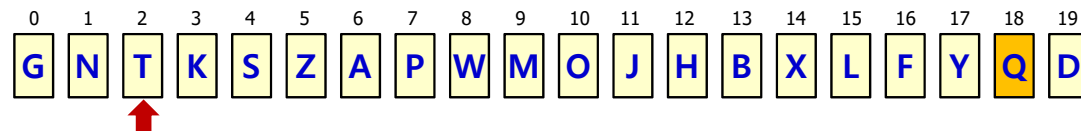
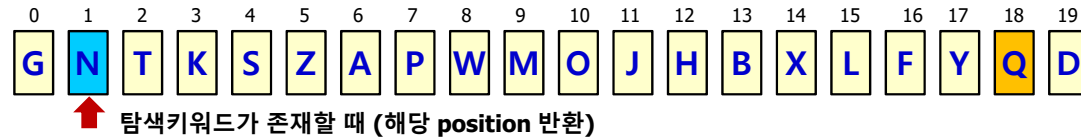
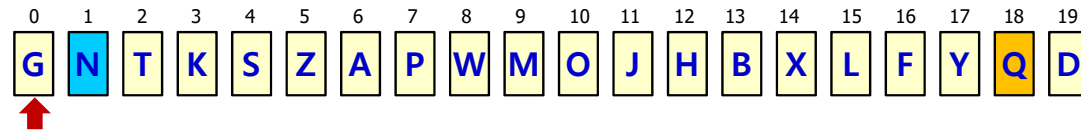
## 대표적인 알고리즘

# 대표적인 알고리즘

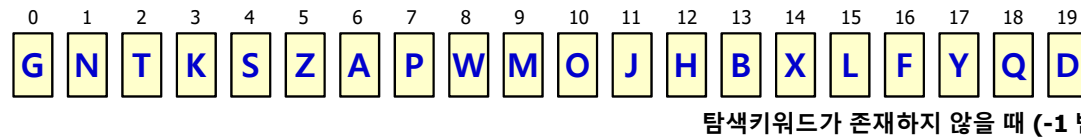
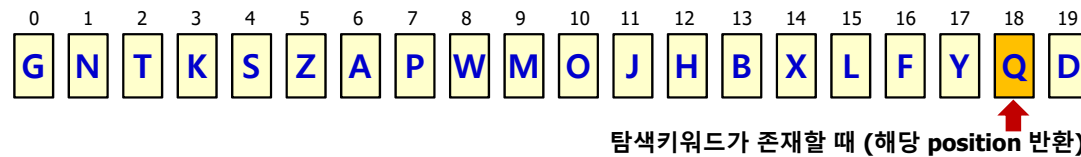
알고리즘	주요 기능
순차 탐색	배열이나 연결형리스트에 포함되어 있는 항목을 차례대로 탐색
이진 탐색	배열에 포함되어 있는 항목의 탐색 구간을 절반씩 줄여가며 탐색
선택 정렬 (selection sort)	배열에 포함되어 있는 항목의 탐색 구간을 하나씩 줄여가며, 가장 작은(또는 큰) 항목을 찾아 그 구간의 맨 앞에 두는 동작을 반복하여 전체 항목들을 정렬. 배열에
퀵 정렬 (quick sort)	배열에 포함되어 있는 항목의 탐색 구간을 중간 지점의 피벗 (pivot)을 중심으로 큰 항목들과 작은 항목들로 구분하여 정돈하며, (평균적으로) 절반씩 줄어든 탐색 구간에 대하여 재귀 함수 실행
병합정렬 (merge sort)	입력 배열을 같은 크기의 2개의 부분 배열로 분할 (partition)한 후, 각 부분 배열을 정렬하며, 각 부분 배열의 크기가 충분히 작지 않은 경우 분할 정복 (divide-and-conquer) 방법을 반복 적용함. 각 부분 배열의 정렬이 완료되면 이 정렬된 부분 배열들을 하나의 배열로 합병함. 추가적인 배열이 필요하며, 실제로 정렬이 이루어지는 시점은 2개의 부분 배열을 합병 (merge)하는 단계에서 이루어 짐.
힙 정렬 (heap sort)	Heap 자료구조를 사용하여 정렬. 전체 자료 중 일부만 정렬 순서로 추출하는 경우에 효율적임. 예를 들어 500명의 학생 중 상위 성적 10명만을 찾고자 할 때 가장 신속하게 처리할 수 있음.
해시	문자열과 같이 길이가 일정하지 않는 키 (key)를 사용하여 해시 값을 생성하고, 이를 배열의 인덱스나 데이터 항목이 저장되어 있는 그룹(버킷)을 찾는 데 사용
그래프 깊이 우선 탐색 (DFS)	그래프에서 지정된 시작 정점으로부터 목적지 정점까지의 경로를 깊이 우선 방식으로 탐색. 목적지까지의 탐색 시간은 상대적으로 빠를 수 있으나, 탐색된 경로가 최단 거리를 보장하지는 않음
그래프 넓이 우선탐색 (BFS)	그래프에서 지정된 시작 정점으로부터 목적지 정점까지의 경로를 넓이 우선 방식으로 탐색. 탐색된 경로는 경유하는 간선수가 최소인 것을 보장함
최단거리 경로 탐색	간선 (edge)에 가중치 (weight)이 지정된 그래프에서 시작 정점으로부터 목적지 정점까지의 경로 중에서 전체 거리가 가장 짧은 경로를 탐색. 탐색된 경로는 최단거리인 것을 보장함
최소비용 신장트리 탐색	주어진 그래프에서 사용되는 간선들의 총 거리 (또는 가중치)가 가장 작으면서 모든 노드 (정점)들을 연결할 수 있는 최소비용 신장트리 (minimum spanning tree)를 탐색



# 순차 탐색 (Sequential Search)



...



# Sequential Search

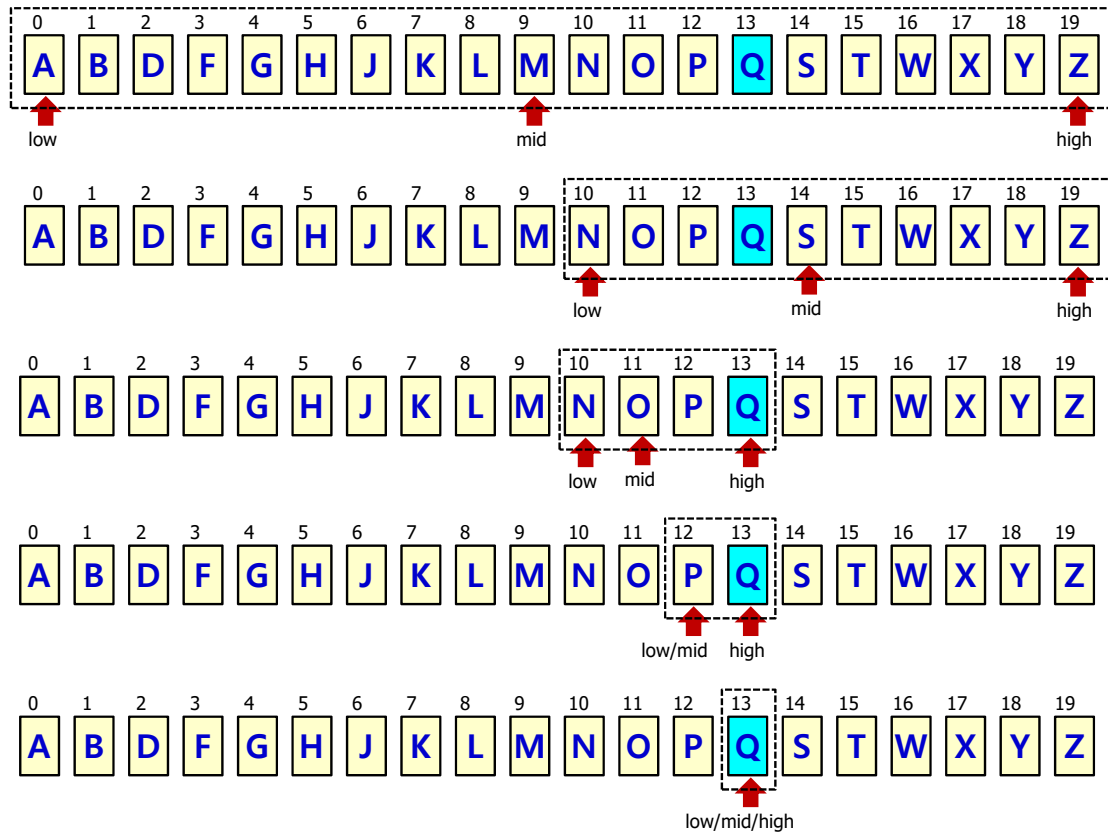
Procedure **Sequential\_Search(L, N, Key)**

1. /\* input argument: non-sortedList L of Length N, Key to find \*/
2. /\* output result: the position (index) of the Key, -1 if not exists \*/
3. idx = 0 /\* index \*/
4. while (idx < N) :
5.     if (L[idx] == Key):
6.         return idx
7.     else:
8.         idx = idx + 1
9. return -1 /\* -1 means the given key was not found in the L \*/
10. end procedure /\* end of Procedure Sequential\_Search() \*/





# 이진 탐색 (Binary Search)



# Binary Search

Procedure **Binary\_Search(L, N, key)**

1. /\* input argument: SortedList L of Length N, key to find \*/
2. /\* output result: the position of the Key, -1 if not exists \*/
3. left = 0; /\* left index \*/
4. right = N-1; /\* right index \*/
5. while (left <= right):
6.     mid = (left + right) / 2
7.     if (L[mid] > key):
8.         right = mid - 1
9.     else if (L[mid] < key):
10.         left = mid + 1
11.     else: /\* L[mid] == key \*/
12.         return mid;
13. return -1
14. end procedure /\* end of Procedure Binary\_Search() \*/



## 순차 탐색과 이진 탐색의 비교

### ◆ 순차탐색 (Sequential Search)

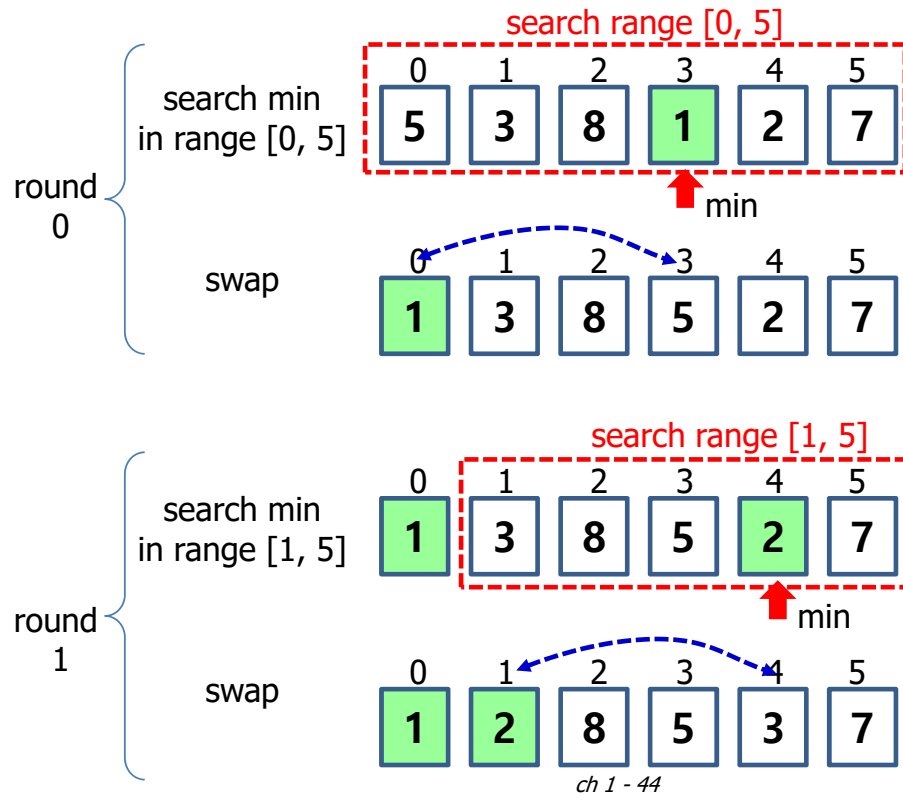
- 탐색 대상 데이터 베이스가 정렬되어 있지 않은 상태에서 맨 처음 항목부터 차례대로 비교하여 탐색 기준 값이 동일한 항목을 찾아냄
- 탐색에 걸리는 시간은 탐색 대상 자료에 포함된 데이터 베이스에 포함된 항목의 개수에 비례하며, 최대  $O(N)$ 가 될 수 있고, 평균적으로  $O(N/2)$  정도의 시간이 걸림

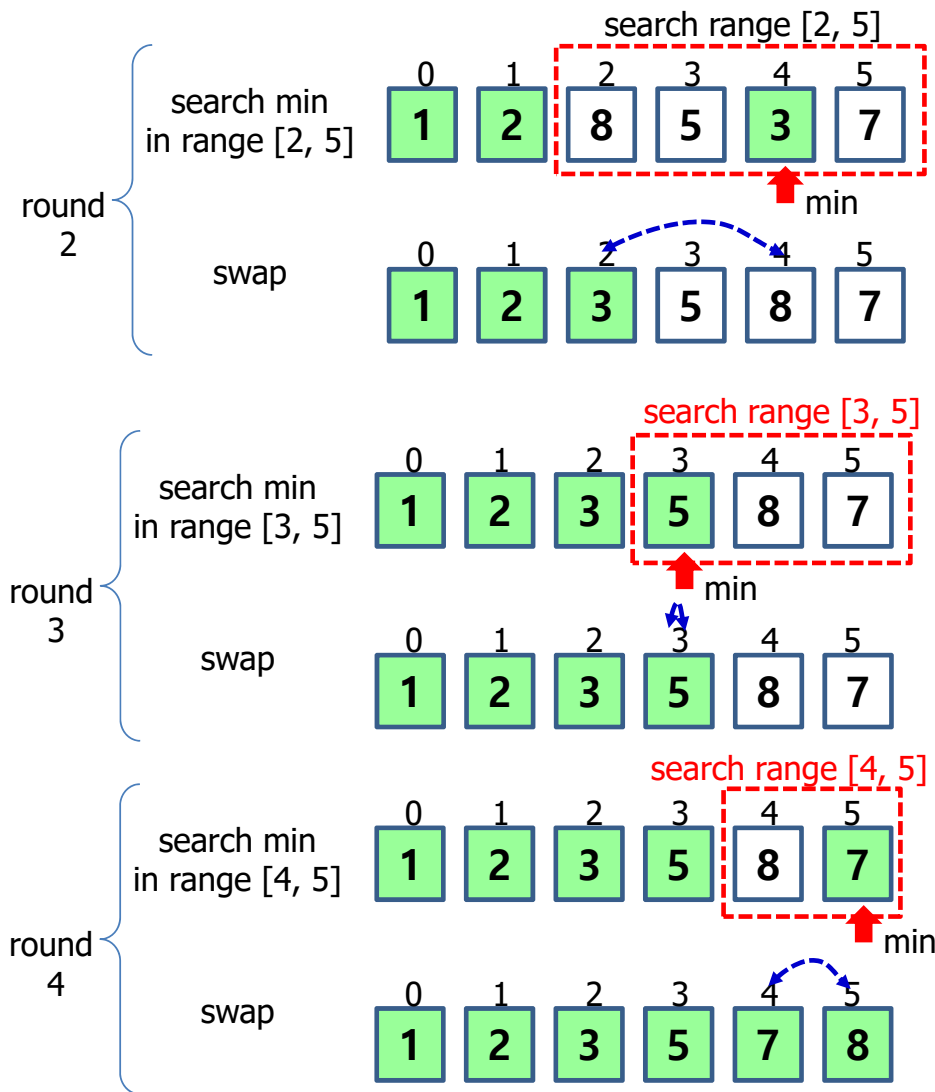
### ◆ 이진탐색 (Binary Search)

- 탐색 대상 데이터 베이스가 정렬되어 있는 경우에 사용되며, 탐색 구간을 절반씩 줄여가며 탐색 기준 값이 동일한 항목을 찾아냄
- 탐색에 걸리는 시간은 탐색 대상 자료에 포함된 데이터 베이스에 포함된 항목 개수에  $O(\log_2 N)$ 에 비례함
- 정렬에 걸리는 시간이 추가적으로 필요함
- 한번 정렬한 후, 많은 이진 탐색이 수행되는 경우에 효율적임

## 선택정렬(selection sort)

- ◆ 선택정렬(selection sort): 정렬이 안된 숫자들 중에서 최소값을 선택하여 배열의 첫 번째 요소와 교환
- ◆ 단계별 진행





# Selection Sorting

Procedure **Selection\_Sort(L, N)**

```
1.  /* input argument: List L of Length N */
2.  /* output result: sorted List */

3.  i = 0
4.  while (i<N-1):
5.      min = L[i]
6.      min_idx = i
7.      j = i + 1
8.      while (j<N):
9.          if (min > L[j]):
10.             min = L[j]
11.             min_idx = j
12.         j = j + 1
13.     if (min_idx != i):
14.         L[min_idx] = L[i]
15.         L[i] = min
16.     i = i + 1
17. end procedure /* end of Procedure Selection_Sort() */
```



## 대표적인 자료구조

# 자료구조 란?

## ◆ 자료구조 (Data Structure)

- 컴퓨터 프로그램으로 문제를 해결하는 알고리즘에서 데이터/정보를 저장(store)하는 기능을 제공하며, 관련 정보를 신속하게 탐색(search)할 수 있는 기능을 제공하여야 함
- 알고리즘과 긴밀한 연관성을 가짐





# 정보의 정리와 검색 (Search)

## ◆ 사전에서의 단어 검색과 도서관에서의 도서 검색

- 사전에서 어떤 어휘 (vocabulary)의 뜻과 이 단어와 동일한 의미의 유사어 (thesaurus)를 신속하게 찾기 위하여 어떤 자료구조와 어떤 알고리즘을 사용할 수 있는가?
- 도서관에서 특정 정보를 가진 도서를 신속하게 찾기 위하여 어떤 자료구조와 어떤 알고리즘을 사용할 수 있는가 ?



(a) 사전에서의 단어 검색

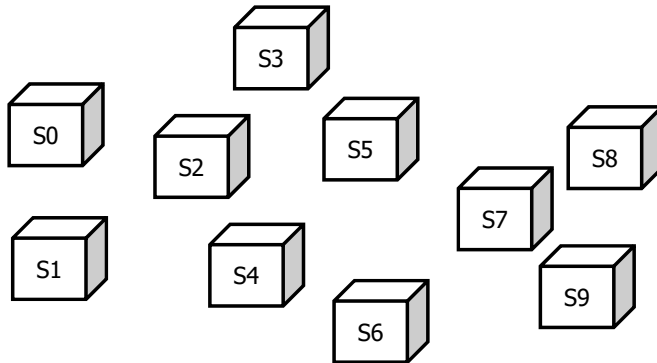


(b) 도서관에서의 도서 검색

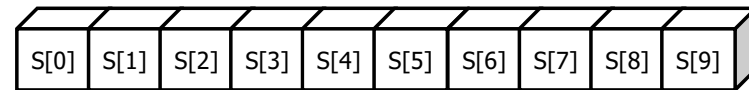
# 배열 (Array)

## ◆ 배열(array)

- 동일한 자료형의 데이터가 여러 개 저장되어 있는 데이터 저장 장소
- 배열 안에 들어있는 각각의 데이터들은 정수로 되어 있는 인덱스(첨자)에 의하여 접근
- 배열을 이용하면 동일한 자료형의 여러 개의 데이터 들을 하나의 이름으로 관리할 수 있다.



(a) 각 원소들이 개별적인 이름을 사용하는 경우

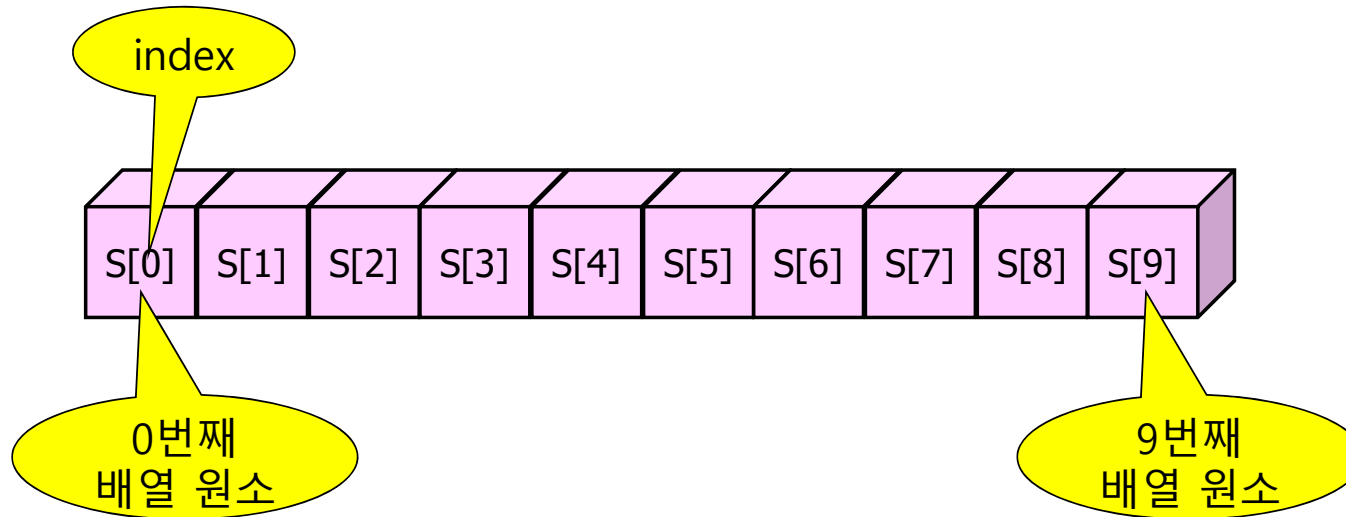


(b) 하나의 이름을 사용하며, 연속적인 메모리 공간에 저장되어 인덱스로 각 원소를 개별적으로 사용하는 배열의 경우

# 배열 원소와 인덱스

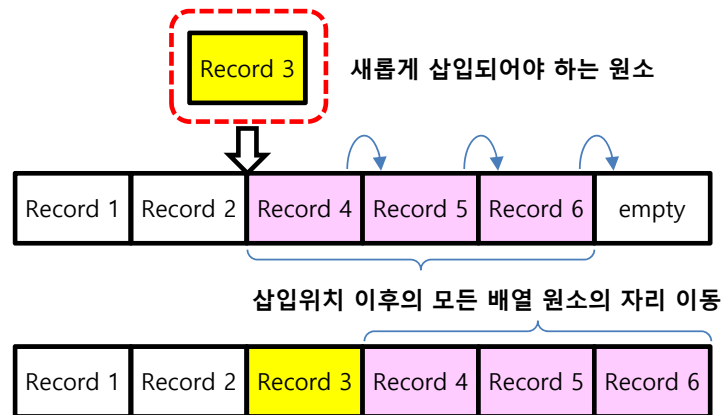
## ◆ 인덱스(index)

- 배열은 하나의 이름을 사용하며, 연속적인 메모리 공간에 저장되어 있는 배열 원소를 인덱스 (첨자, 색인 번호)로 개별적인 접근이 가능
- 총 N개의 배열 원소가 포함되어 있을 경우 인덱스는  $0 \sim N-1$



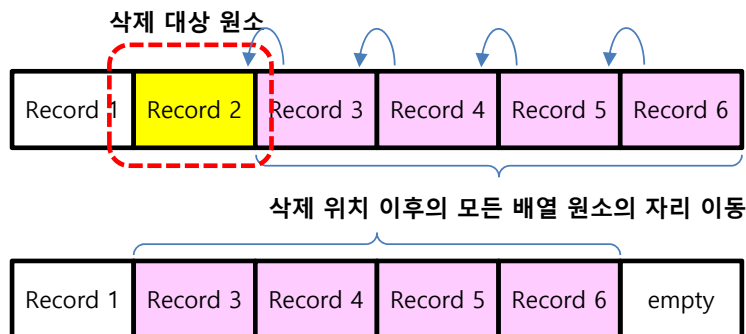
# 배열의 원소 삽입과 삭제 성능 분석

## ◆ 새로운 원소의 삽입



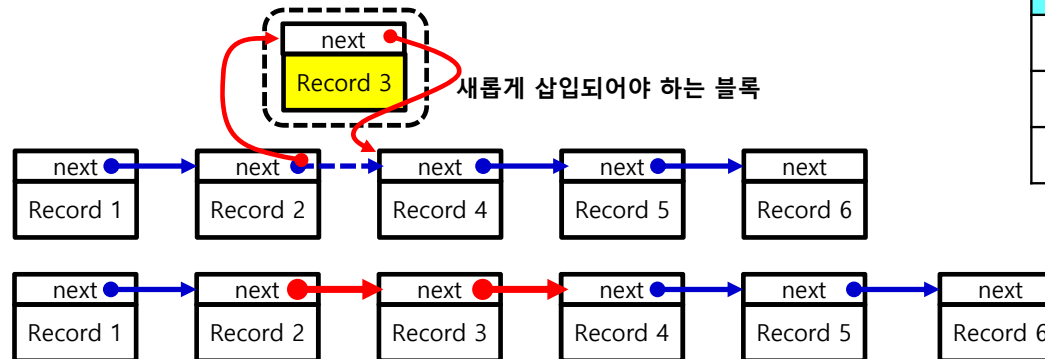
Operation	Complexity in Array
search	$O(n)$
insert	$O(n)$
remove	$O(n)$

## ◆ 배열 원소의 삭제



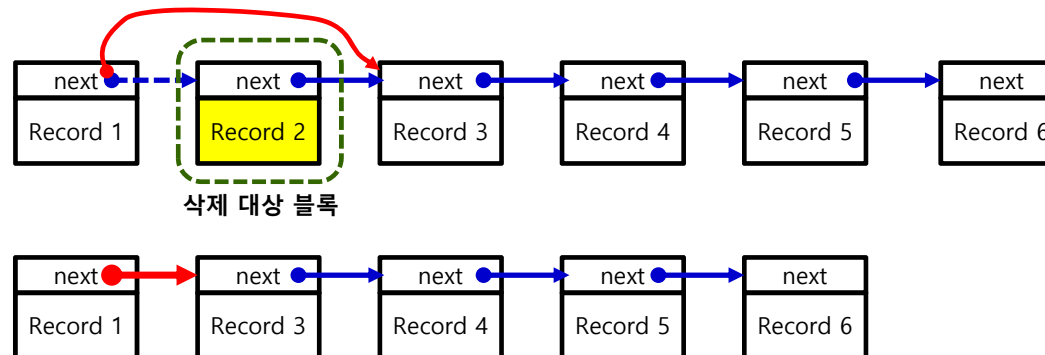
# 연결형 리스트에서의 삽입과 삭제

## ◆ Insert in Linked List



Operation	Complexity in Linked List
search	$O(n)$
insert	$O(1)$
remove	$O(1)$

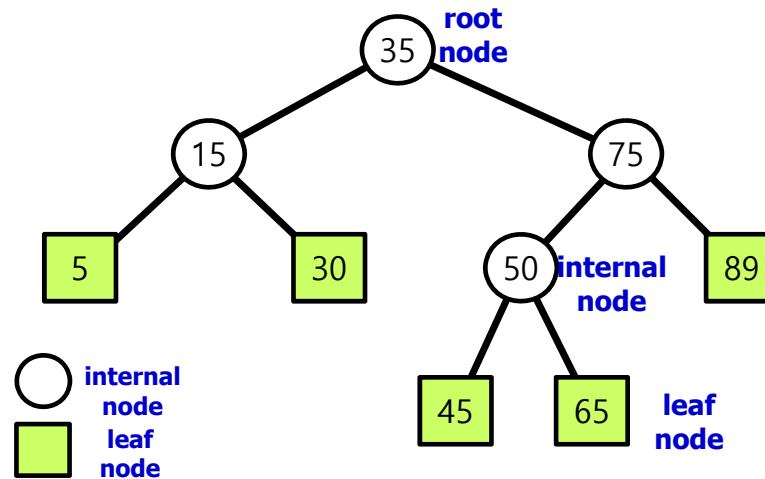
## ◆ Delete/Remove in Linked List



# 이진 탐색 트리 (Binary Search Tree)

## ◆ 이진 탐색 트리 (Binary search tree):

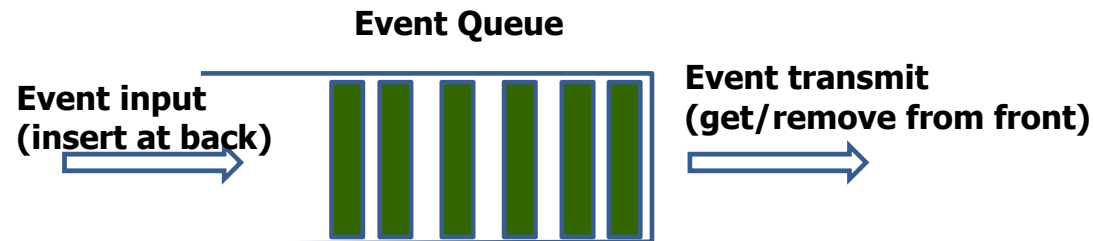
- 데이터 탐색을 쉽게 수행할 수 있도록 구성된 이진 트리
- 각 노드의 왼쪽 서브 트리는 그 노드의 값보다 작은 데이터를 가지는 노드들로만 구성
- 각 노드의 오른쪽 서브 트리는 그 노드의 데이터 값보다 같거나 큰 데이터 값을 가지는 노드들로만 구성



# 큐 (Queue)

## ◆ FIFO Queue

- queue: 선착순 개념 (First In First Out: FIFO)
- **enqueue (insert)** a newly arrived event in the tail of the queue
- **front() get** the event at the front of the queue
- **dequeue(delete)** the acknowledged event from the queue



# 스택 (Stack)

## ◆ LIFO (Last In First Out) Stack

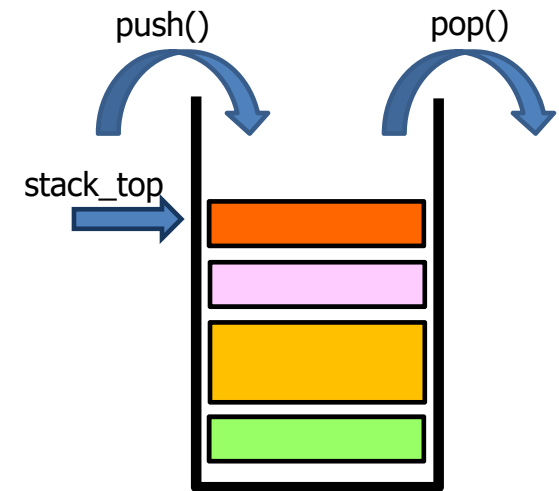
- 부페 식당에서 접시를 쌓아두는 장치와 같이 동작
- 가장 나중에 쌓아둔 접시 (가장 위에 놓인 접시)가 가장 먼저 사용됨

## ◆ Stack의 주요 기능

- **push(Element elm)**: inserts an element
- **Element pop()**: removes the last inserted element

## ◆ Stack의 주요 응용 분야

- 문서 편집에서 되돌리기 (undo)
- 인터넷 웹 검색에서 가장 최근에 검색된 웹 페이지를 다시 검색하고자 할 때





# 파이썬 자료형

## ◆ 파이썬 내장 자료형 (built-in data type)

type		mutable	iterable	remarks
boolean	bool	no	no	
numeric	int	no	no	
	float	no	yes	
	complex	no	yes	
sequence	str	no	yes	
	bytes	no	yes	
	bytearray	yes	yes	
	memoryview	no/yes	yes	
	list	no	yes	
	tuple	no	yes	
	range	no	yes	
mapping	dict	yes	yes	
set	set	yes	yes	
	frozenset	no	yes	



## 대표적인 자료구조

자료구조	특성
단순 배열	컴파일 단계에서 크기가 지정되어 변경되지 않는 크기의 배열
동적 배열	프로그램 실행 단계에서 크기가 지정되며, 프로그램 실행 중에 크기를 변경할 수 있는 배열
구조체 배열	구조체로 배열원소가 지정되는 배열
연결형 리스트	확장성이 있으며, 단일연결형 또는 이중 연결형으로 구성 가능. 다양한 컨테이너 자료구조의 내부적인 자료구조로도 활용됨
스택 (stack)	Last In First Out (LIFO) 특성을 가짐. 배열 또는 연결형 리스트로 구현할 수 있음.
큐 (queue)	First In First Out (FIFO) 특성을 가짐. 배열 또는 연결형 리스트로 구현할 수 있음.
힙 우선 순위 큐 (heap priority queue)	컨테이너 내부에 가장 우선순위가 높은 데이터 항목을 추출할 수 있도록 관리하며, 배열 또는 자기 참조 구조체로 구현할 수 있음
이진 탐색 트리	컨테이너 내부에 포함된 데이터 항목들을 정렬된 상태로 관리하여야 할 때 매우 효율적임. 단순 이진 탐색 트리의 경우 편중될 수 있으며, 편중된 경우 검색 성능이 저하되기 때문에, 밸런싱이 필요함.
해시 테이블 (Hash Table)	컨테이너 자료구조에 포함된 항목들을 문자열 (string) 또는 긴숫자를 키 (key)로 사용하여 관리하여야 하는 경우, key로부터 해시 값을 구하고, 이 해시 값을 배열의 인덱스로 사용함.
Map	key와 항목 간에 1:1 관계가 유지되어야 하는 경우에 사용되며, 해시 테이블을 기반으로 구현할 수 있음
Dictionary	key와 항목 간에 1:N 관계가 유지되어야 하는 경우에 사용되며, 해시 테이블을 기반으로 구현할 수 있음
그래프	정점 (vertex)/노드 (node)로 개체 (object)가 표현되고, 간선 (edge)/링크(link)들을 사용하여 개체 간의 관계를 표현하는 경우에 적합함. 그래프를 기반으로 경로 탐색, 최단 거리 경로 탐색, 신장트리 (spanning tree) 탐색 등에 활용됨.

## 파이썬 프로그래밍 환경 준비

# 파이썬 프로그래밍 언어 개요

## ◆ 파이썬 (Python )

- 네덜란드 Guido van Rossum이 1990년대 초에 개발
- 영국 TV 프로그램 “Monty Python” 제목
- 공개 소프트웨어, general-purpose language
- 객체 지향형 개념을 사용하며, 일반적인 함수도 사용
- 인터프리터 사용 (컴파일러를 사용하여 사전에 실행코드를 만들지 않음)
- 동적 자료형 설정 (dynamically typed): 프로그램에서 사용되는 변수 (variable)의 자료형이 사전에 설정되지 않고, 실제 값이 대입될 때 결정됨
- 다양한 고급 자료형 및 자료구조를 지원: Lists, Tuples, Dictionaries (maps), Sets
- 스크립트 언어로서 간결함
- C/C++, Object-C, Java, Fortran 등의 언어로 개발된 모듈을 쉽게 통합시킬 수 있음



# 파이썬 프로그래밍 환경

## ◆ 파이썬 프로그래밍 환경 설치

- Downloads: <http://www.python.org>
- Documentation: <http://www.python.org/doc/>
- Free book: <http://www.diveintopython.org>



# www.python.org

Welcome to Python.org

python.org

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

```
# Python 3: Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a float
5.666666666666667
>>> 17 // 3 # floor division
5
```

**Intuitive Interpretation**

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `-`, `*` and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Help us raise \$60,000 USD by December 31st! Building the PSF Q4 Fundraiser

**Get Started**  
Whether you're new to programming or an experienced developer, it's easy to learn and use Python.  
[Start with our Beginner's Guide](#)

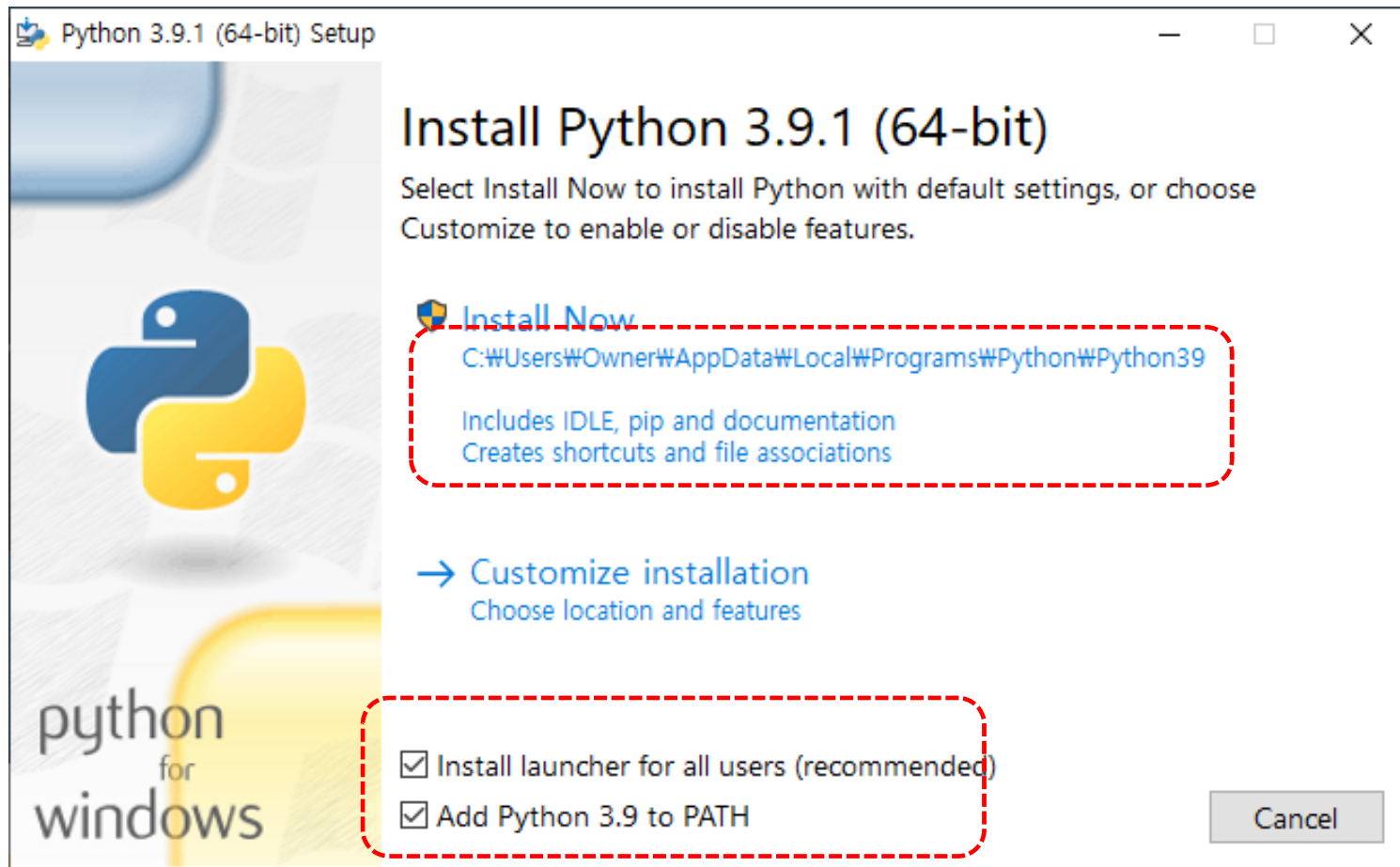
**Download**  
Python source code and installers are available for download for all versions!  
Latest: Python 3.9.1

**Docs**  
Documentation for Python's standard library, along with tutorials and guides, are available online.  
[docs.python.org](#)

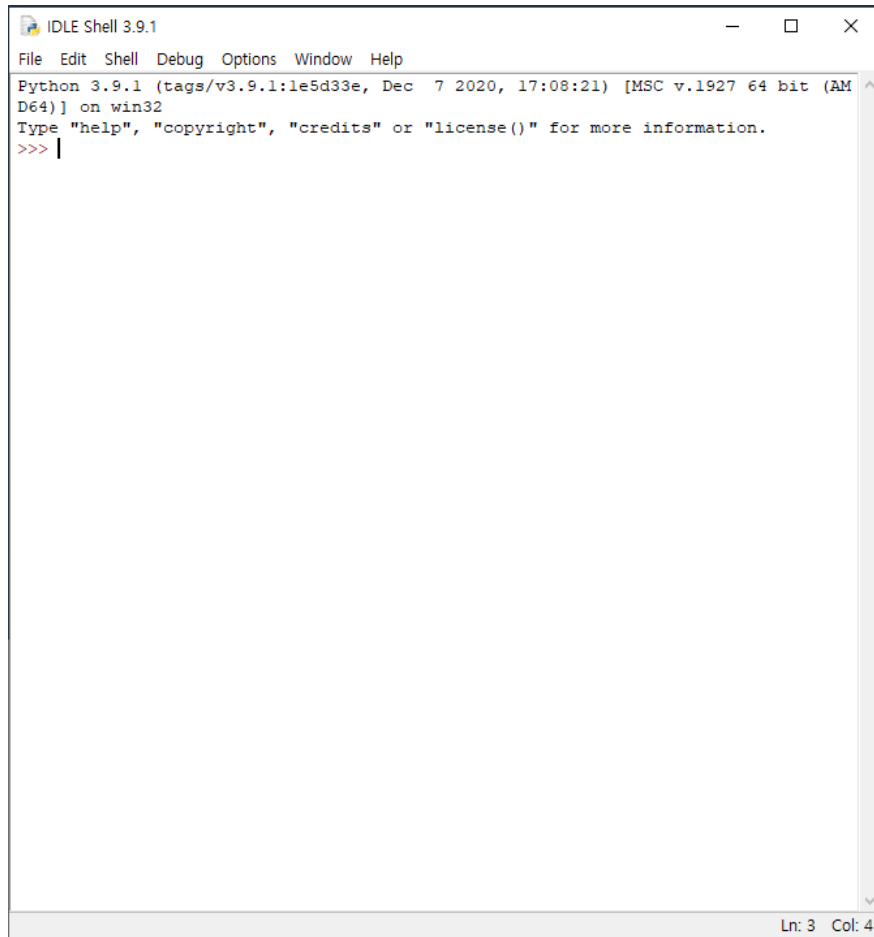
**Jobs**  
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.  
[jobs.python.org](#)



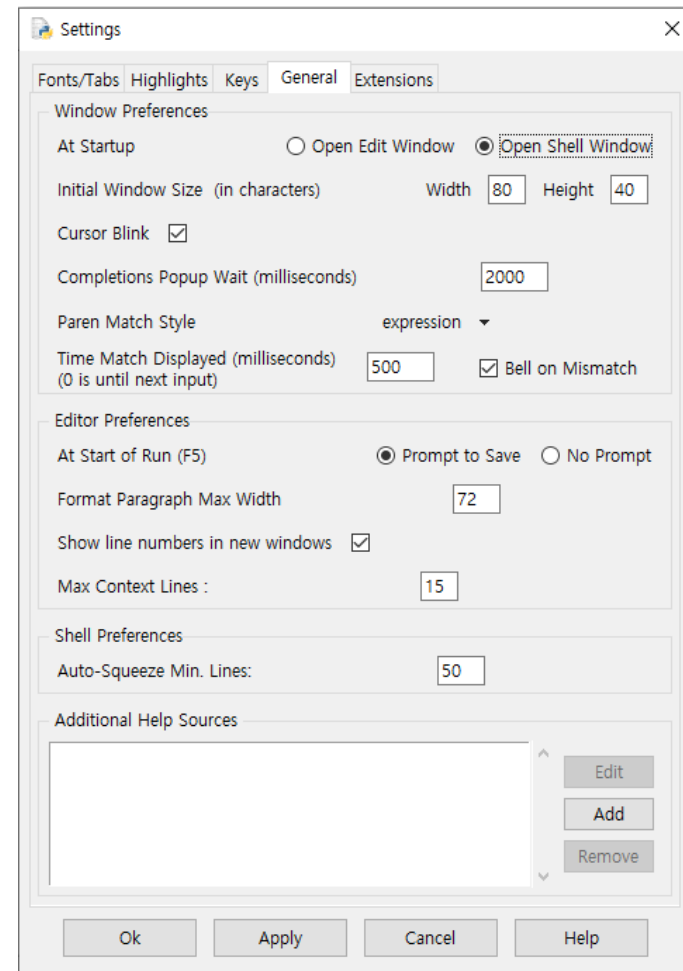
## 파이썬 최신 버전 설치



# 파이썬 셸 (Shell)



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```



Settings

Fonts/Tabs Highlights Keys General Extensions

Window Preferences

At Startup ☐ Open Edit Window ☒ Open Shell Window

Initial Window Size (in characters) Width 80 Height 40

Cursor Blink ☒

Completions Popup Wait (milliseconds) 2000

Paren Match Style expression

Time Match Displayed (milliseconds) 500 ☒ Bell on Mismatch

Editor Preferences

At Start of Run (F5) ☒ Prompt to Save ☐ No Prompt

Format Paragraph Max Width 72

Show line numbers in new windows ☒

Max Context Lines : 15

Shell Preferences

Auto-Squeeze Min. Lines: 50

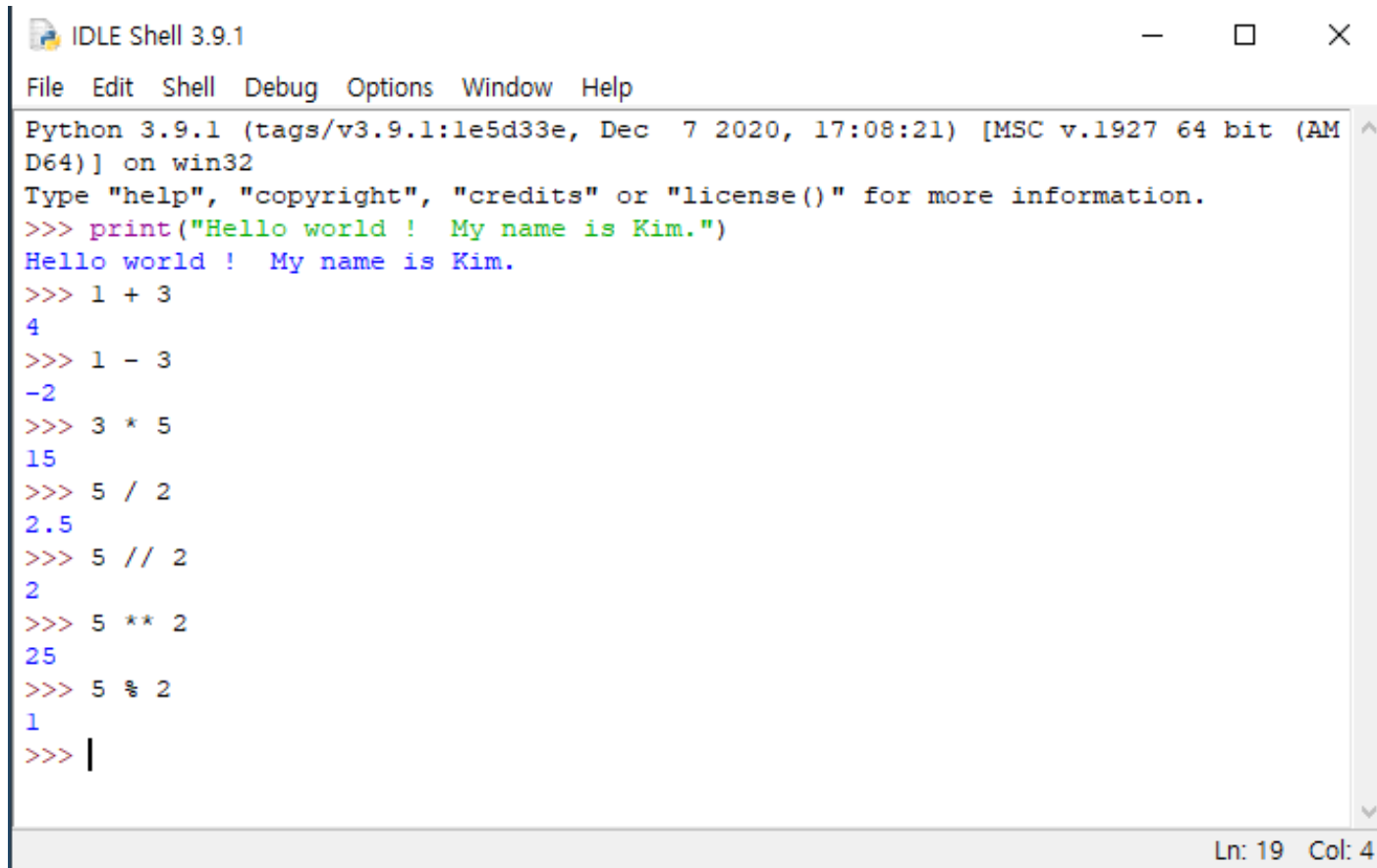
Additional Help Sources

Edit Add Remove

Ok Apply Cancel Help



## 파이썬 셸 (Shell)에서의 간단한 프로그램 실행



The screenshot shows the IDLE Shell 3.9.1 window. The title bar reads 'IDLE Shell 3.9.1'. The menu bar includes 'File', 'Edit', 'Shell', 'Debug', 'Options', 'Window', and 'Help'. The main text area displays the following content:

```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Hello world ! My name is Kim.")
Hello world ! My name is Kim.
>>> 1 + 3
4
>>> 1 - 3
-2
>>> 3 * 5
15
>>> 5 / 2
2.5
>>> 5 // 2
2
>>> 5 ** 2
25
>>> 5 % 2
1
>>> |
```

The status bar at the bottom right indicates 'Ln: 19 Col: 4'.

# 파이썬 프로그래밍 개요 (1)

## ◆ 파이썬 프로그래밍의 특징

- 프로그램의 구역을 들여쓰기 (indentation)으로 구분
- 편리한 시퀀스 자료형 제공
  - Strings '...': 문자열, 문자의 배열로 처리 가능
  - Lists [...]: 다양한 자료형의 원소들을 리스트로 구성
  - Tuples(...): 관련항목들을 (name, type, value)와 같은 형식으로 묶어서 사용하도록 함
- 시퀀스 자료형에 대하여 편리한 인덱싱과 슬라이싱 기능 제공
- 프로그램의 기능별로 함수를 구성하여 모듈화 시킬 수 있으며, 다양한 응용 프로그램에서 재사용 가능하게 함
- Java 프로그램 환경과 유사한 예외처리 (exception handling)
- C++ 및 Java의 객체 지향형 (object-oriented) 프로그래밍 지원
- 시퀀스 자료형에 대한 반복자 (iterator)를 제공하며, 코루틴 개념의 제네레이터 (generator) 함수 제공

## 파이썬 프로그래밍 개요 (2)

### ◆ 파이썬 프로그래밍의 장점?

- 스크립트 기반의 프로그래밍으로 쉽게 배울 수 있음
- “Pythonic” 스타일은 매우 간결한 프로그램 구성 가능
- 객체 지향형 프로그래밍의 장점을 활용하지만,  
C++나 Java보다 쉽게 사용할 수 있는 환경 제공
  - 파이썬에서 사용하는 모든 값과 개체들은 객체 (object)임
- 동적 자료형 (dynamic typing)을 사용하여 프로그램의 일반화/추상화의 장점을 활용
  - 시퀀스, 딕셔너리, 셀, 튜플 등의 고급 자료형을 다양한 기본 자료형에 활용할 수 있게 함
  - 파이썬 고급 자료형을 사용하여 복잡한 응용 자료구조 및 알고리즘을 일반화 (generalized)/추상화 (abstraction) 형식으로 구현 가능



## 파이썬 프로그래밍 개요(3)

### ◆ 동적 자료형 설정 (Dynamic typing)

#### ● 정적 자료형 설정 : C/C++, Java

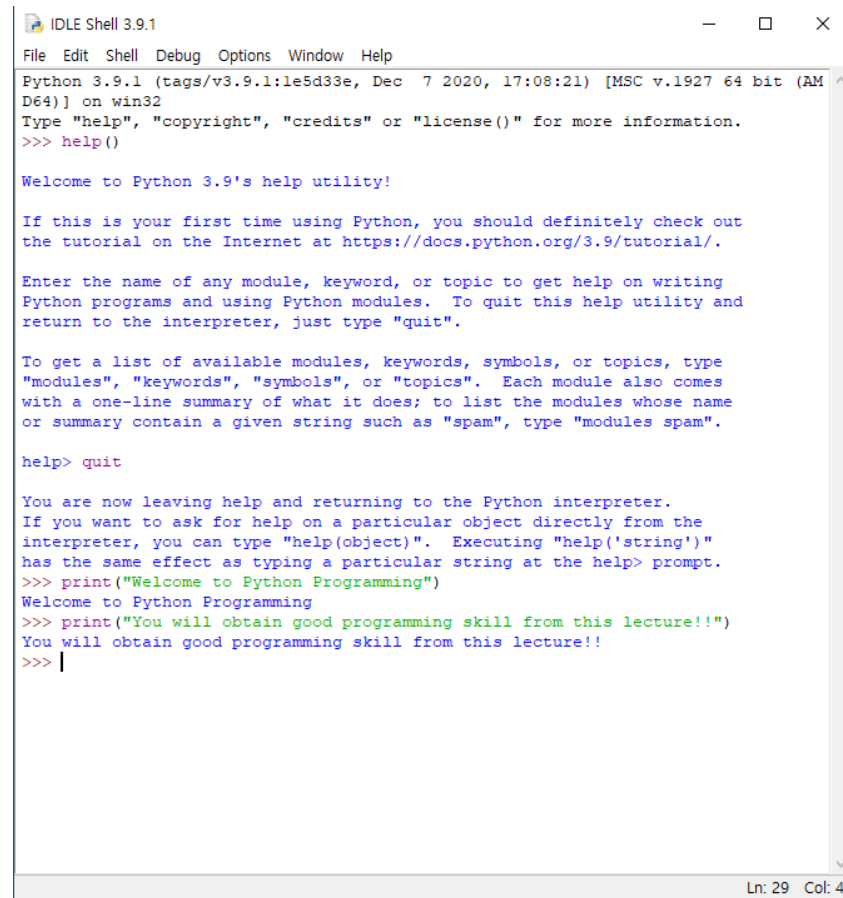
- 프로그램에서 사용되는 변수(variable)들의 자료형은 소스코드 작성 단계에서 미리 설정 됨
- 변수들의 사용에서 정확한 자료형의 사용을 엄격하게 관리함
- 함수호출이나 변수를 사용한 대입 (assignment) 등에서 자료형이 서로 다른 경우 문법 오류 발생

#### ● 동적 자료형: Python

- 변수 (variable)들은 처음 대입이 될 때 생성되며, 대입되는 데이터에 따라 자료형이 동적으로 결정됨
- 하나의 변수가 다양한 자료형의 객체로 대입될 수 있음
- 모든 자료형이 동일한 방식으로 관리됨
- **문제점 (main drawback)**: 자료형에 관련된 오류가 실행되는 단계에서만 확인될 수 있음

# 파이썬 프로그래밍 환경 (1)

## ◆ 파이썬 셸 (shell)



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> help()

Welcome to Python 3.9's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.9/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

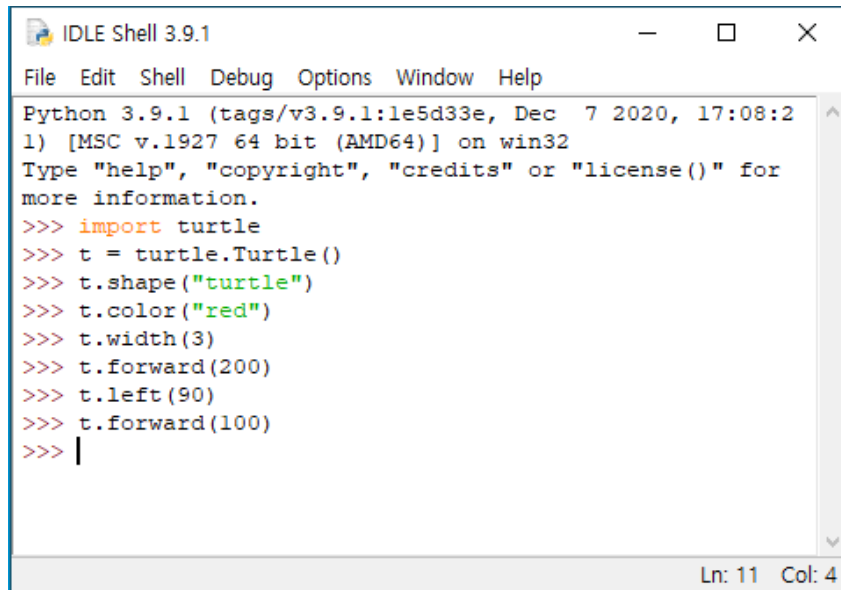
To get a list of available modules, keywords, symbols, or topics, type
"modules", "keywords", "symbols", or "topics".  Each module also comes
with a one-line summary of what it does; to list the modules whose name
or summary contain a given string such as "spam", type "modules spam".

help> quit

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)".  Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>> print("Welcome to Python Programming")
Welcome to Python Programming
>>> print("You will obtain good programming skill from this lecture!!")
You will obtain good programming skill from this lecture!!
>>> |
```

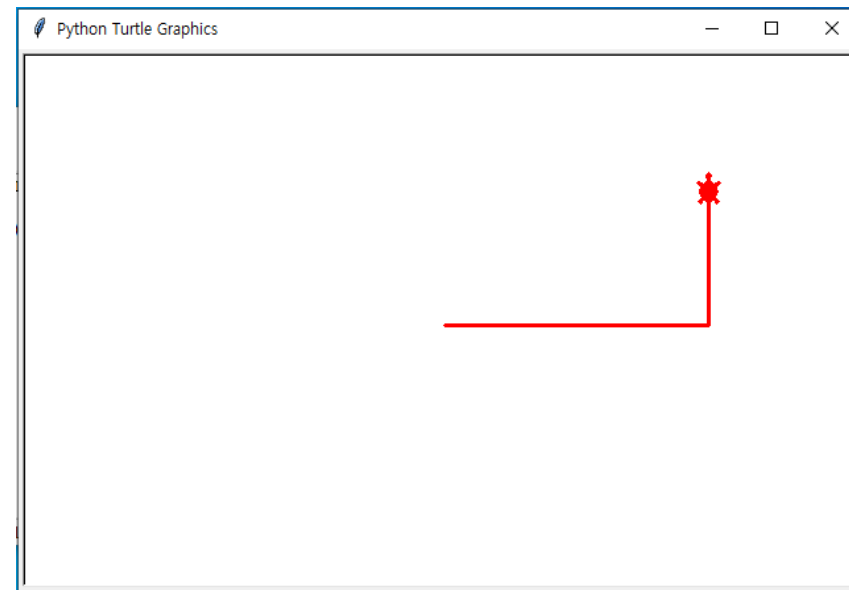
# 파이썬 쉘 환경에서의 프로그래밍

## ◆ turtle graphic



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> t = turtle.Turtle()
>>> t.shape("turtle")
>>> t.color("red")
>>> t.width(3)
>>> t.forward(200)
>>> t.left(90)
>>> t.forward(100)
>>> |
```

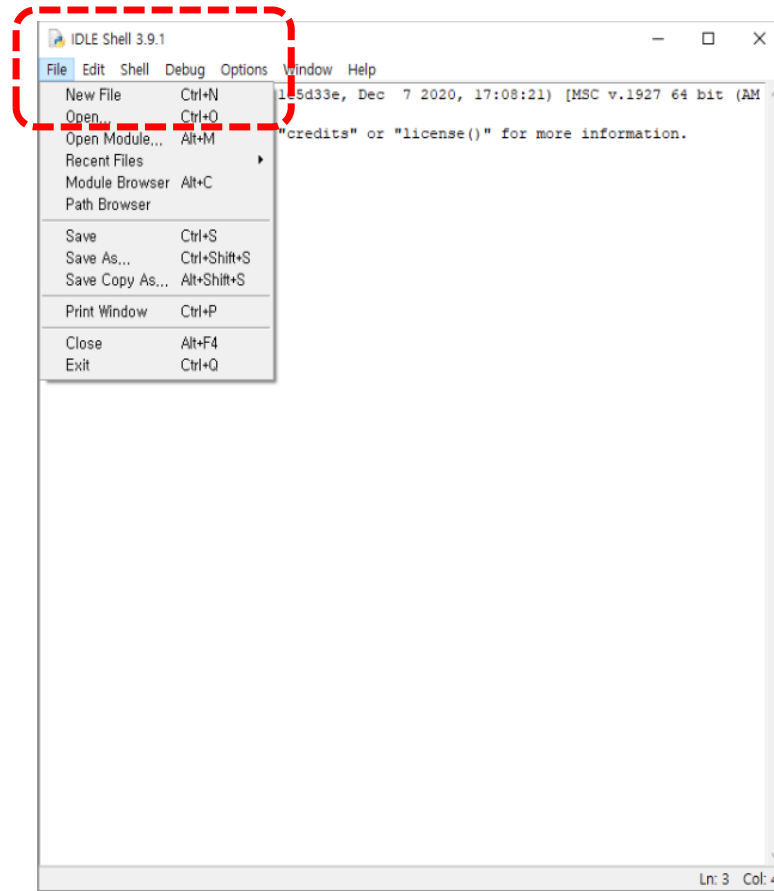
Ln: 11 Col: 4



# Python Shell

## ◆ File Menu

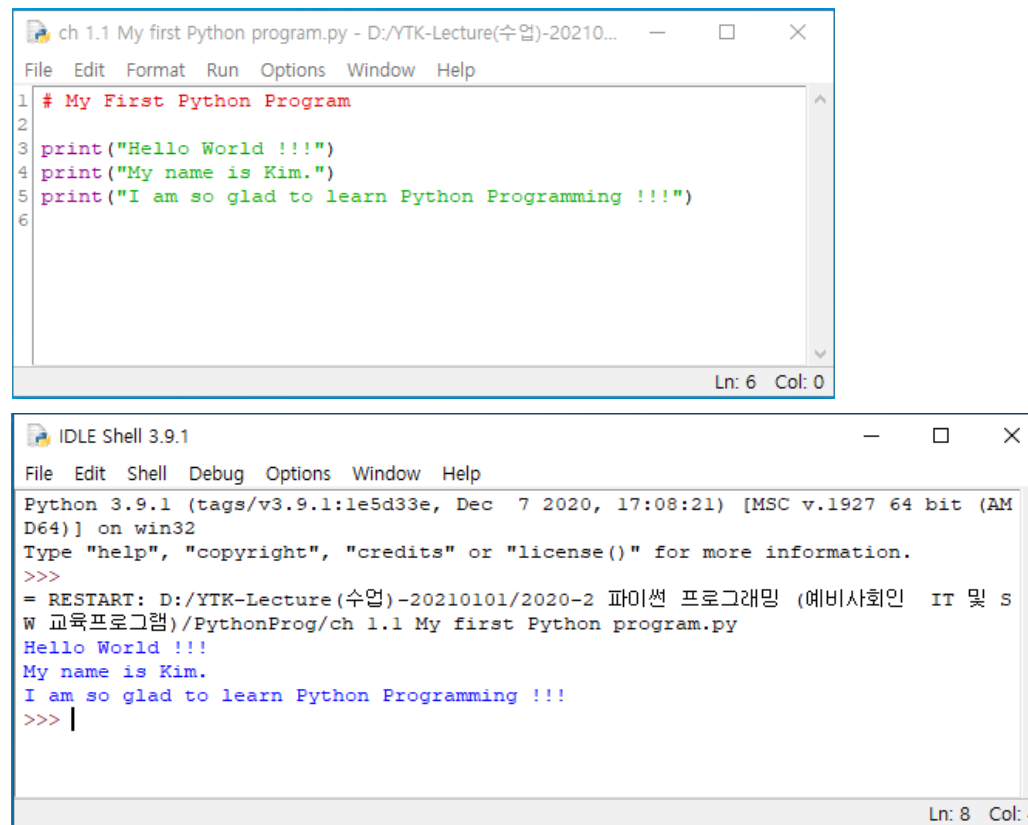
- New File



# IDLE에서 파이썬 프로그램 작성 및 실행

## ◆ My First Python Program

- Save as (Ctrl+shift+S)
- Save (Ctrl+S)
- Run module (F5)



The screenshot displays the IDLE Python IDE interface. The top window, titled 'ch 1.1 My first Python program.py', contains the following Python code:

```
1 # My First Python Program
2
3 print("Hello World !!!")
4 print("My name is Kim.")
5 print("I am so glad to learn Python Programming !!!")
6
```

The bottom window, titled 'IDLE Shell 3.9.1', shows the output of the executed program:

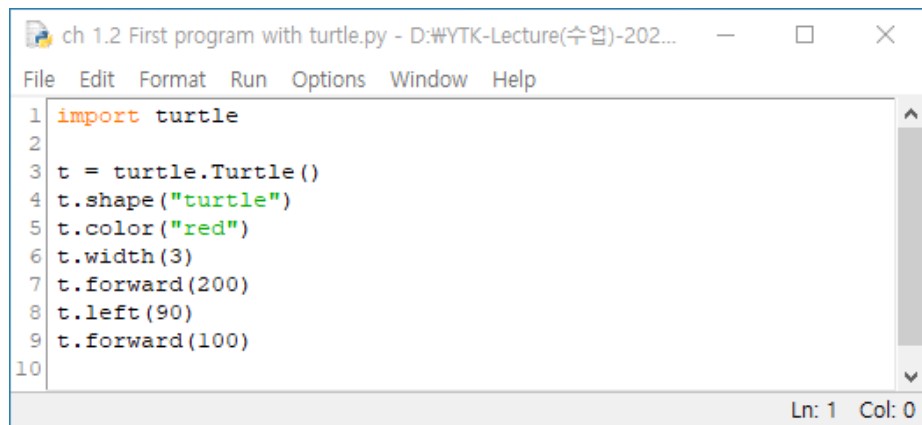
```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/YTK-Lecture(수업)-20210101/2020-2 파이썬 프로그래밍 (예비사회인 IT 및 S
W 교육프로그램)/PythonProg/ch 1.1 My first Python program.py
Hello World !!!
My name is Kim.
I am so glad to learn Python Programming !!!
>>> |
```



# IDLE에서의 Turtle Graphic 예제

## ◆ My First Python Turtle Graphic Program

- Save as (Ctrl+shift+S)
- Save (Ctrl+S)
- Run module (F5)



```
ch 1.2 First program with turtle.py - D:\WYTK-Lecture(수업)-202...  
File Edit Format Run Options Window Help  
1 import turtle  
2  
3 t = turtle.Turtle()  
4 t.shape("turtle")  
5 t.color("red")  
6 t.width(3)  
7 t.forward(200)  
8 t.left(90)  
9 t.forward(100)  
10  
Ln: 1 Col: 0
```



# 파이썬 IDLE 환경에서의 프로그램 구현 및 실행

## ◆ Simple script editing, save and execute

- IDLE [File] → [New File]
- [File] → [Save]
- [Run] → [Run Module]
- [File] → [Open]

## ◆ 단축키

- 저장 (save) : Ctrl + S
- 이름지정 저장 (save as) : Ctrl+Shift+S
- 실행 (run module) : F5

## References

- [1] 김영탁, 자료구조와 알고리즘을 함께 배우는 파이썬 프로그래밍, 홍릉과학 출판사, 2021. 8. (수정판2쇄)
- [2] Kent D. Lee, Steve Hubbard, Data Structures and Algorithms with Python, Springer, 2015.
- [3] Nikhil Ketakar, Deep Learning with Python, Apress, 2017.
- [4] Antonio Gulli, Amita Kappor, Sujit Pal, Deep Learning with TensorFlow 2 and Keras, Packt, 2019.
- [5] 천인국, 어서와 파이썬은 처음이지, 인피니티, 2018.
- [6] 김동근, 쉽게 배우는 파이썬 프로그래밍, 가메출판사, 2016.
- [7] 천인국, 두근두근 파이썬, 생능출판사, 2017.
- [8] 한혁수, 창의적 프로그래밍을 위한 파이썬, 생능출판사, 2016.
- [9] Python Software Foundation, <https://www.python.org/>.
- [10] Python Tutorial, <https://docs.python.org/3/tutorial/>.
- [11] Summary of Turtle Methods, <http://interactivepython.org/runestone/static/IntroPythonTurtles/Summary/summary.html>.
- [12] 김영탁, 자료구조와 알고리즘을 함께 배우는 C 프로그래밍, 배움터, 2019. 2.
- [13] 김영탁, 자료구조와 알고리즘을 함께 배우는 C++ 프로그래밍, 배움터, 2019. 2.



# Homework 1

- 1.1 원의 반지름을 입력받고, 이 원의 넓이와 원둘레를 계산하여 출력하는 알고리즘을 유사코드 (pseudo code)로 작성하라. (파이썬 프로그램 소스코드와 실행 결과를 포함하면 50% bonus 점수 추가)
- 1.2 직사각형의 가로 (width)와 세로(length)를 입력받고, 이 크기의 사각형을 지정된 중심 위치에 그리는 알고리즘을 유사코드 (pseudo code)로 작성하라. (파이썬 프로그램 소스코드와 Turtle 그래픽으로 출력한 결과를 포함하면 50% bonus 점수 추가)
- 1.3 삼각형의 밑변 (base)과 높이 (height)를 입력받고, 이 크기의 삼각형을 지정된 중심 위치에 그리는 알고리즘을 유사코드 (pseudo code)로 작성하라. (파이썬 프로그램 소스코드와 Turtle 그래픽으로 출력한 결과를 포함하면 50% bonus 점수 추가)
- 1.4 한 번의 길이를 입력받고, 5개의 꼭지점을 가지는 별 (star)을 그리기 위한 알고리즘을 유사코드 (pseudo code)로 작성하라. (파이썬 프로그램 소스코드와 Turtle 그래픽으로 출력한 결과를 포함하면 50% bonus 점수 추가)
- 1.5 10개의 정수 데이터를 차례대로 입력 받아 배열이나 리스트에 저장하고, 입력 데이터 중 가장 큰 값, 가장 작은 값, 평균값을 계산하는 알고리즘을 유사코드 (pseudo code)로 작성하라. (파이썬 프로그램 소스코드와 실행 결과를 포함하면 50% bonus 점수 추가)

