CS2109S: Introduction to AI and Machine Learning

Lecture 1:
Intro to CS2109S and AI

18 August 2023

Intro to CS2109S

Instructors

Muhammad **Rizki** Maulana
rizki@nus.edu.sg

Ben Leong Wing Lup
benleong@nus.edu.sg

Chonghui Hu
conghui@nus.edu.sg

Teaching Assistants


Rafeed


Nyamdavaa


Minh Tuan


Yu-Chuan


Edison


Eric Han


Linus Lee


Jiefan


Yu Pei


Sheryl


Zheng Long


Florentiana

Teaching Assistants

John


Amadeus


Viet Anh


Zhenhao


Fu Di

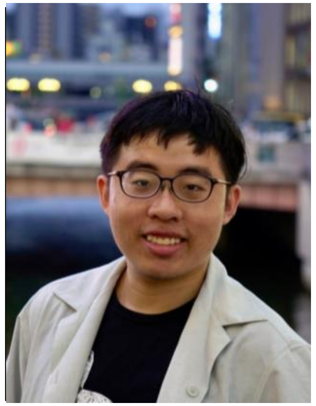

Chng Ian


Guo Yang


Jin Waye


Chu Thanh


Hung-Yu


Jie Feng

Rights Infringements on NUS Course Materials

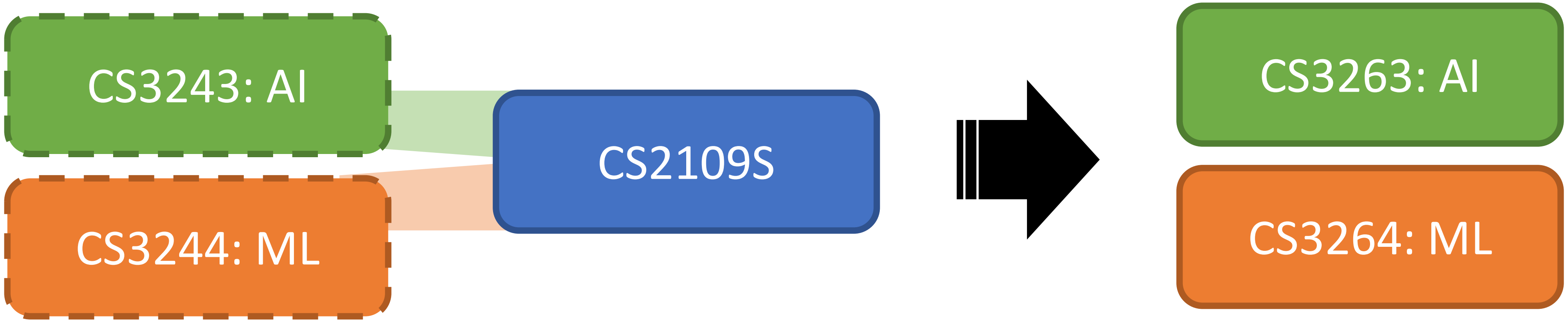
All course participants (including permitted guest students) who have access to the course materials on Canvas or any approved platforms by NUS for delivery of NUS modules are not allowed to re-distribute the contents in any forms to third parties without the explicit consent from the module instructors or authorized NUS officials

Motivations behind CS2109S

- AI/Machine learning is a big deal

ChatGPT AlphaGo

- Exposure to AI/ML for CS majors is deemed important by the school



Pre-survey Results

Some worries about:

- None (most popular)
- Math
  - Pre-requisites (more on this) + practice problems
- Heavy Workload
  - Follows CS2040
  - Added more scaffolding (last semester)
  - Common mistakes, e.g., using Jupyter notebook for debugging algorithms
- Difficulty
  - Not meant to be hard. Still adjusting the difficulty…
- Final Exam Format
  - We are trying to improve this

Course Pre-requisites

- CS1101S, CS1010S or equivalent
- CS1231 or equivalent
  - Trees, graphs, counting & combinatorics
- MA1521
  - Differentiation, chain rule
- CS2040S or equivalent
  - Tree and graph search
- Linear algebra
  - Vector, matrix, and their operations
- Python

Problem Set 0

Overview

Midterm
(During lecture time)
Subject to successfully securing MPSH

Final Assessment
(2 days contest**)**

Week 1 | Week 5 | Week 7 | Week 13

"Classical" AI | Machine Learning

Search Algorithms

- Uninformed search: BFS, DFS
- Informed search: A*
- Adversarial search: Minimax

Applied CS2040S, CS1231
Python

ML Models & Techniques

- Decision Trees
- Linear/Logistic Regression
- Support Vector Machines
- Neural Networks

Miscellaneous

- Unsupervised Learning
- AI & Ethics

Applied Linear Algebra, Calculus, Statistics & Probabilities
Numpy, Scikit-learn, PyTorch

"hands-on" problem sets

Textbook

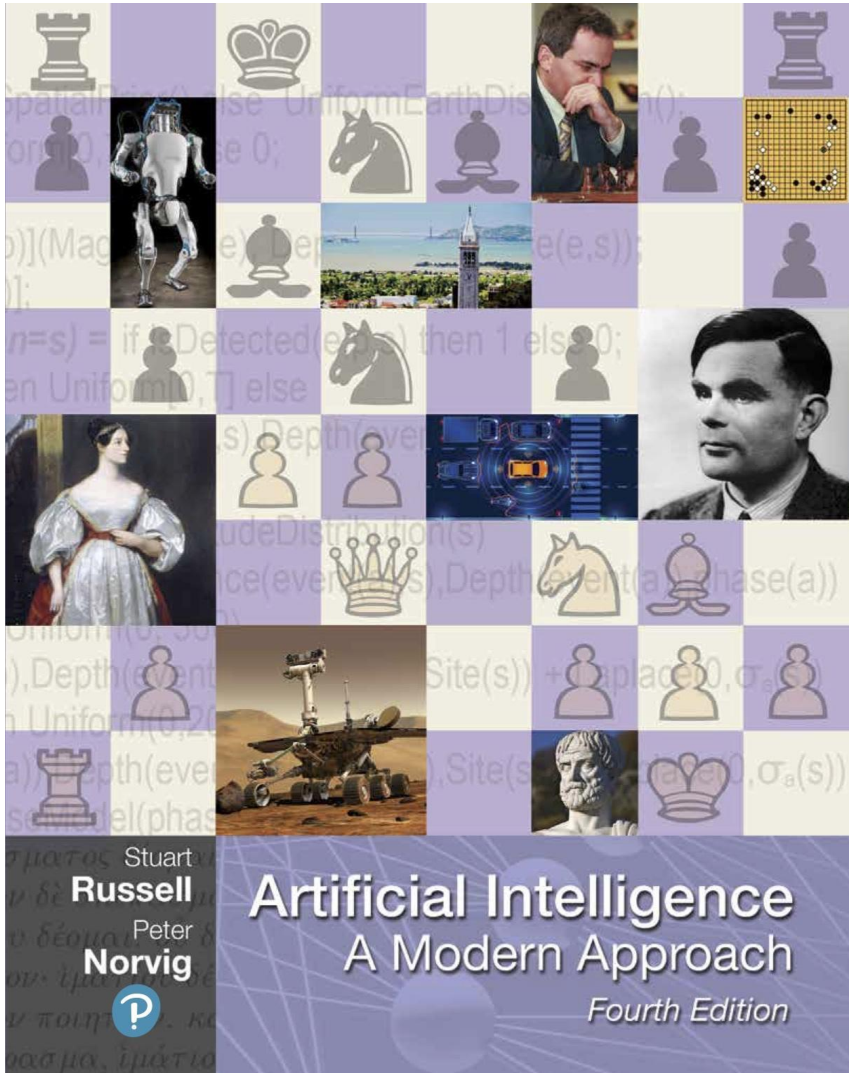
Russell and Norvig (2021)

Artificial Intelligence:

A Modern Approach (4th Ed)

a.k.a. **AIMA**

There is no need to buy this book.



Learning Management System

coursemology

Tutorial Allocation is done through **Tutorial Survey, <u>not</u> Edurec**

Due Saturday, 19 Aug 23:59!

canvas

Only for **Gradebook** to record marks
for checking and **webcast**

Plagiarism Policy

Dos

- Discussions without sharing/consulting/taking away any code
- Use ChatGPT with proof (e.g., give ShareGPT links)

Don’ts

- Use codes from those who has done or currently doing the course
- Use codes from the internet without proper citations
- Publish codes to any publicly accessible sites (e.g., GitHub, Google Drive) or send your codes to anyone

Plagiarism checker will be performed against all previous batches!

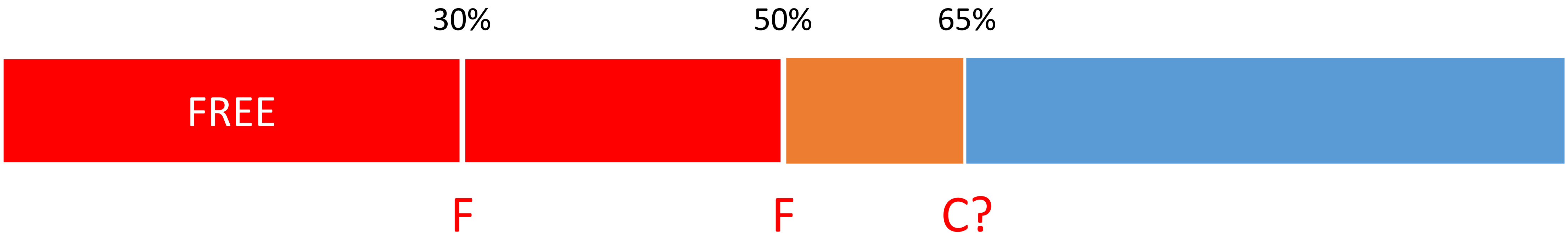
Case Studies of Plagiarisms

- Created a base source code and **derived** solutions from the same base
  - Caught and cases **submitted** to BOD
- Fully/partially copy-pasted friend's solutions (current/past student) and **tried to be smart** by doing some **nontrivial modifications**
  - Caught and cases **submitted** to BOD
- Fully/partially copy-pasted friend's solutions (current/past student) and **claimed** that solutions were from **ChatGPT** or **ported** from publicly available source codes
  - Caught and cases **submitted** to BOD

There were many more cases, but basically, we dealt with them **appropriately**

Grading

Name	Percentage
Coursemology (CA)	30% (expected: full marks)
Midterm	35%
Final Assessment	35%



Gamified CA (30%)

Background Survey

+100 EXP

+50 EXP early bonus!

+100 EXP

+500 EXP

Lecture → Training → Tutorial → Problem Set

(every ~2 weeks)

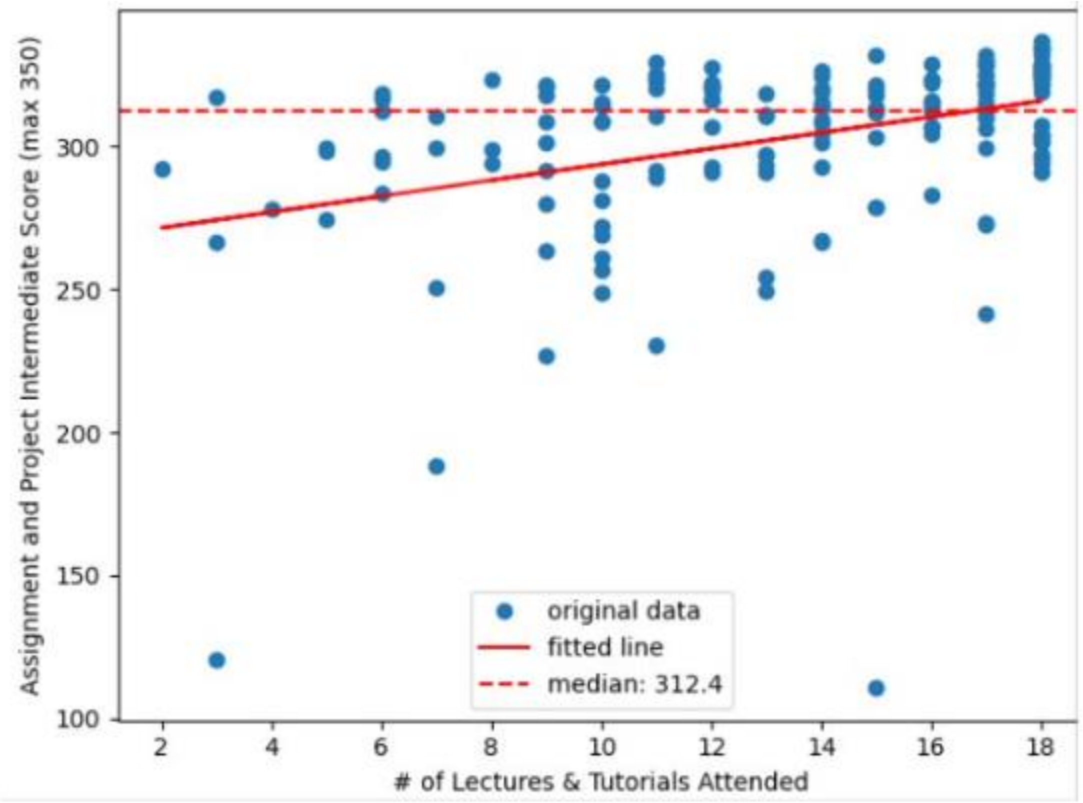
~1300 EXP

Lecture ↘ Forum EXP

+100 EXP

- Enough EXP? Level Up!
- **Your <u>final level</u> is your CA grade**

Attendance vs Performance



Assessments

- **Midterm (35%)**
  - Closed book, open-sheet (as good as open book)
  - Focus on application, <u>not</u> memorization
- **Final (35%)**
  - ~2 days take-home assessment
  - Focus on practical machine learning

Late Policy

- Up to 24 hours, -20%
- Up to 3 days, -30%
- Beyond 3 days, -50%

If you need extension (for valid reasons), please ask early

Intro to Artificial Intelligence

Outline

- What is AI?
- A Brief History of AI
- Intelligent Agents
- The Structure of Agents

Outline

- **What is AI?**
- A Brief History of AI
- Intelligent Agents
- The Structure of Agents

What is AI?

ChatGPT

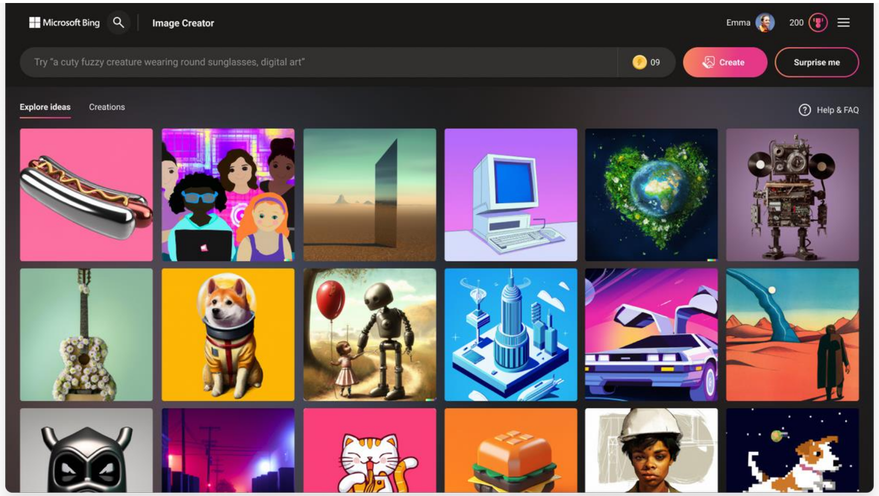
Credit: IEEE Spectrum

Credit: Guardian

Credit: NYTimes

What is AI?

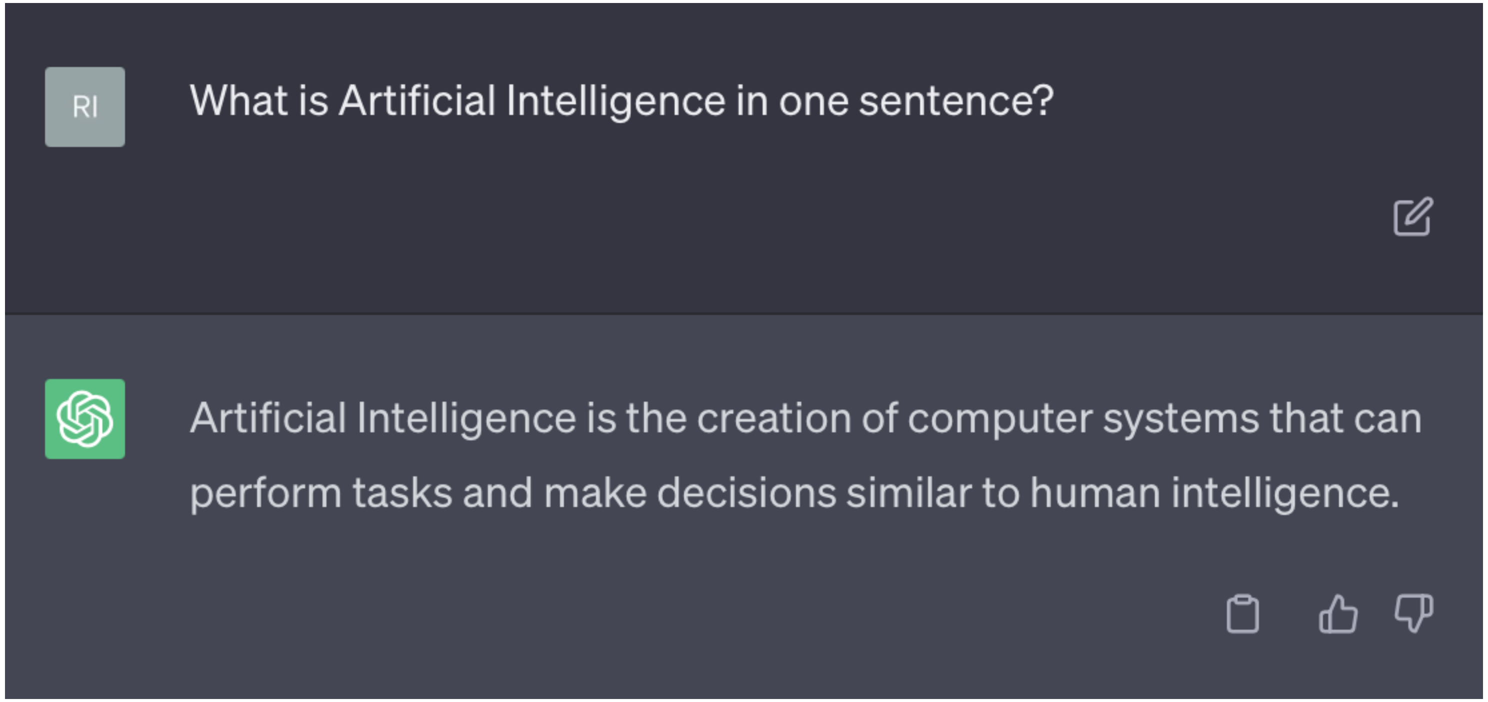
Credit: Tesla



Credit: Microsoft

Credit: Eden AI

What is AI? Let’s ask AI



What is AI?

Write it as a **poem***

This poem succinctly summarizes this course.

Week 1

In realms of circuits and digital might,
Artificial Intelligence takes its flight.
AlphaGo, the master of strategic plays,
DeepBlue conquers chess in astonishing ways.

Watson, the genius with knowledge immense,
ChatGPT converses with eloquence.
Self-driving cars, navigating with grace,
Image generation, artistry embraced.

Week 2-4

Search algorithms, exploring vast terrain,

Week 5-11

Decision trees, guiding paths to gain.
Linear models, simple and clear,
Neural networks, complexity they steer.

Week 12

Unsupervised learning, a quest for patterns,
Unveiling insights, where hidden knowledge hammers.

Week 13

Ethics guide us in this AI realm,
Respecting privacy, and doing no harm.

Artificial Intelligence, a powerful force,
With potential vast, both boon and discourse.
May we harness its strength with a watchful eye,
To shape a better future as days roll by.

*) This is not the actual (full) prompt

Outline

- What is AI?
- **A Brief History of AI**
- Intelligent Agents
- The Structure of Agents

A Brief History of AI

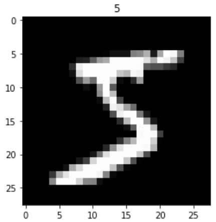
- **Before 1900**
- **Early 1900**
  - Atanasoff-Berry Computer: solve linear equations
  - Neural networks
- **1950s-1960s**
  - Alan Turing: Turing machines, Turing test
  - Checkers AI, Lisp programming language, ELIZA
- **1970s**
  - 1st AI Winter
- **1980s**
  - Expert Systems, Fifth Generation Computers
- **1990s**
  - Deep Blue
- **2000-2010**
  - 2nd AI Winter, Big data
- **2011-Beyond**
  - Deep neural networks, Watson, AlphaGo, ChatGPT

Outline

- What is AI?
- A Brief History of AI
- **Intelligent Agents**
- The Structure of Agents

Intelligent Agents: 1st Attempt

- Handcraft → "Classical" AI
- Learn → Machine Learning



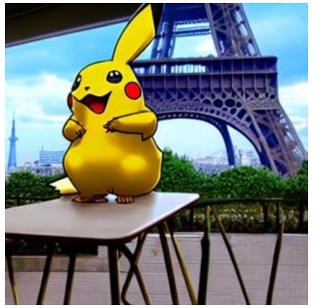
→ Function → 5

"What is AI?" → Function → "AI is …"

Credit: Chess.com

→ Function → Bb6 (chess move)

"Pikachu in the Eiffel tower" → Function →



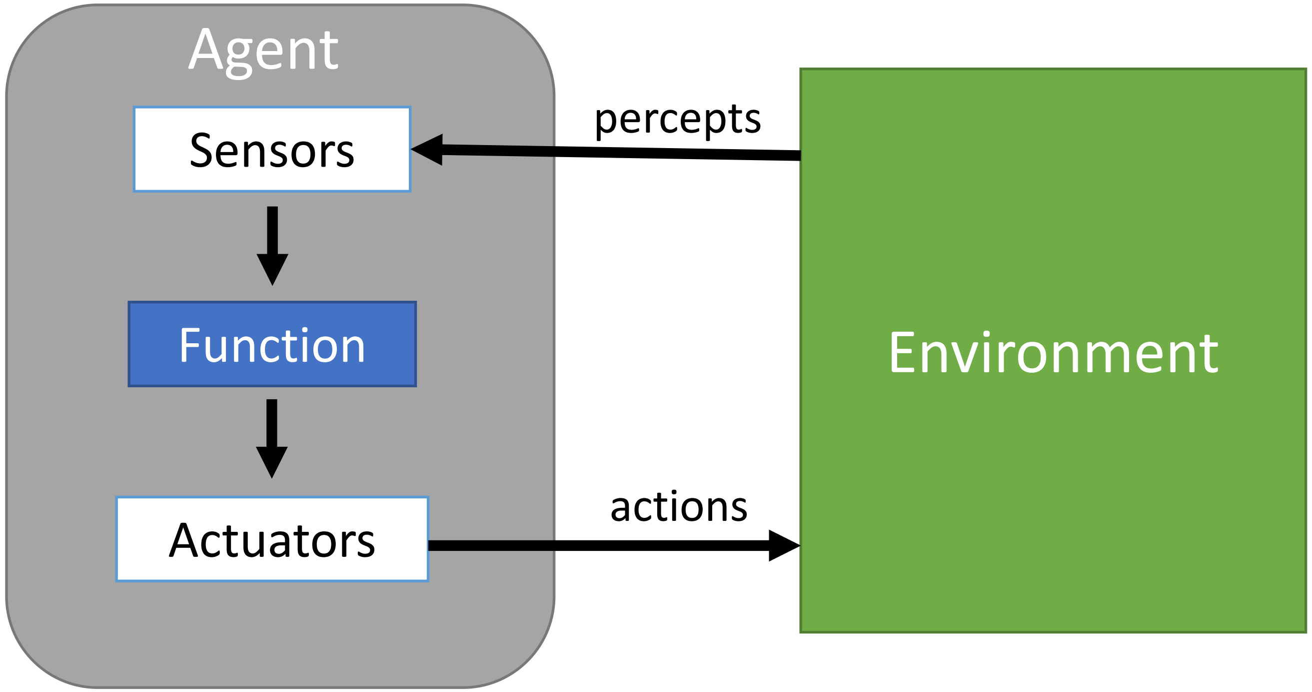
How do we write this Function?

Intelligent Agents: 1st Attempt

Real-world problems are usually not that simple

It usually requires a feedback loop between the AI and the world

Intelligent Agents



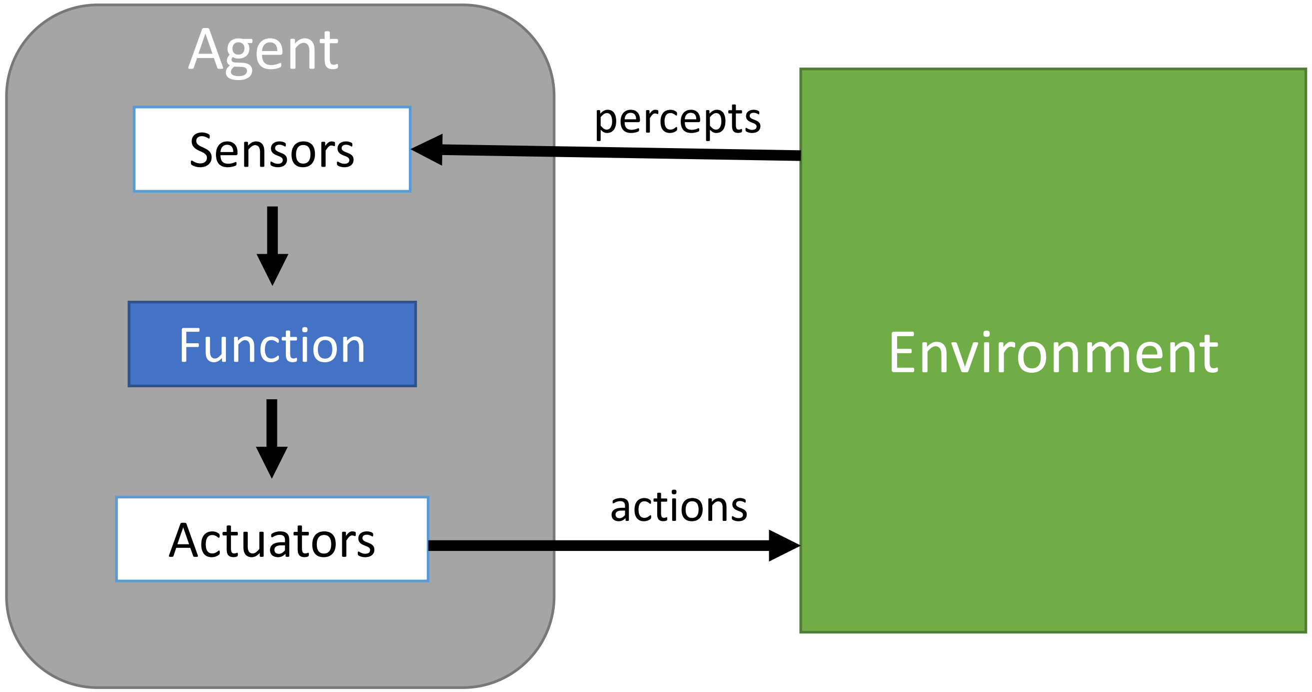
The agent function maps from percept histories to actions:

$$[f\colon \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$

The agent program runs on the physical architecture to produce function f

Intelligent Agents

How would agent know to do the right thing?



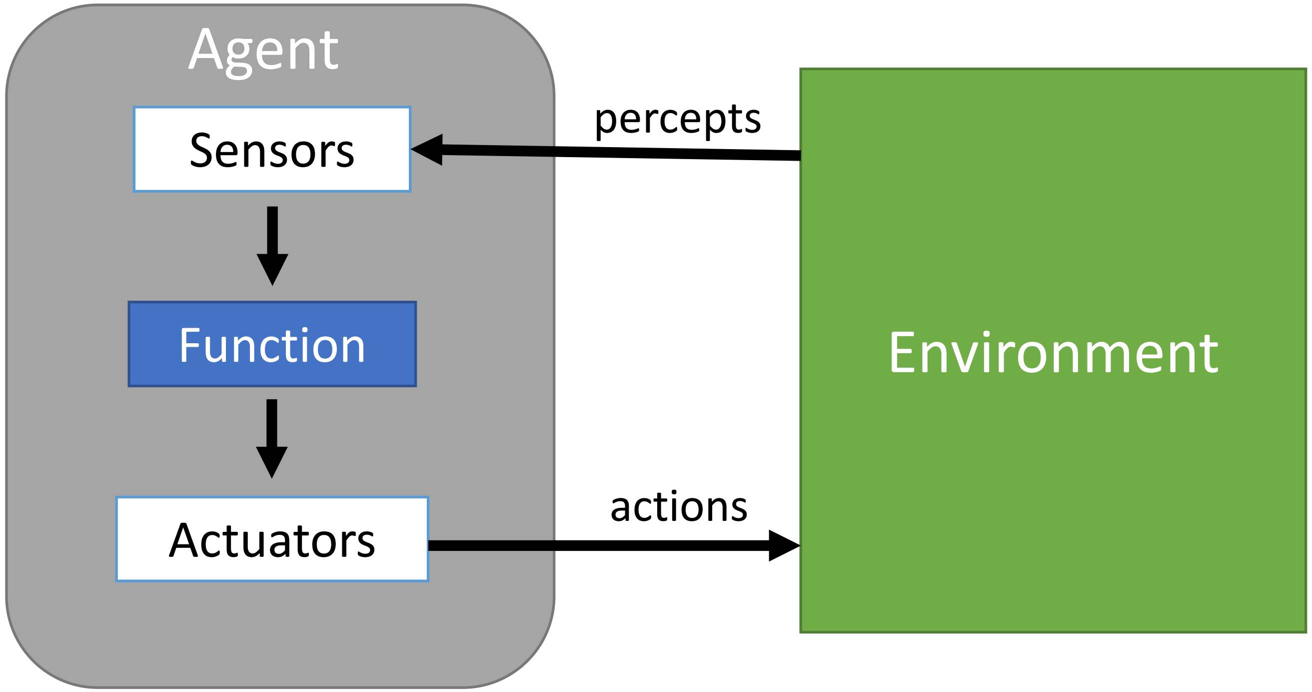
The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^{\star} \rightarrow \mathcal{A}]$$

The agent program runs on the physical architecture to produce function f

Intelligent Agents

How would agent know to do the right thing?



Performance Measure

Things to consider:

- Best for whom?
- What are we optimizing?
- What information is available?
- Any unintended effects?
- What are the costs?

A **rational agent** will choose actions that maximize performance measure.

PEAS

Performance Measure, **E**nvironment, **A**ctuators, **S**ensors

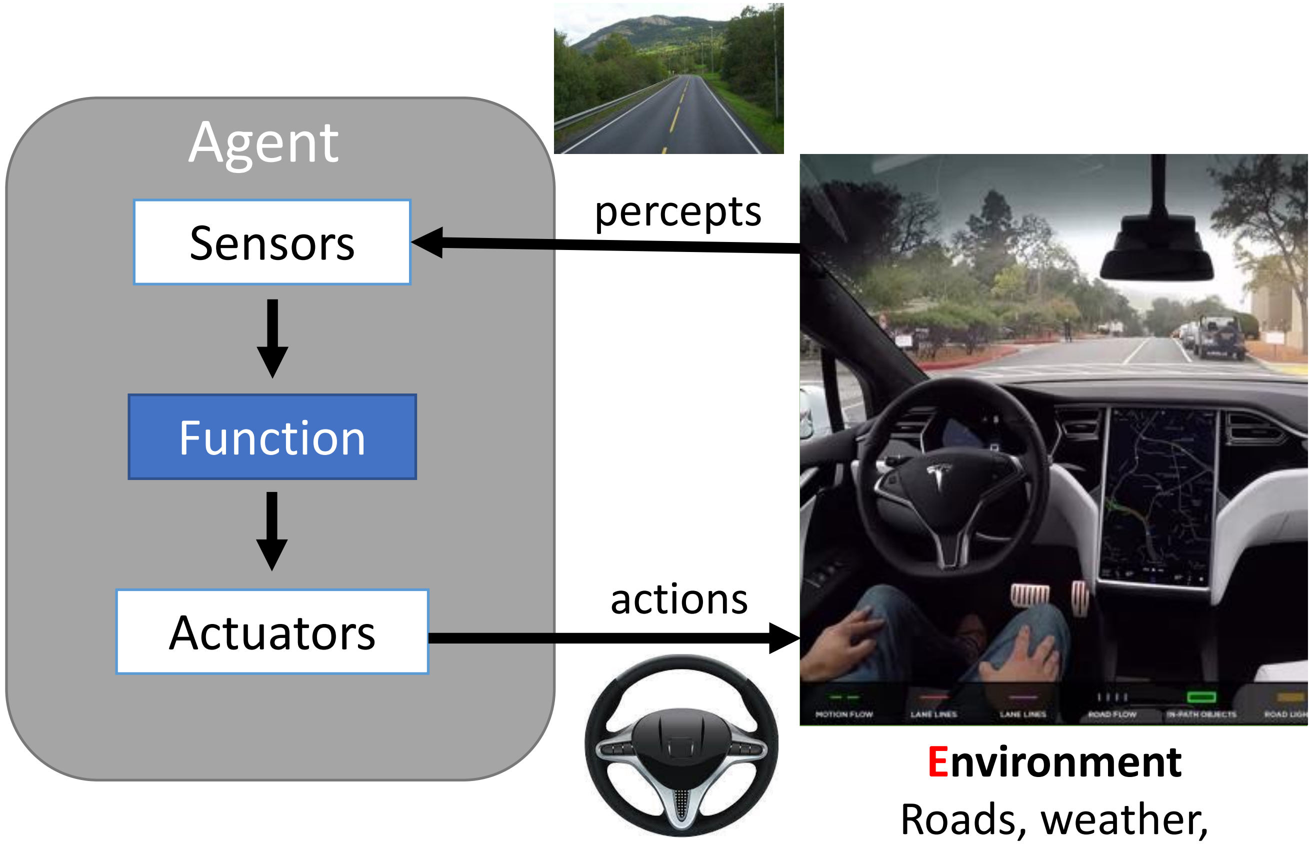
Intelligent Agents: Autonomous Driving

Sensors

- Camera
- LIDAR
- Speedometer
- …

Actuators

- Steering wheel
- Accelerator
- Brake
- …



Environment
Roads, weather,
vehicles, obstacles

Performance Measure

- Safety
- Speed
- Legal
- Comfort
- …

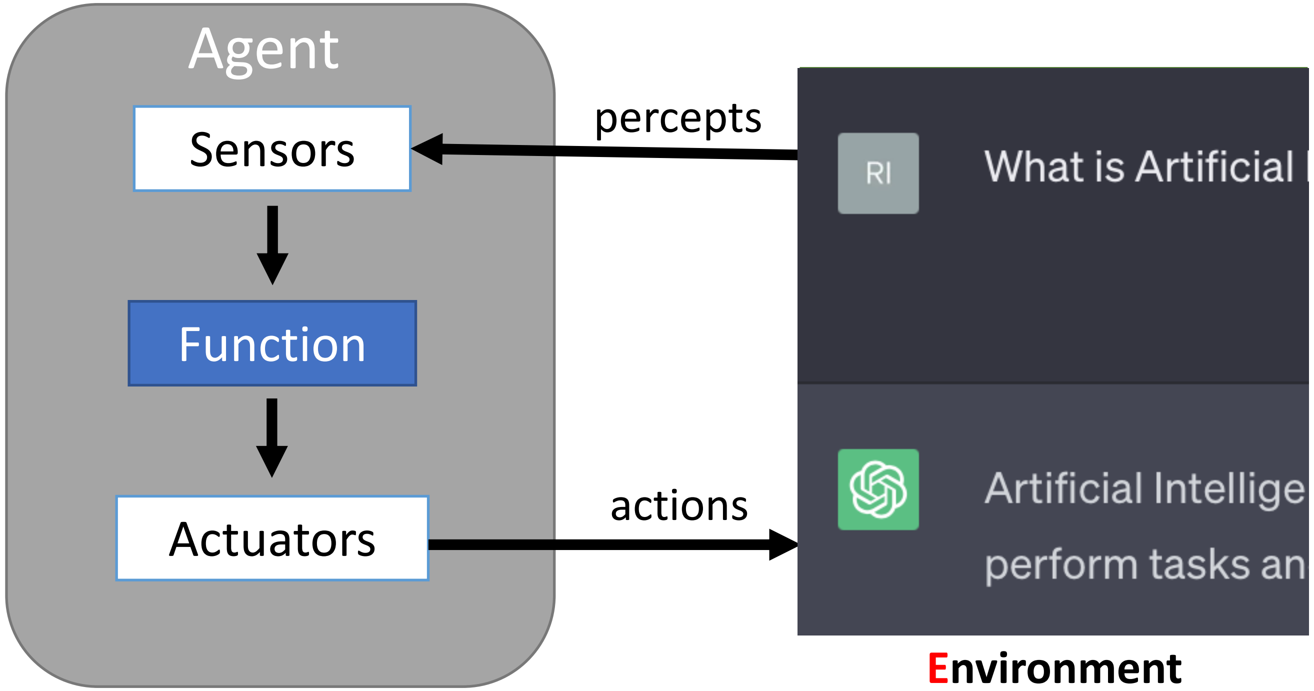
PEAS

Performance Measure, Environment, Actuators, Sensors

Intelligent Agents: AI Assistant/Chatbot

Sensors
- Text input
- Chat history
- Context
- …

Actuators
- Text output
- Image output
- API
- …



Environment
Chat interface, third-party plugins, user

Performance Measure
- Correctness
- Conciseness
- Legal
- Safety
- …

PEAS

Performance Measure, **E**nvironment, **A**ctuators, **S**ensors

Properties of Task Environment

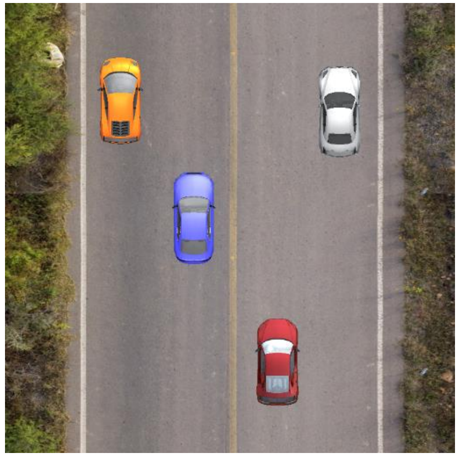
Full

Strategic

Partial

Stochastic/Strategic

Credit: Google Play



Deterministic

Fully observable (vs. partially observable)

An agent's sensors give it access to the complete state of the environment at each point in time.

Deterministic (vs. stochastic)

The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is **strategic**)

Properties of Task Environment

Sequential

Static/Semi-Dynamic

Credit: Keyence

Episodic

Sequential

Dynamic

Episodic (vs. sequential)

The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Static (vs. dynamic)

The environment is unchanged while an agent is deliberating. (The environment is semi-dynamic if the environment itself does not change with the passage of time, but the agent's performance score does)

Properties of Task Environment

Discrete

Multi-agent

Continuous

Multi-agent

Credit: Keyence

Single-agent

Discrete (vs. continuous)

A limited number of distinct, clearly defined percepts and actions.

Single agent (vs. multi-agent)

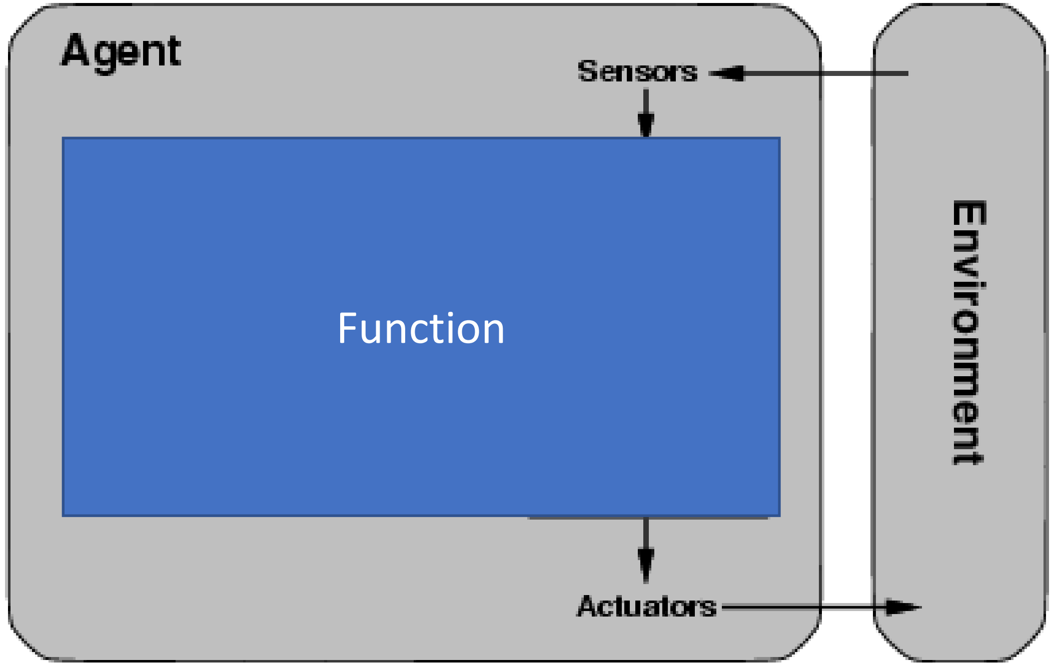
An agent operating by itself in an environment.

Outline

- What is AI?
- A Brief History of AI
- Intelligent Agents
- **The Structure of Agents**

The Structure of Agents

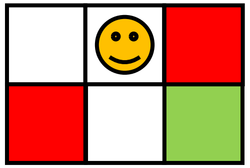
An agent is <u>completely specified</u> by the **agent function** mapping percept sequences to actions.



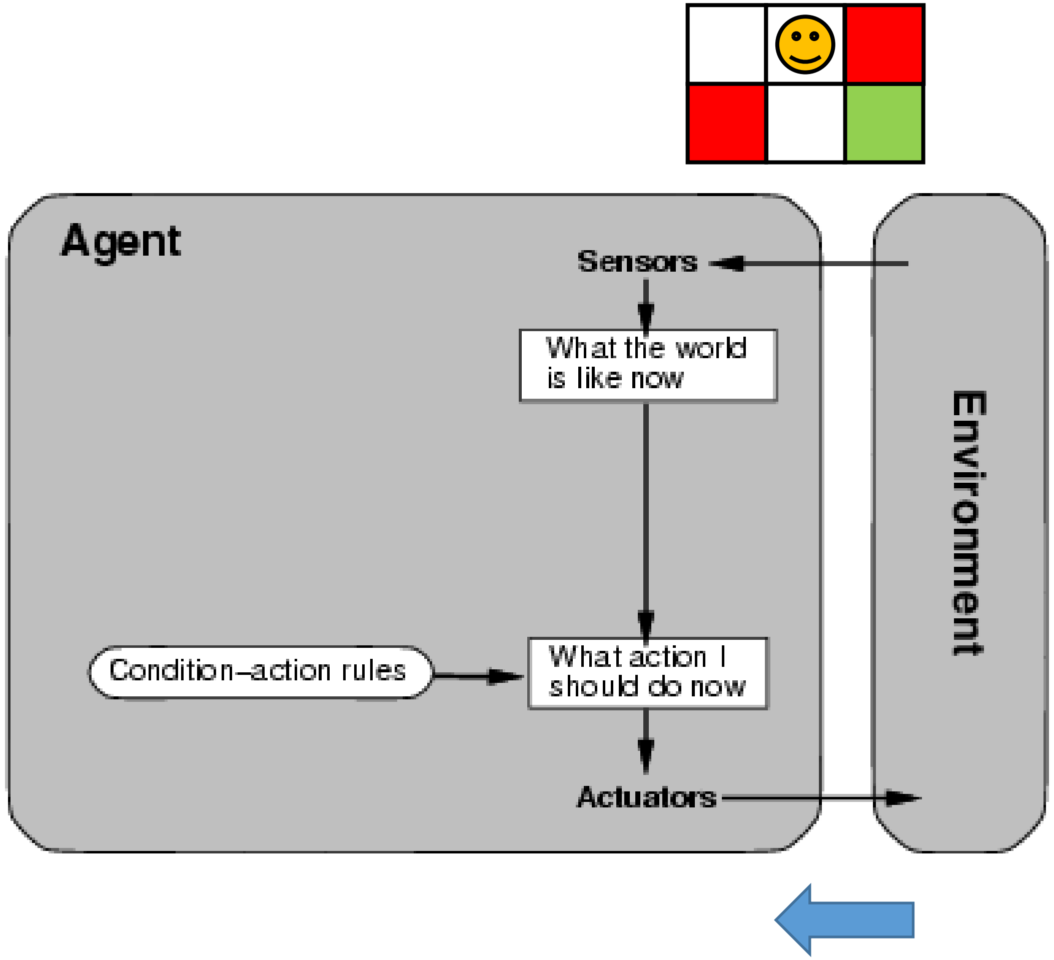
Common agent structures

- Simple reflex agents
- Model-based reflex agents
- Goal-based agents
- Utility-based agents
- Learning agents

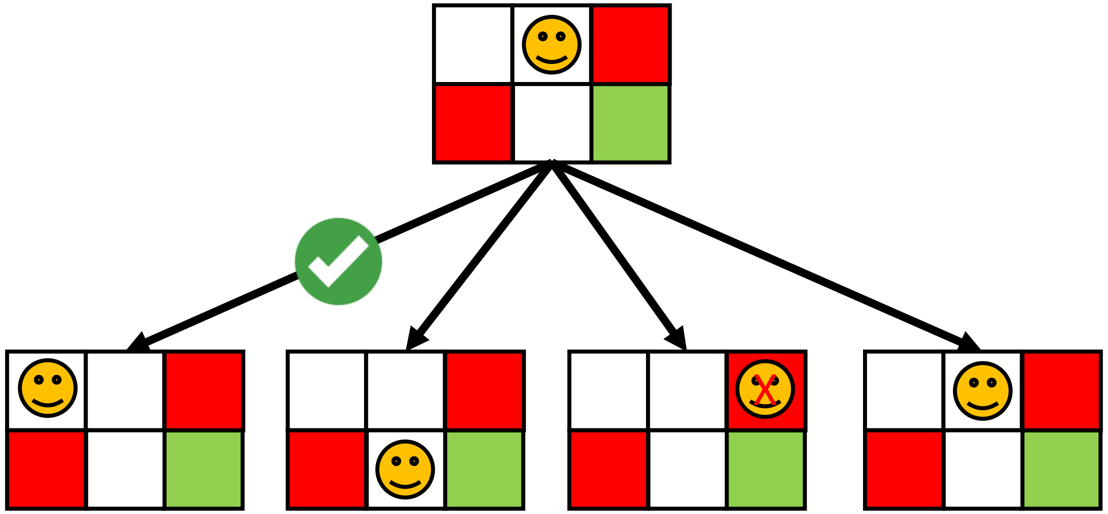
Simple Reflex Agent



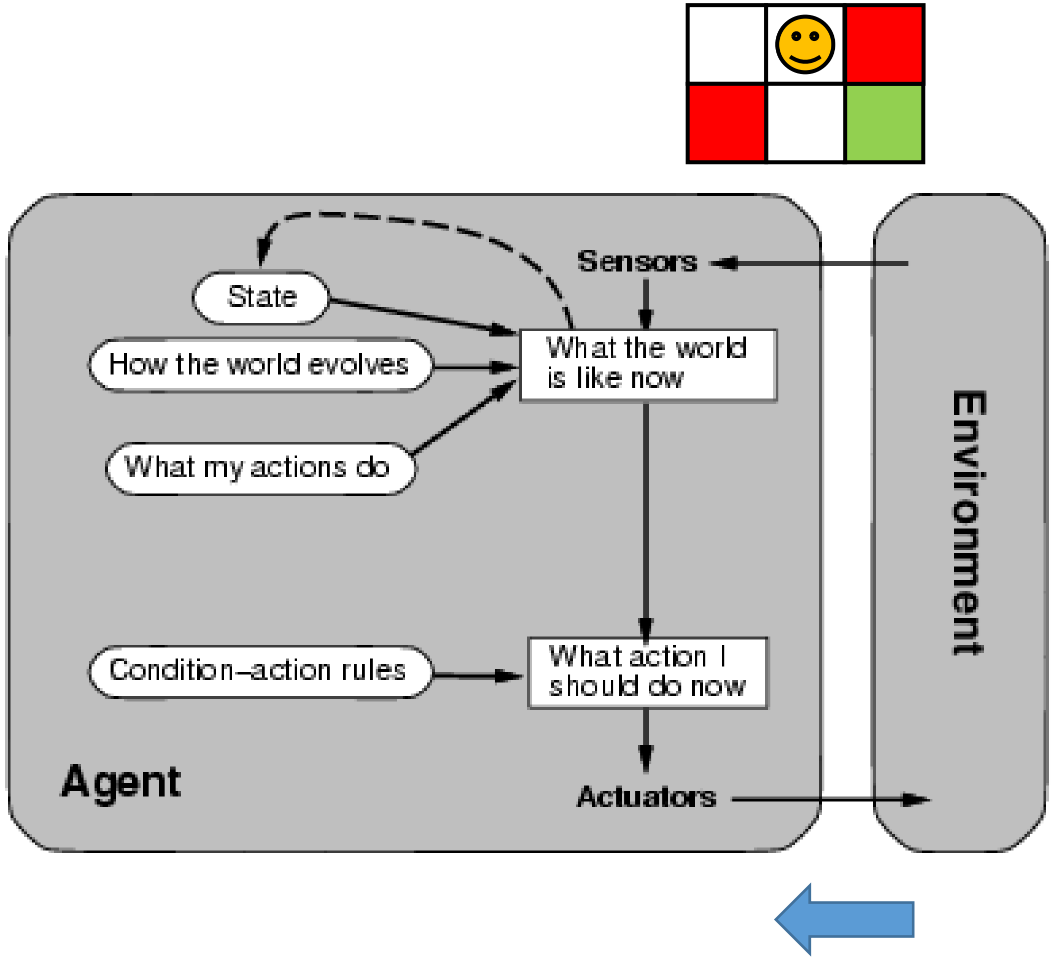
Condition-action rule
If left empty: left
If right empty: right
If up empty: up
If down empty: down



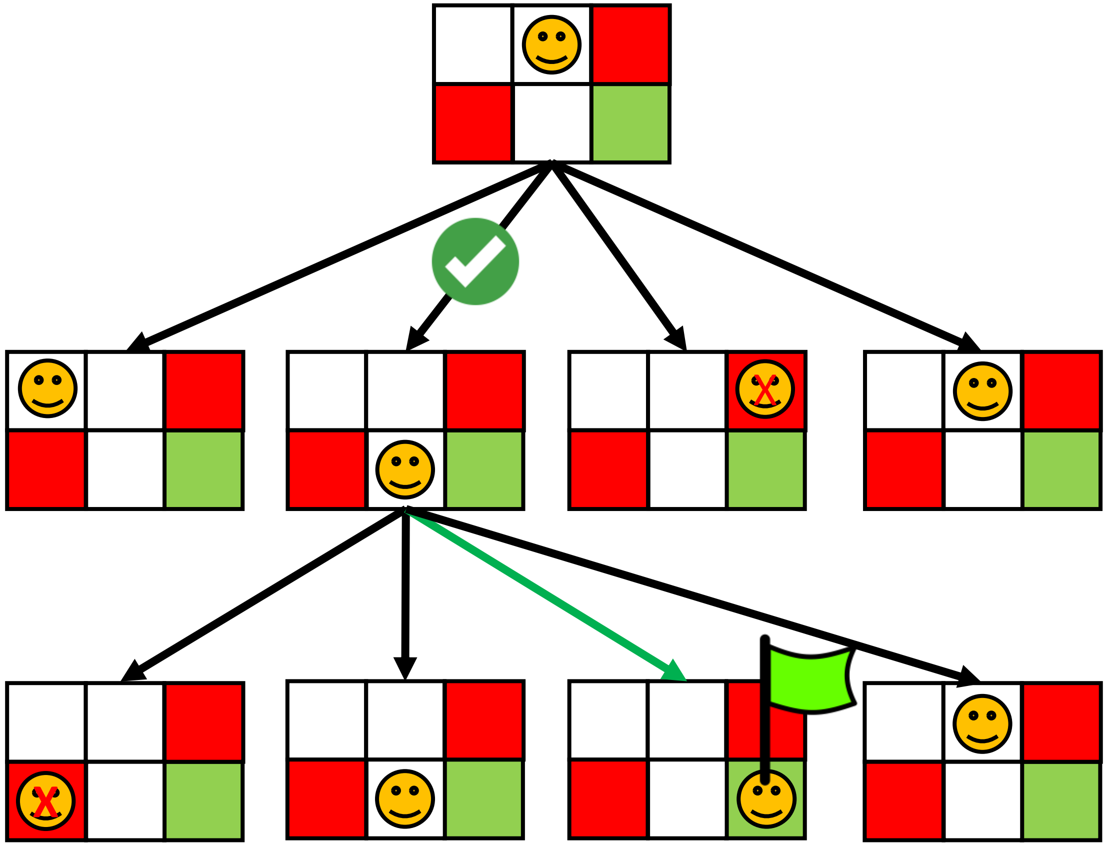
Model-based Agent

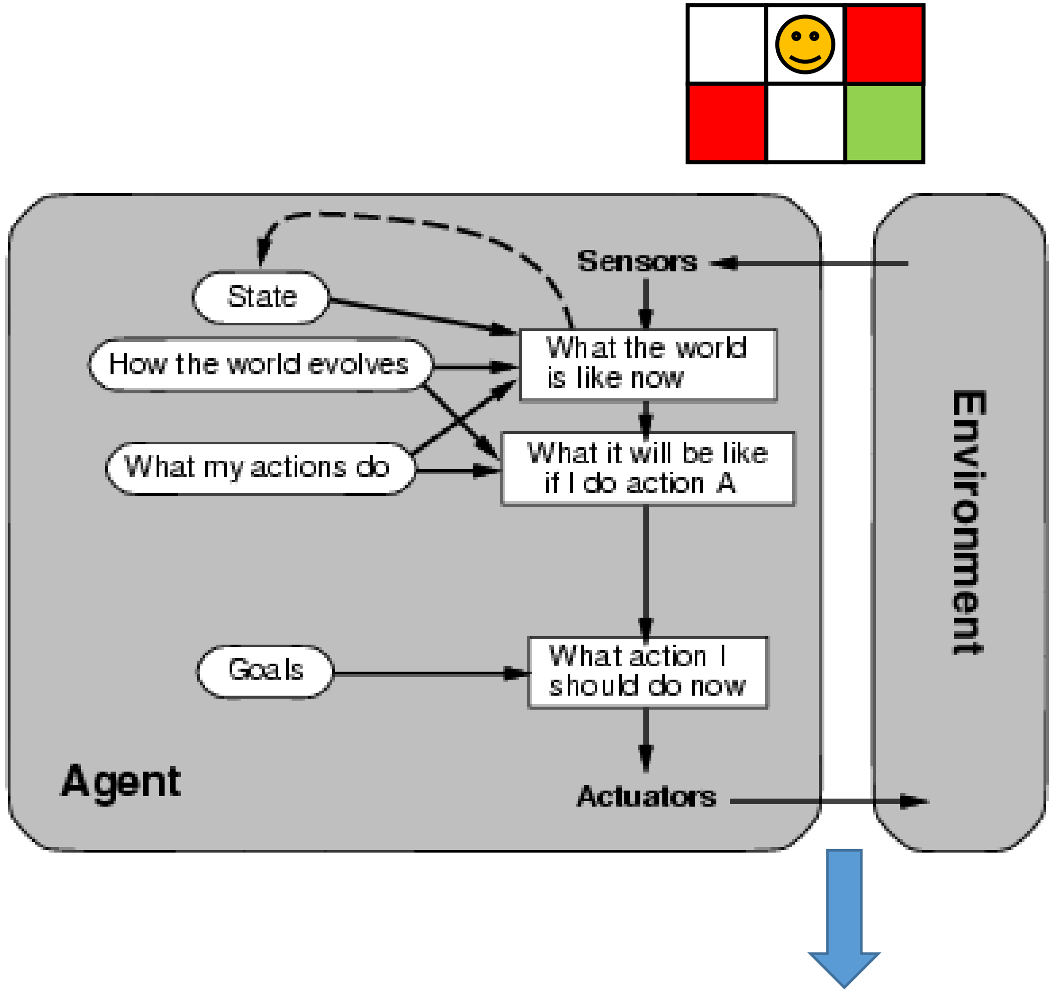


<u>Condition-action rule</u>
If not die, pick the action

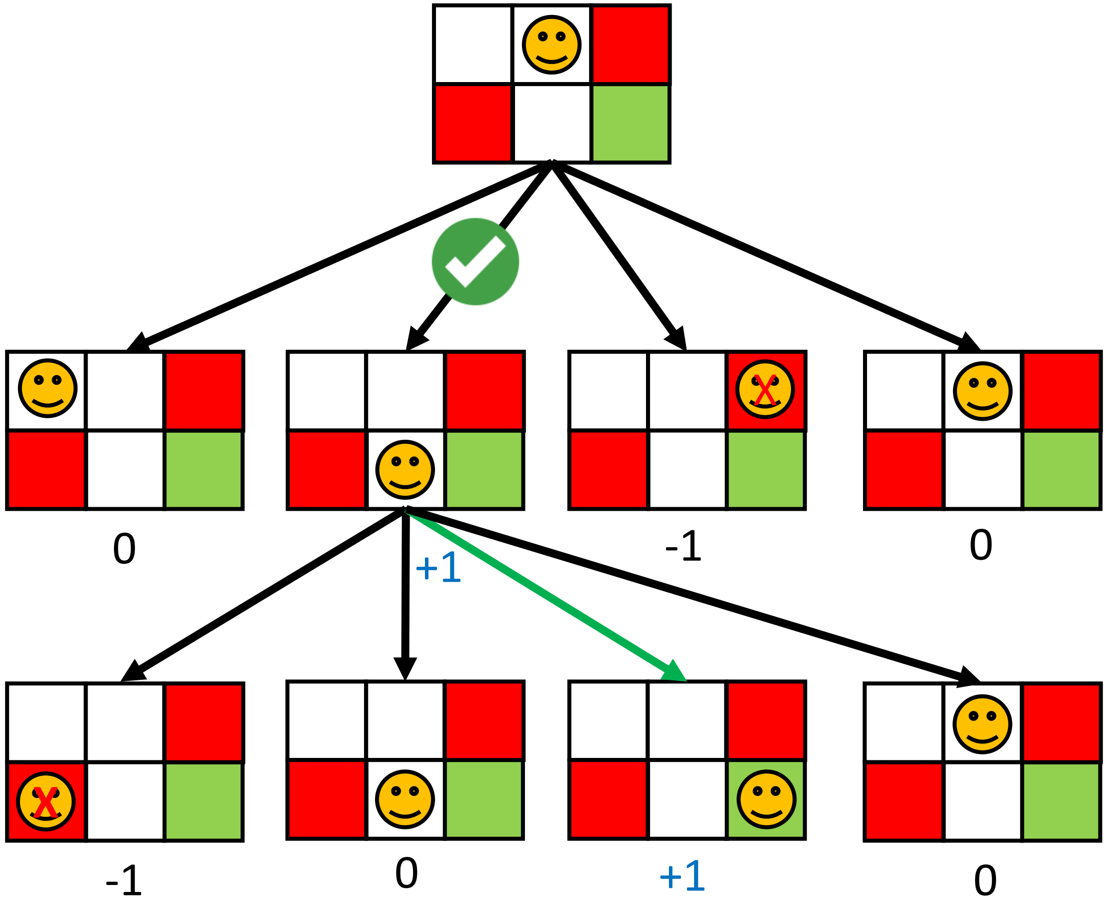


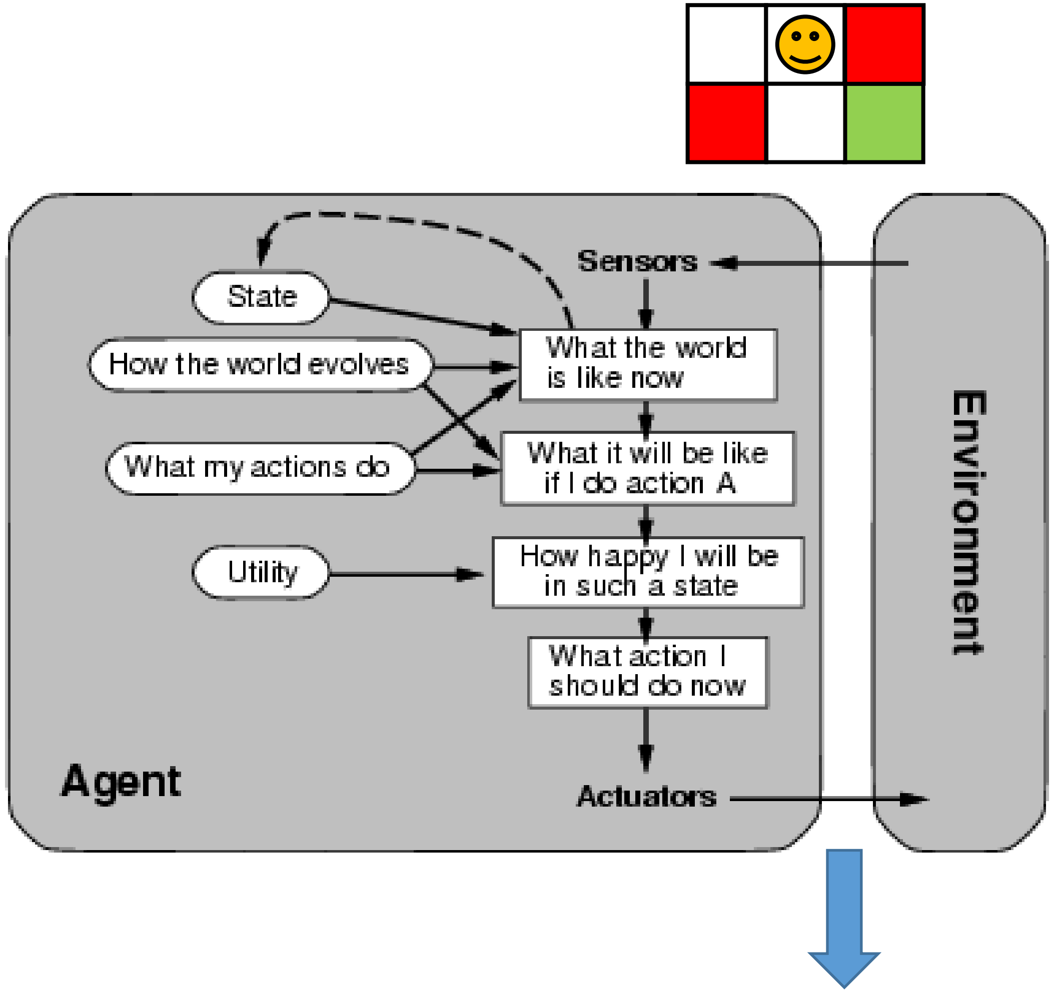
Goal-based Agent



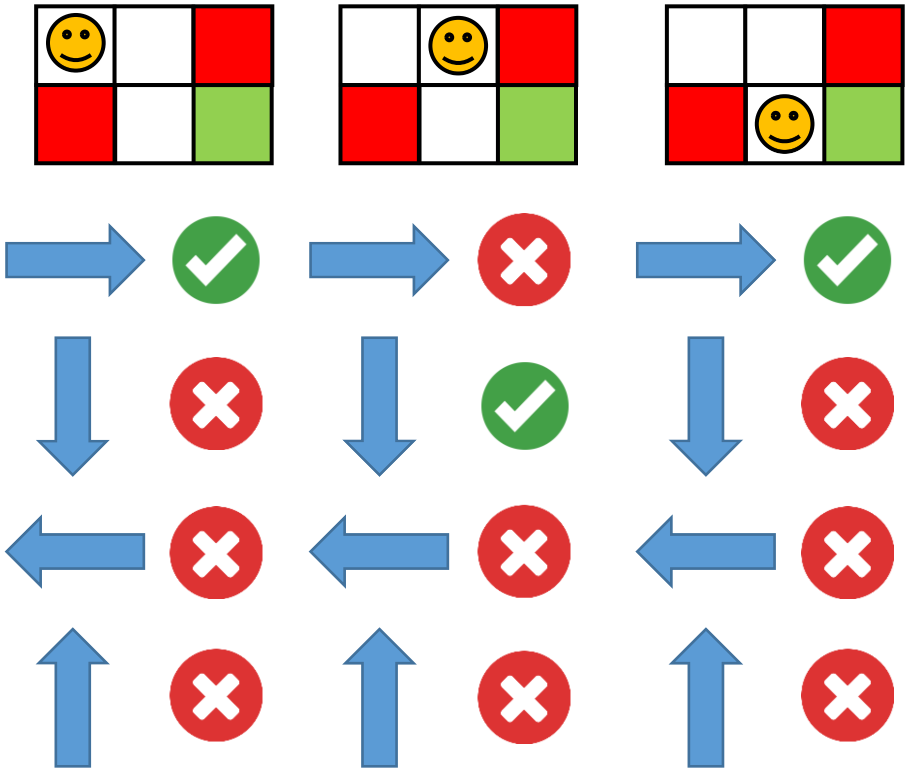


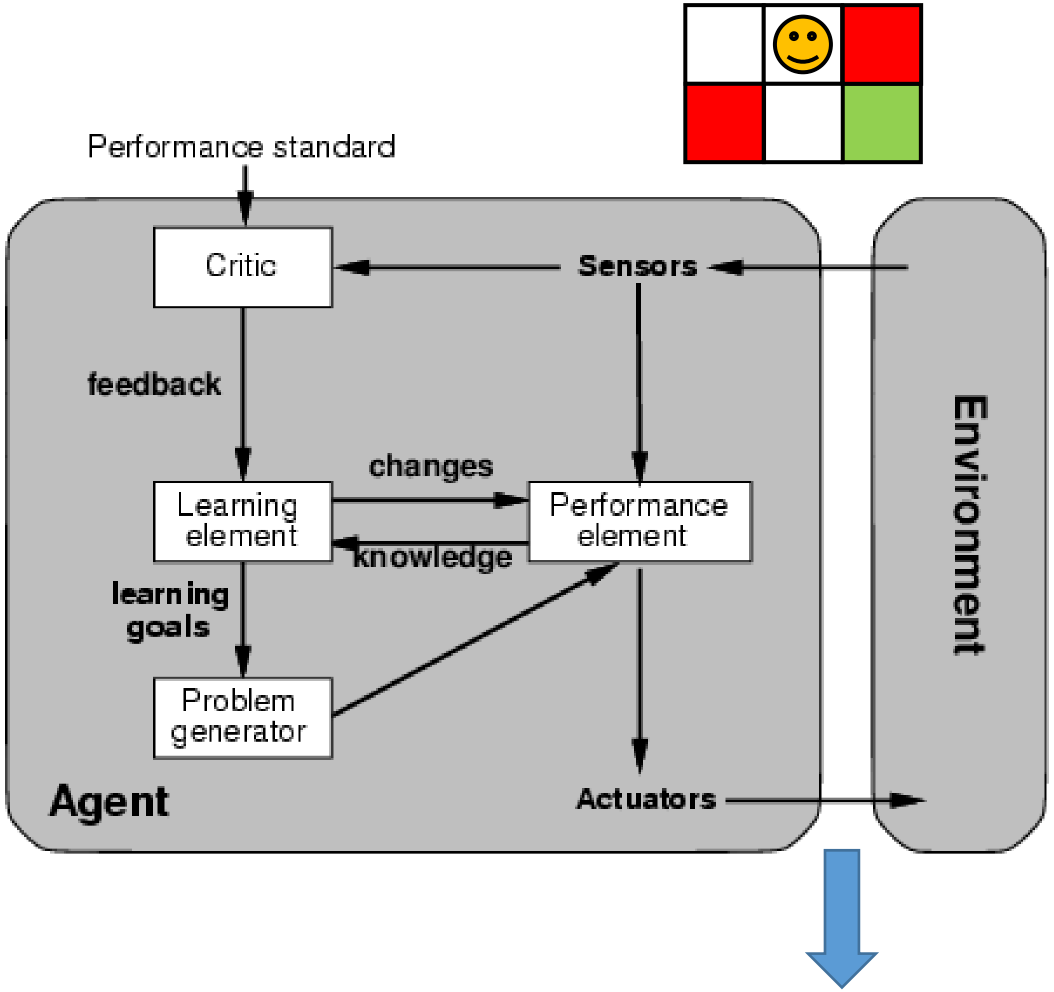
Utility-based Agent





Learning Agent





Can be reflex, model-based, goal-based, and utility-based

Exploitation vs Exploration

An agent operating in the real world must often choose between:

- Maximizing its expected utility according to its current knowledge about the world (Exploit)
- Trying to learn more about the world (Explore)

Credit: Indomie

VS

Credit: Amazon

Summary

- AI: computers trying to **behave like humans**
- **PEAS** Framework:
  - **P**erformance measure: define "goodness" of a solution
  - **E**nvironment: define what the agent can and cannot do
  - **A**ctuators: outputs
  - **S**ensors: inputs
- Agent function is <u>sufficient</u> to define an AI agent
- Common agent structures:
  - Reflex, model-based, goal-based, utility-based, learning
- Exploration vs exploitation

Coming Up Next Week

- **Formulating search problems**
- **Uninformed search algorithms**
  - Breadth-first search
  - Depth-first search
  - Uniform-cost search
- **How to handle repeated states?**
  - Memoization / graph search
- **How to handle infinite depth in search?**
  - Depth-limited search
  - Iterative deepening search
- **Forward, Backward, Bidirectional search**

To Do

- **Lecture Training 1**
  - +100 Free EXP
  - +50 Early bird bonus
- **Problem Set 0**
  - Due 26 August
- **Practice Problems: Python and Numpy (optional)**