

“본 강의 동영상 및 자료는 대한민국 저작권법을 준수합니다. 본 강의 동영상 및 자료는 상명대학교 재학생들의 수업목적으로 제작·배포되는 것이므로, 수업목적으로 내려받은 강의 동영상 및 자료는 수업목적 이외에 다른 용도로 사용할 수 없으며, 다른 장소 및 타인에게 복제, 전송하여 공유할 수 없습니다. 이를 위반해서 발생하는 모든 법적 책임은 행위 주체인 본인에게 있습니다.”

Python 개발환경 설치하기

- 지능정보공학설계, Week 2 -

이 지 항 (Ph.D.)

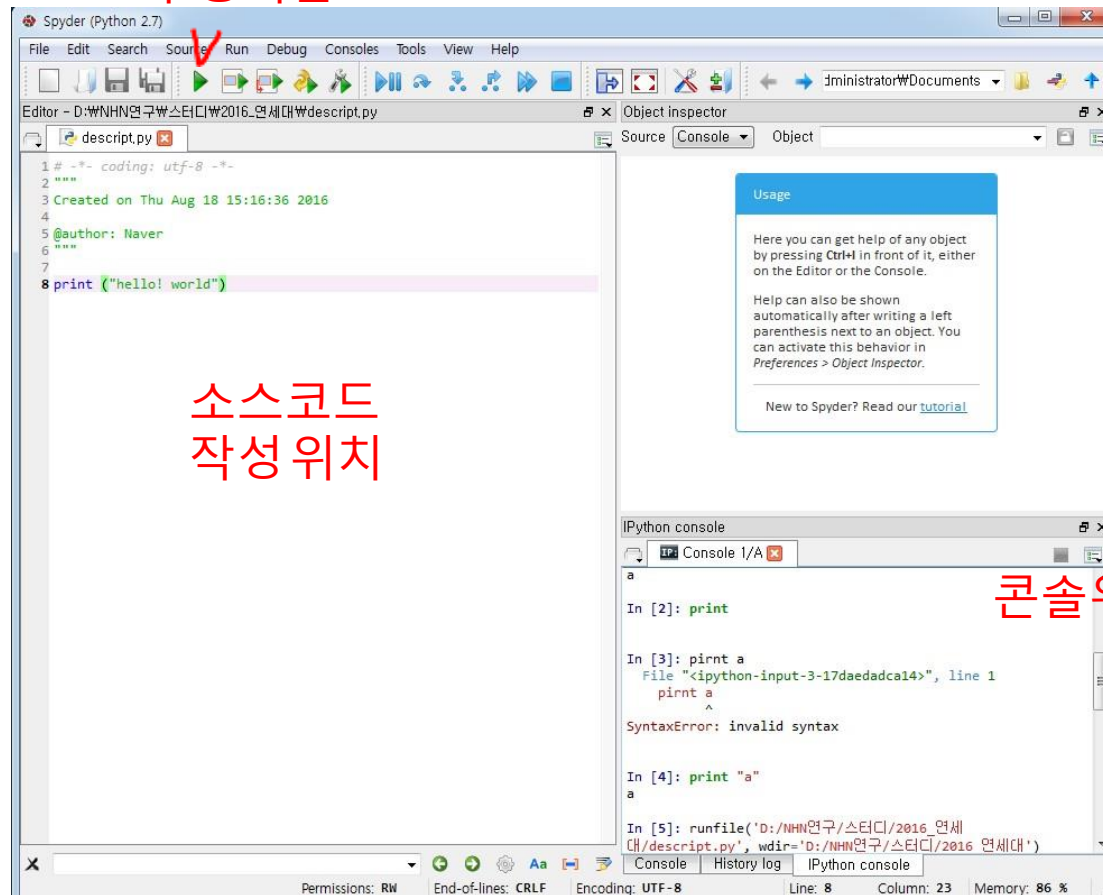
Assistant Professor
Department of Human-Centered AI
Sangmyung University, Seoul, KR
(jeehang@smu.ac.kr)

- 파이썬 통합환경인 아나콘다를 설치한다
 - 통합환경: 소스코드를 작성하고 수행할 수 있는 환경
 - <https://www.anaconda.com/products/individual>
 - 수강생 운영체제에 맞는 버전을 설치하면 됩니다.

Anaconda Installers		
Windows 	MacOS 	Linux 
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (457 MB)	64-Bit Graphical Installer (435 MB)	64-Bit (x86) Installer (529 MB)
32-Bit Graphical Installer (403 MB)	64-Bit Command Line Installer (428 MB)	64-Bit (Power8 and Power9) Installer (279 MB)

- Anaconda → Spyder를 선택해서 실행 가능

코드수행버튼




소스코드
작성위치

콘솔위치

- 또는 pyCharm을 설치해서 사용 가능합니다
 - <https://www.jetbrains.com/pycharm/download/#section=windows>

PyCharmComing in 2021.1 [What's New](#) [Features](#) [Learn](#) [Buy](#) [Download](#)



Version: 2020.3.3
Build: 203.7148.72
27 January 2021

[System requirements](#)
[Installation Instructions](#)
[Other versions](#)

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

Free trial

Community

For pure Python development

[Download](#)

Free, open-source

- Hello world 출력하기
 - 되든 안되든 일단 한 번 시도해 봅시다.
 - Spyder 또는 pyCharm 둘 중 하나를 사용해서 시도해 봅시다.
 - 못해도 괜찮습니다.
 - 코드는 아래와 같습니다.

```
print ("hello world!")
```

- 잘 안되면 최대한 검색해서 방법을 찾아봅시다.

인공지능 에이전트란 무엇인가?

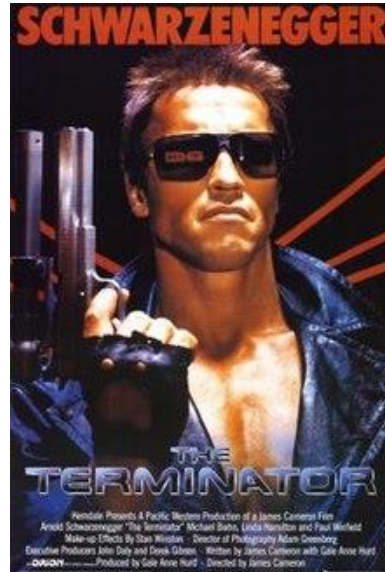
- 지능정보공학설계, Week 2 -

이 지 항 (Ph.D.)

Assistant Professor
Department of Human-Centered AI
Sangmyung University, Seoul, KR
(jeehang@smu.ac.kr)



Blade runner (1982)



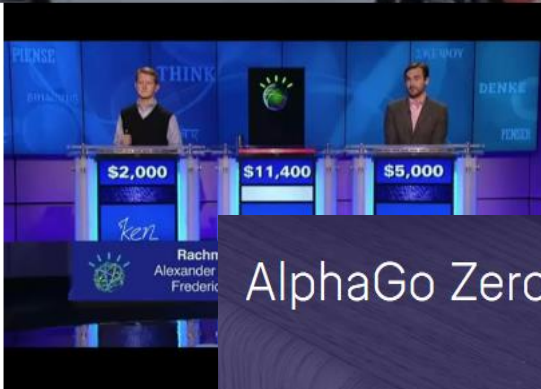
Terminator (1984)



A.I. (2001)



Her (2013)

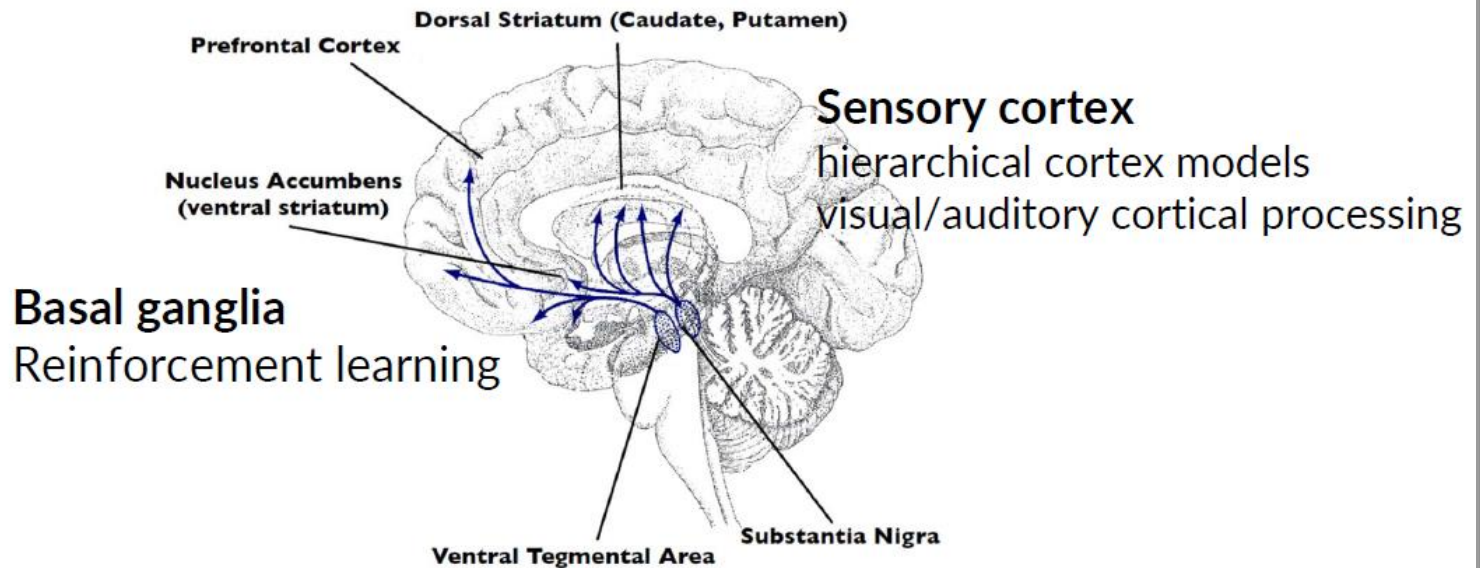


AlphaGo Zero: Learning from scratch



Prefrontal cortex

value computation, cognitive control
social decision making, collective intelligence



[Image from S.W. Lee; Lecture slide, 2018]

“The exciting new effort to make computers think ...
machines with minds, in the full and literal sense”
(Haugeland, 1985)

“The art of creating machines that perform functions
that require intelligence when performed by people”
(Kurzweil, 1990)

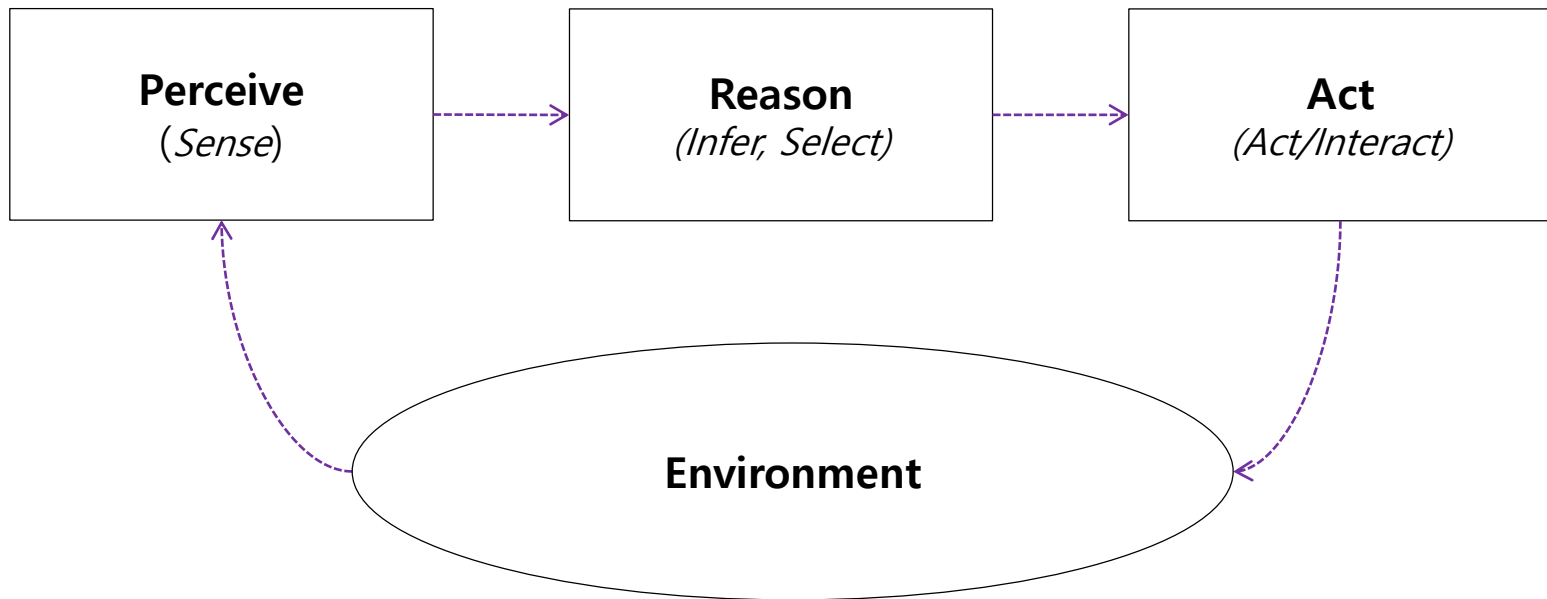
**“The study of the computations that make it
possible to perceive, reason, and act”**
(Winston, 1992)

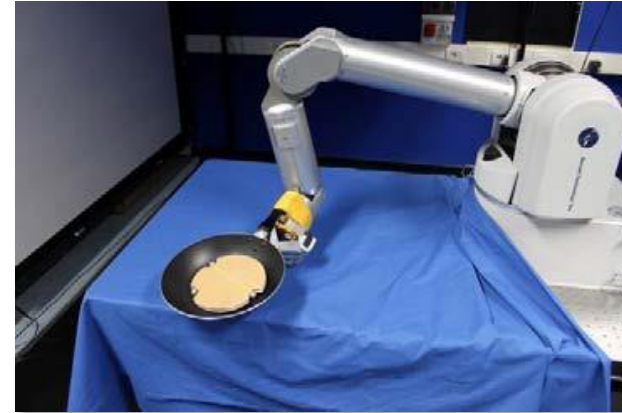
“A field of study that seeks to explain and emulate
intelligent behavior in terms of computational
processes” (Schalkoff, 1990)

“The branch of computer science that is concerned
with the automation of ***intelligent behavior***”
(Luger and Stubblefield, 1993)

“The automation of activities that we associate with
human thinking, activities such as decision-making,
problem solving, learning ...”
(Bellman, 1978)

[Ref: Artificial Intelligence - A modern approach]



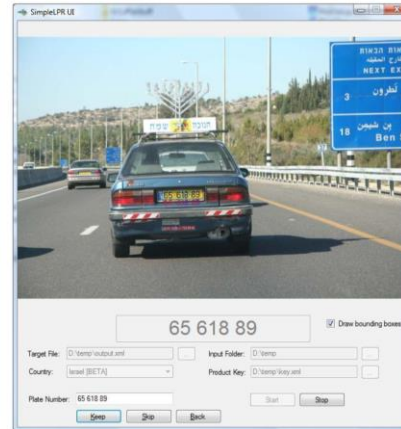


We might be able to say “YES”

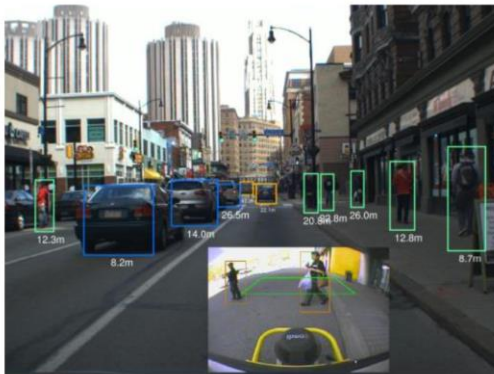
Agent primitive	Input	Output
Perceive	Sensor data	Sensed Info
Reason	Sensed Info Cognitive knowledge	Directives
Act	Sensed Info Directives	Commands



Text recognition






Speech Recognition




Computer vision

People You May Know

-  **Alexander H. Levis**, University Professor of Electrical, [Connect](#)
-  **Doo-Hwan Bae**, -- [Connect](#)
-  **Paul Davidsson**, Professor at Malmö University [Connect](#)

[See more »](#)

Ads You May Be Interested In

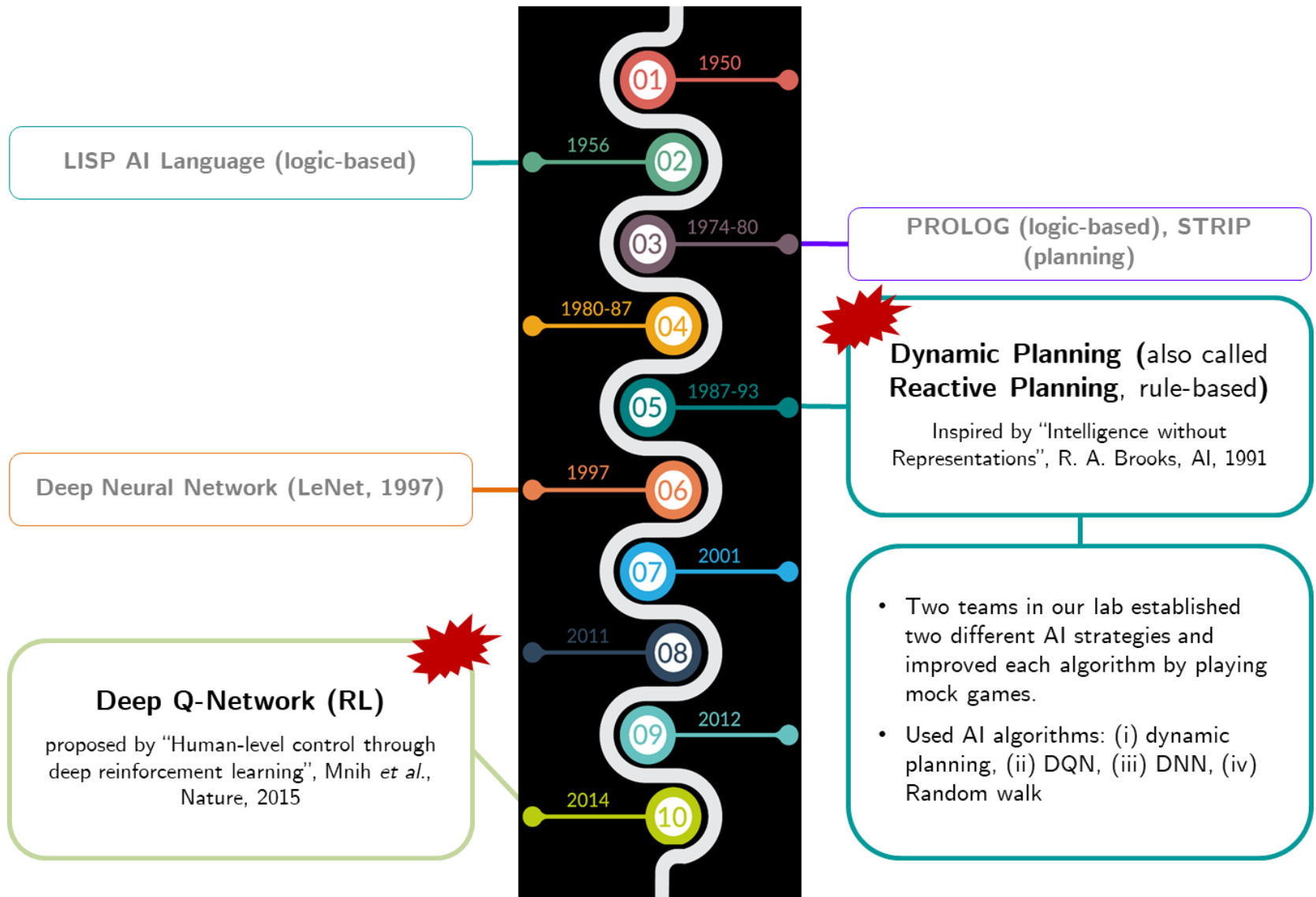
-  **The-CCP Plenum Deciphered**
Tsinghua-INSEAD EMBA Master Class on Mar. 7, Seoul.Discuss

Recommendation

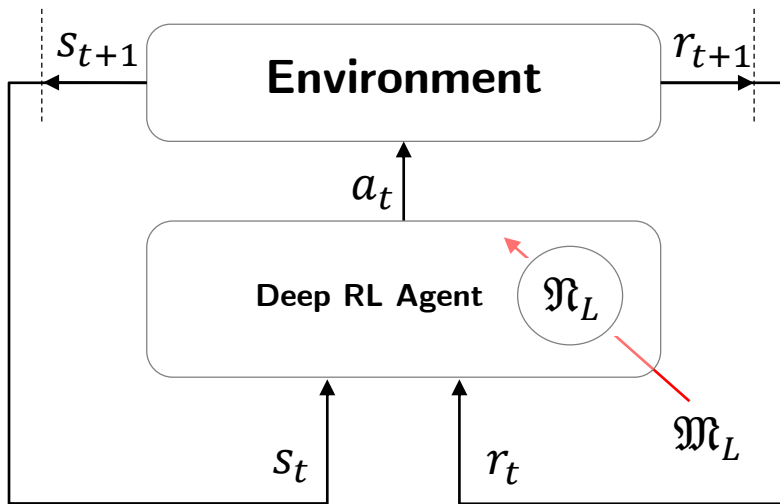


Even in Art! [Figure: L. Gatys et. al, 2015]

Living with AI (in the near future)

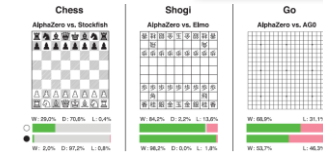
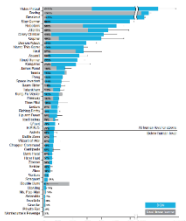
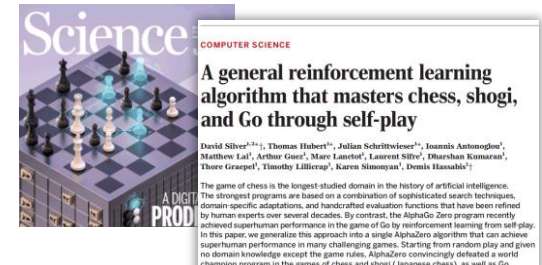


Super human performance of AI



$$Q^*(s, a) = E_{(s, a, s')} \left[R + \gamma \max_{a'} Q^*(s', a') \right]$$

$$= \sum_{s'} P(s, a, s') \left(R + \gamma \max_{a'} Q^*(s', a') \right)$$



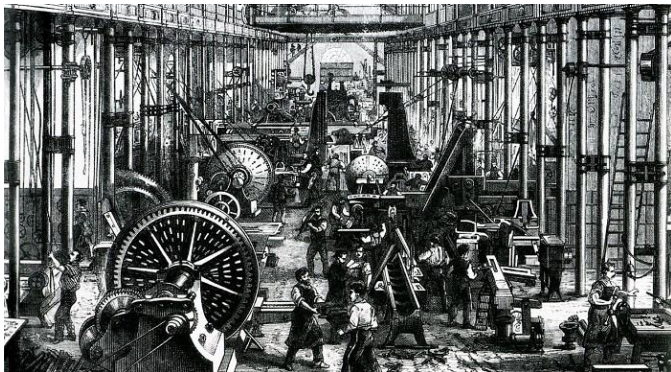
✓ DQN (2015)

✓ Alpha Go (2016)

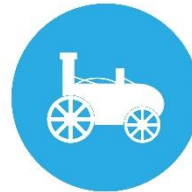
✓ Alpha Zero (2018)

Another Industrial Revolution?

© J.Lee. 2022

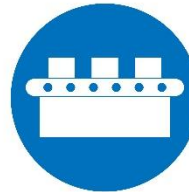


1st revolution



Mechanization, steam
and water power

2nd revolution



Mass production and
electricity

3rd revolution



Electronic and IT
systems, automation

4th revolution



Cyber physical
systems



labour on information

- Who made a decision
 - using what kind of knowledge – expertise
 - how long was the knowledge developed from where – reliability
 - did it work well – efficiency
- Big data that is very hard to analyse in a second, but machine can do
 - What happens if all accumulated knowledge is encoded in a form that machine can compute?
 - A matter of computing power and reliability of machine
 - Who or what can inspect/amend/govern the machines?

English Spanish French English - detected Korean English Spanish Translate

Hope you have a great time in this class. This is truly enjoyable. I look forward to seeing your brilliant final outcome in the final draft of your

이 수업에서 즐거운 시간 보내길 바랍니다. 이것은 정말 즐겁습니다. 나는 당신의 종이의 마지막 초안에서 당신의 화려한 최종 결과를 보길 고대합니다.

i sueob-eseo jeulgeoun sigan bonaegil balabnida. igeos-eun jeongmal jeulgeobseubnida. naneun dangsin-ui jong-iui majimag choan-eseo dangsin-ui hwalyeohan choejong gyeolgwaleul bogil godaehabnida.

English Spanish French English - detected Korean English Spanish Translate

I could not save the money

나는 돈을 구할 수 없었다.

naneun don-eul guhal su eobs-eossda.

Google Translate

https://translate.google.com/#ko/en/%ED%87%B4%EA%B7%BC%ED%95%98?&hl=ko

Google Translate

Translate

Japanese English Korean Detect language Korean English Spanish Translate

퇴근하세요 Please work

toegeunhaseyo

Turn off instant translation



Image from
<http://www.blockchaintechnologies.com/blockchain-smart-contracts>

- Companion?
- Assistant?
- Substituter?
- Competitor?

Python 기초: 변수와 입출력 함수

- 지능정보공학설계, Week 2 -

이 지 항 (Ph.D.)

Assistant Professor
Department of Human-Centered AI
Sangmyung University, Seoul, KR
(jeehang@smu.ac.kr)

데이터형과 변수
변수의 연산
입·출력함수

- 자료형, 데이터형; not 테스트형
- 컴퓨터에서 다루는 다양한 데이터 값들의 유형을 의미
- 예) 정수형 (integer), 실수형 (float), 문자열형 (string)

■ 파이썬의 데이터형

- 정수형 (int) – 1, 3, 5, 2, 10, ...
- 실수형 (float) – 1.2, 3.9, 10.456, ...
- 문자열형 (str) – “hello”, “how are you?”, ...
- 부울 (bool) – true, false

■ 정수를 이용하여 사칙연산을 실행

- 정수 3과 정수 5를 덧셈 연산자를 사용하여 덧셈을 수행
- 프롬프트 '>>>' 이후에 덧셈식 $3 + 5$ 를 입력하고 엔터키를 입력하면 명령문 아래에 8이라는 값이 출력됨
- 파이썬은 입력 후 출력 결과를 대화식으로 즉시 확인할 수 있음

```
>>> 3 + 5
```

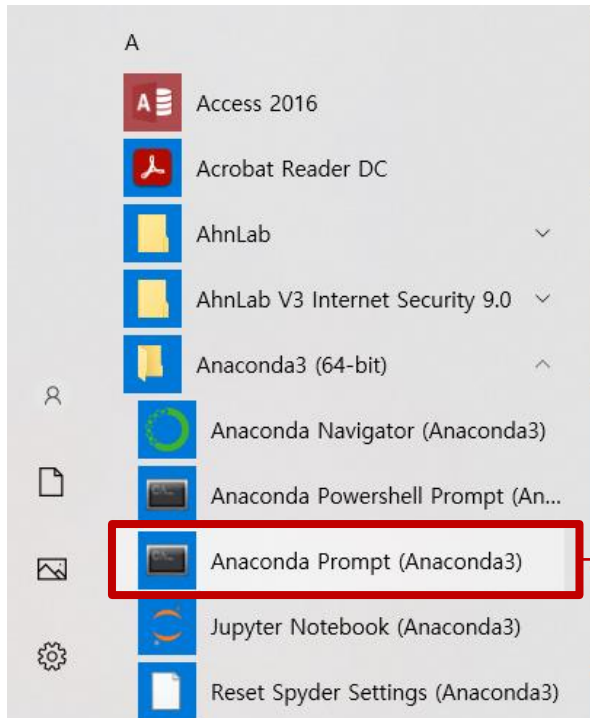
```
8
```

console에서 할 수 있습니다.

How to find “Python console”?

© J.Lee. 2022

- Anaconda를 사용한다고 가정하면



Anaconda Prompt (Anaconda3) - python

```
(base) C:\Users\admin>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

- 문자열형(str)
 - 단어나 문장을 저장하기 위한 데이터형

명령문

```
>>> 'hello'  
'hello'
```

데이터 형과 변수

- 변수 (Variable = Vary + able ≈ 변할 수 있어!)
 - 데이터를 저장할 수 있는 메모리 공간의 이름
 - 프로그래머가 쉽게 데이터에 액세스할 수 있도록 특정 메모리에 이름을 붙인 것
 - 지속적으로 내용을 다른 값으로 갱신할 수 있음

용어	정의
변수(variable)	데이터를 저장할 수 있는 메모리 공간의 이름
데이터형(Data Type)	컴퓨터에서 다루어지는 다양한 데이터 값들의 유형

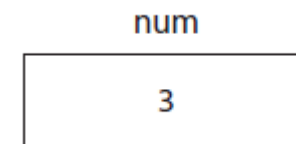
■ 등호(=)

- 산술적인 계산이 아님
- 오른쪽에 위치한 값을 왼쪽에 저장하라는 연산자
- 예) name = 'JeeHang'

연산자	기능
=	오른쪽 값을 왼쪽에 대입(저장)

■ 정수형 변수

- ① num이라는 정수형 변수가 메모리에 생성
- ② 정수형 변수 num에 3을 저장



```
>>> num = 3          # 정수형 변수 num을 생성하고 정수 3을 저장
>>> num
3
```

■ 실수형 변수

```
>>> num = 3.5        # 실수형 변수 num을 생성하고 실수 3.5를 저장
```


■ 파이썬의 문자열형 변수

- 작은따옴표(' ')를 이용
- 문자형과 문자열형을 같이 사용
- 큰따옴표(" ")를 사용해도 됨

```
>>> user = 'John'    // 문자열 John을 변수 user에 저장하라는 의미
>>> user
'John'
```

```
>>> name = '홍길동'  // 어떤 의미?
>>> name
'홍길동'
```

■ 부울형 변수

- 부울 또는 불린
- boolean의 약어인 bool로 데이터형을 표시
- True 또는 False

```
>>> exist = True  
>>> exist  
True
```

변수와 사칙연산

■ 산술연산자

- 정수와 실수의 합이 특별한 형변환(Type Conversion) 없이 수행됨

```
>>> a = 3                # 정수형
>>> b = 3.5              # 실수형
>>> a + b
6.5
```

!! 이보게 교수 양반, 갑자기 형변환이라니 무슨 말인가?



** 컴퓨터에서는 데이터형이 다르면, 원래 계산이 안됨 (예를 들어 C언어...). 연산에 참여하는 변수들의 데이터형이 같게 만들어서 계산을 해야 함.

** 그런데 여기서는 데이터형이 달라도 계산이 됨.
파이썬은 그것을 지원함

■ 관계연산자

- 2개의 값이나 변수 간의 관계를 연산
- 결과는 참(True) 또는 거짓(False)

연산자	의미
>	크다.
<	작다.
>=	크거나 같다.
<=	작거나 같다.
==	같다.
!=	같지 않다.

■ 관계식


- “a는 0보다 크다.”
- 참이면 True, otherwise False 출력
- 연산자의 왼쪽을 주어로 해석

관계식	a > 0
의미	a는 0보다 크다.

■ 실습

명령문

```
>>> a = 3
>>> b = 5
>>> a > b                # a는 b보다 크다.
False
>>> a < b                # a는 b보다 작다.
True
>>> a >= b               # a는 b보다 크거나 같다.
False
>>> a <= b               # a는 b보다 작거나 같다.
True
>>> a == b               # a는 b와 같다.
False
>>> a != b               # a는 b와 같지 않다.
True
```

 Anaconda Prompt (Anaconda3) - python

```
(base) C:\Users\admin>python
Python 3.8.3 (default, Jul 2 2020, 17:30:36) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

잊지말자 콘솔 (Anaconda prompt)

■ 논리연산자

- 2개의 값이나 변수 간에 논리적인 연산을 수행
- 결과는 참(True) 또는 거짓(False)

연산자	의미
and	그리고
or	또는
not	부정(아니다)

논리식	$(a > 0) \text{ and } (a > b)$
의미	a는 0보다 크고 a는 b보다 크다.

■ and

- 두 개의 피연산자(operand) 중에 하나라도 False이면. 결과도 False

■ or

- 두 개의 피연산자 중에 하나라도 True이면 결과도 True

■ not

- 피연산자 값의 반대값으로 전환

a	b	a and b	a or b	not a
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

입출력함수

■ print() 함수

- 해당 객체를 출력장치(스크린) 또는 문자 스트림 파일로 전송

명령문

```
>>> print('안녕하세요!')  
안녕하세요!
```

명령문

```
>>> print(3)          # 정수형  
3  
>>> print(3.5)        # 실수형  
3.5  
>>> print(True)      # 부울형  
True
```

■ print() 함수

- 함수의 입력값으로 수식을 지정할 수 있음
- 해당 수식을 계산한 후 출력

명령문

```
>>> print(3+5)  
8
```

- 변수를 이용한 수식을 계산한 동시에 결과를 출력할 수 있음

명령문

```
a = 3
b = 3.5
c = True
print(a)          # 정수형(int)
print(b)          # 실수형(float)
print(c)          # 부울형(bool)
```

명령문

```
>>> a = 3
>>> b = 3.5
>>> print(a+b)
6.5
```

- 다수의 값이나 변수에 저장된 값들을 출력하려면 각각의 내용들을 콤마(,)로 연결

명령문

```
>>> age = 21
>>> print('나이: ', age)
나이: 21
```

명령문

```
>>> name = '홍길동'
>>> print('나는', name, '입니다.')
나는 홍길동 입니다.
```

- 변수에 저장된 값과 문자열들을 같이 출력할 수 있음

명령문

```
>>> a = 3
>>> b = 3.5
>>> print(a, '*', b, '=', a*b)
3 * 3.5 = 10.5
```

■ 문자열 데이터값 입력

- input() 함수를 사용
- 사용자는 문자열 입력 후 Enter를 눌러야 함
- 사용자를 위한 지시사항을 출력해줄 필요가 있음

명령문	<pre>>>> name = input('이름: ') 이름: 홍길동 # 이름: 이라고 표시되면 그 옆에 자신의 이름을 입력 >>> print(name) 홍길동 # name에 자신이 입력한 이름을 확인 가능</pre>
-----	--

■ 정수형 데이터값 입력

- input() 함수로 입력받으면 숫자든 문자열이든 모두 문자열로 저장됨
- 문자열을 정수 형태로 변환해서 계산하는 int() 함수를 이용

명령문

```
>>> age = input('나이 : ')
나이 : 21
>>> after = input('몇 년 후? ')
몇 년 후? 10
>>> print(int(age) + int(after))
31
```

이런게 형변환 (문자열형에서 정수형으로 변환함)

■ 실수형 데이터값 입력

- input() 함수로 입력받으면 문자열로 저장됨
- 문자열을 실수형으로 변환하기 위해서는 float() 함수를 사용

명령문

```
>>> height = input('키 : ')
키 : 170.2
>>> print(float(height) + 1.5, 'cm')
171.7 cm
```


데이터형과 변수

변수의 연산 – 산술연산, 논리연산

입·출력함수 – print(), input()

See you in the next lecture!