



알기쉬운코딩

04

# 컴퓨팅 사고와 단계

# 3R의 개념 및 Computing Thinking과의 관계

## 다음 지넷 윙 교수의 Computational Thinking에 관한 기고문입니다. 3R은 무엇이고, 3R은 Computing Thinking과 어떤 관계가 있을지 생각해 봅시다.

Viewpoint | Jeannette M. Wing

### Computational Thinking

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately inescapable about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions precisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is

Viewpoint |

modulating something in anticipation of multiple uses or prefetching and caching in anticipation of future use.

Computational thinking is thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction. It is calling gridlock deadlock and contracts interfaces. It is learning to avoid race conditions when synchronizing meetings with one another.

Computational thinking is using heuristic reasoning to discover a solution. It is planning, learning, and scheduling in the presence of uncertainty. It is search, search, and more search, resulting in a list of

ultra; when nondeterminism and garbage collection take on the meanings used by computer scientists; and when trees are drawn upside down.

We have witnessed the influence of computational thinking on other disciplines. For example, machine learning has transformed statistics. Statistical learning is being used for problems on a scale, in terms of both data size and dimension, unimaginable only a few years ago. Statistics departments in all kinds of organizations are hiring computer scientists. Schools of computer science are embracing existing or starting up new statistics departments.

Computer scientists' recent interest in biology is driven by their belief that biologists can benefit

Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction.

Web pages, a strategy for winning a game, or a computer scientist's thinking is using massive amounts of data to speed up computation. It is making trade-offs between time and space and between processing power and storage capacity.

Consider these everyday examples: When your daughter goes to school in the morning, she puts in her backpack the things she needs for the day: that's prefetching and caching. When your son loses his mittens, you suggest he retrace his steps that's backtracking. As what point do you stop reading this and buy yourself a pair? that's the supermarket; which line do you stand in at the online market? that's performance modeling for multi-server systems. Why does your telephone still work during a power outage? that's independence of failure and redundancy in design. How do Complexly Automated Public Turing Text(s) to Tell Computers and Humans Apart, or CAPTCHAs, authenticate humans? that's exploiting the difficulty of solving hard AI problems to fool computing agents.

Computational thinking will have become ingrained in everyone's lives when words like algorithm and precondition are part of everyone's vocab-

ulary; when nondeterminism and garbage collection take on the meanings used by computer scientists; and when trees are drawn upside down.

This kind of thinking will be part of the skill set of not only other scientists but of everyone else. Ubiquitous computing is to today as computational thinking is to tomorrow. Ubiquitous computing was yesterday's dream that became today's reality; computational thinking is tomorrow's reality.

What It Is, and Isn't

Computer science is the study of computation—what can be computed and how to compute it. Computational thinking thus has the following characteristics:

**Conceptualizing, not programming.** Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction. **Fundamental, not rote skill.** A fundamental skill is something every human being must know to function in modern society. Rote means a mechanical routine. Ironically, not until computer science solves the AI Grand Challenge of making computers think like humans will thinking be rote.

**A way that humans, not computers, think.** Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations. **Completeness and combines mathematical and engineering thinking.** Computer science inherently draws on mathematical thinking given that, like all sciences, its formal foundations rest on mathematics. Computer science inherently draws on engineering thinking given that we build systems that interact with the real world. The constraints of the underlying computing device force computer scientists to think computationally, not just mathematically. Being free to build virtual worlds enables us to engineer systems beyond the physical world.

**Ideas, not artifacts.** It's not just the software and hardware artifacts we produce that will be physically present everywhere and touch our lives all the time, it will be the computational concepts we use to approach and solve problems, manage our daily lives, and communicate and interact with other people; and **For everyone, everywhere.** Computational thinking will be a reality when it is so integral to human endeavors it disappears as an explicit philosophy.

Many people equate computer science with computer programming. Some parents see only a narrow range of job opportunities for their children who major in computer science. Many people think the fundamental research in computer science is done and that only the engineering remains. Computational thinking is a grand vision to guide computer science educators, researchers, and practitioners as we act to change society's image of the field. We especially need to reach the pre-college audience, including teachers, parents, and students, sending them two main messages:

**Intellectually challenging and engaging scientific problems remain to be understood and solved.** The problem domain and solution domain are limited only by our own curiosity and creativity; and **One can major in computer science and do anything.** One can major in English or mathematics and go on to a multitude of different careers. Ditto computer science. One can major in computer science and go on to a career in medicine, law, business, politics, any type of science or engineering, and even the arts.

Professors of computer science should teach a course called "Ways to Think Like a Computer Scientist" to college freshmen, making it available to non-majors, not just to computer science majors. We should expose pre-college students to computational methods and models. Rather than bemoan the decline of interest in computer science or the decline in funding for research in computer science, we should look to inspire the public's interest in the intellectual adventure of the field. We'll thus spread the joy, awe, and power of computer science, aiming to make computational thinking commonplace.

JEANNETTE M. WING (wing@cmu.edu) is the President's Professor of Computer Science and head of the Computer Science Department at Carnegie Mellon University, Pittsburgh, PA.



## 3R의 개념 및 Computing Thinking과의 관계

### Computational Thinking에 관한 기고문 (Jeannette M. Wing, 2006)

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. (중략)

Professors of computer science should teach a course called “Ways to Think Like a Computer Scientist” to college freshmen, making it available to non-majors, not just to computer science majors. We should expose pre-college students to computational methods and models.

### [네이버 지식백과] [三 —, 3 Rs] (교육학용어사전)

[3Rs] 독(讀, Reading)·서(書, Writing)·산(算, Arithmetic)을 가리키는 말.  
서양의 고대나 중세사회에서 초등과정 학생들에게 생활의 필요를 충족시키기 위해 가르쳐진 기본적 교육 내용이었다.



## 학습내용

- 컴퓨팅 사고의 배경 및 이해
- 컴퓨팅 사고 교육의 세계 추세 및 생활과의 관계
- 컴퓨팅 사고의 단계 모델과 응용



## 학습목표

- 컴퓨팅 사고와 인간의 사고 체계를 비교하여 설명할 수 있다.
- 각 나라의 컴퓨팅 사고 교육과 생활과의 관계에 대해 설명할 수 있다.
- 컴퓨팅 사고를 하기 전의 단계와 컴퓨팅 사고 단계에 대해 설명할 수 있다.



01

# 컴퓨팅 사고의 배경 및 이해



## 1 | 컴퓨팅 사고의 생성

“2006년 자넷 윙 교수는 ACM 통신 논문  
(Communication of the Association for Computing Machinery)에서  
처음으로 ‘**컴퓨팅 사고**’라는 용어를 탄생시킴”



컴퓨팅적 사고를 바탕으로 세상을 이해함



문제를 이해하고 해결책을 찾는 데 필요한 특별한 렌즈를 얻음



컴퓨터 과학에서 사용하는 여러가지 원리를  
타 학문분야에 적용함



수학이나 과학에만 국한되지 않으며,  
반드시 컴퓨터를 사용할 필요도 없음

## 1 | 컴퓨팅 사고의 생성

**Q** 다음 레고 블록을 즐기기 위해서는?



**A** 두꺼운 직사각형 블록은 전부 한 바구니에,  
얇은 직사각형 블록은 다른 바구니에 넣음

➡ 특별한 기준에 맞춰 블록을 분류하기 시작함



## 1 | 컴퓨팅 사고의 생성

### 해싱

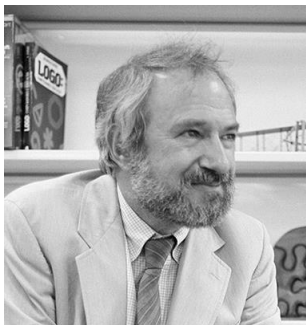
- 특정 규칙에 따라 키를 계산하고 키에 해당하는 주소
- 바로 접근해 데이터를 검색하거나 저장하는 방법



## 1 | 컴퓨팅 사고의 생성

### 컴퓨팅 사고 용어의 유래

#### 시모어 파퍼트(Seymour Papert, 1928~2016)



- MIT 대학교수, 컴퓨터과학자, 수학자, 교육학자
- 1980년 처음 '컴퓨팅 사고'라는 용어를 사용함



1996년에 다시 제안됨

## 1 | 컴퓨팅 사고의 생성

### 컴퓨팅 사고 용어의 유래

2006년 지넷 윙(Jeannette M. Wing) 교수가  
ACM에 컴퓨팅 사고에 대한 글을 게재함



- 컴퓨팅 사고는 누구에게나 일반적으로 적용되는 사고방식임
- 컴퓨팅 사고과정은 문제해결을 위한 데이터 수집과 적절한 논리를 만들기까지 ‘통찰을 얻는 과정’을 총칭함
- 컴퓨팅 사고를 읽고, 쓰고, 셈하는 것과 같은 수준으로 교육해야 함

## 1 | 컴퓨팅 사고의 생성

### 컴퓨팅 사고 용어의 유래

“파퍼트와 지넷 윙 교수의 제안과 주장은  
컴퓨팅 사고에 대한 초석이 됨”

#### 컴퓨팅 사고

- 인간의 사고 능력과 컴퓨터의 능력을 통합한 사고로서, 단편적인 개념에서 벗어나 **복합적인 사고를 통해 창의적으로 문제를 해결**하는 핵심
- 매우 복잡한 문제를 **알고리즘적인 방법**으로 해결하는데 쓰일 수 있음

2 | 인간의 사고 체계



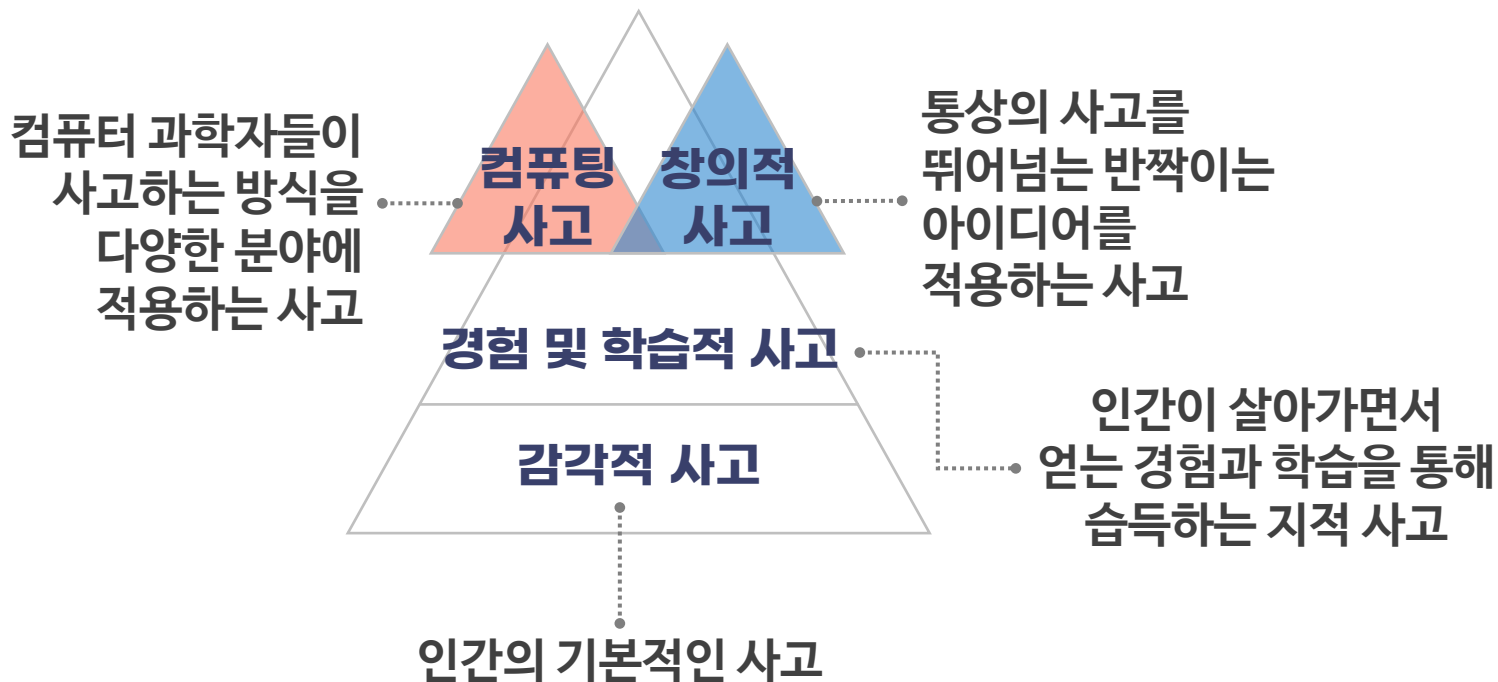
인간의 가장 위대한 능력은 무엇일까요?

“사고하는 능력”



## 2 | 인간의 사고 체계

### 사고의 종류



## 2 | 인간의 사고 체계

### 컴퓨팅 사고의 주요 특징



- 1 인간의 사고와 컴퓨터의 능력을 통합한 사고
- 2 컴퓨터 과학자들이 문제해결을 위해 사고하는 방법에서 출발함
- 3 컴퓨터의 기본 개념과 원리를 기반으로 함
- 4 문제를 효율적으로 해결하는 사고 능력
- 5 복잡한 문제도 일정한 틀에 맞추어 사고하면 효율적임
- 6 최근 컴퓨팅 사고와 관련된 교육과정이 많이 개설됨
- 7 소프트웨어 교육의 최종 목적은 컴퓨팅을 가진 인재를 육성하는 것임
- 8 모든 분야에 활용될 수 있음

## 3 | 지넷 왕의 컴퓨터 사고 요소와 특성

### 왕 교수의 컴퓨팅 사고의 5가지 주요 요소

분 해 (Decomposition)	어려운 문제를 잘게 쪼개어 분할하여 해결할 수 있는 사고
개념화 (Conceptualizing)	여러 단계의 추상화 시작에서 접근할 수 있는 사고
추상화 (Abstraction)	복잡한 문제를 공통부분의 인식을 통해 핵심을 파악할 수 있는 사고
재귀적 사고 (Recursive Thinking)	문제에 대한 해법을 찾고 난 후 그 해법을 문제해결에서 지속적이고 반복적으로 적용할 수 있는 사고
병렬처리 (Parallel Processing)	병렬적으로 해결 방법을 파악하여 문제를 처리할 수 있는 사고



## 3 | 지넷 왕의 컴퓨터 사고 요소와 특성

## 컴퓨팅 사고의 6가지 특성

개 념	설 명
핵심요소	컴퓨팅 사고의 핵심은 프로그래밍이 아닌 개념화에 있으므로 여러 단계의 추상화를 필요로 함
원천기술	컴퓨팅 사고는 틀에 박힌 기술이 아닌 모든 사람이 갖춰야 할 원천 기술임
사고방법	컴퓨터가 아닌 인간의 사고방법으로, 컴퓨터에 인간의 사고방식을 적용하여 복잡한 문제를 해결함
사고의 결합	컴퓨터과학자는 수학적 사고 뿐만 아니라 컴퓨팅 사고도 겸하게 됨
아이디어	컴퓨팅 사고는 문제해결을 위해 고안된 아이디어 발생이 핵심임 → 영향을 크게 줌
대상과 장소	컴퓨팅 사고는 언제 어디서든지 적용될 수 있는 인간 사고에 있어서 필수요소임

## 3 | 지넷 왕의 컴퓨터 사고 요소와 특성

## 컴퓨팅 사고의 10단계 모델(CSTA, ISTE, 구글)

개 념	설 명
데이터 수집	문제해결에 도움이 되는 관련 정보들을 수집함
데이터 표현	데이터를 적절한 그래프, 차트, 단어, 영상 등으로 표현함
데이터 분석	직관적인 판단이나 데이터를 살펴보며 데이터를 분석함
분 해	문제나 데이터를 작고 다루기 쉬운 여러 부분들로 나눔
패턴인식	면밀하게 살펴 패턴, 경향, 규칙성 등의 패턴을 인식함
추 상 화	통합적으로 파악할 수 있는 중심적인 아이디어나 개념을 추출해 냄
알고리즘	문제를 해결하는 일련의 순서화된 명령어인 알고리즘을 만들어냄
시뮬레이션	실제와 같은 효과를 내기 위한 시범모델을 만들어 모의실험함
병 렬 화	여러 개의 업무를 나눈 후 동시에 병렬로 처리함
자 동 화	컴퓨터가 반복적인 업무를 자동적으로 수행하게 함



02

## 컴퓨팅 사고 교육의 세계 추세 및 생활과의 관계



### 1 | 세계 주요국과 한국의 컴퓨팅 사고 교육

#### 미 국

- ‘차세대 과학표준’에서 컴퓨팅 사고를 과학적, 공학적 실천 과제로 설정함
- 다양한 과목에 컴퓨팅 사고를 적용함
- 초·중·고등학교 교육 뿐만 아니라 하버드, MIT, 프린스턴 등 우수 대학에서 컴퓨팅 사고 교육을 진행함

#### 영 국

- 2014년 이후 초·중·고 교과과정에 컴퓨터과학을 의무교육으로 실시함
- ‘컴퓨팅 사고’ 교육을 정규 교과과정으로 채택함

#### 일본, 중국

- ‘컴퓨팅 사고’ 교과목을 정규과목으로 실시함

#### 한 국

- ‘소프트웨어와 코딩’, ‘컴퓨팅 사고’, ‘컴퓨팅 사고와 문제해결’ 등과 같은 과목으로 교양과목을 선정하여 강의하는 추세임



Q

다음 제시되는 두 음악을 들어 봅시다.



Q 다음 제시되는 두 음악을 들어 보시다.



첫 번째 음악

- 베토벤 피아노협주곡 5번 황제 1악장 일부

두 번째 음악

- 베토벤 9번 합창교향곡 1악장 일부

패턴, 분위기, 코드 변화, 화성/대위법 등

➡ 음악의 패턴 파악

“컴퓨팅 사고를 음악에 적용”

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 음악



후렴구 들어보기

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 음악

코드와 화성에  
대한 분석 및  
패턴 인식



음악교육의  
중요한 요소



### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 영어 문장

**“영어를 해석할 때  
кома, 문맥 등을 고려하여 문장을 분해한 후 해석함”**

#### 번역하기

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed.

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 작문

글 쓰기 전에 계획을 세움

1 글의 주제

2 중심이 되는 사건

3 등장인물

4 글의 전개방식

5 글의 마무리

분해하고 단계적으로  
글을 완성시키는 것이  
글짓기를 하는데 효율적임

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 수학문제 풀기

8 다음에 나올 숫자는?

1	2	3	5	8	?	?	?
---	---	---	---	---	---	---	---

#### 피보나치 수열

앞의 수와 뒤의 수를 더한 합이  
그 다음 수가 되는 규칙

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 생활 속의 줄서기

#### 어디에 줄을 설 것인가?



### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 요리



찌개



나물



생선구이

### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 인공지능



### 2 | 컴퓨팅 사고를 적용한 생활 속의 예

#### 인공지능





컴퓨팅 사고는 어떤 태도와 능력 향상에  
도움이 될 수 있을지 생각해 봅시다.





Q

컴퓨팅 사고는 어떤 태도와 능력 향상에 도움이 될 수 있을지 생각해 봅시다.

A

- 1 인간의 사고와 컴퓨터의 능력을 통합한 사고
- 2 컴퓨터 과학자들이 문제해결을 위해 사고하는 방법에서 출발함
- 3 컴퓨터의 기본 개념과 원리를 기반으로 함
- 4 문제를 효율적으로 해결하는 사고 능력
- 5 복잡한 문제도 일정한 틀에 맞추어 사고하면 효율적임

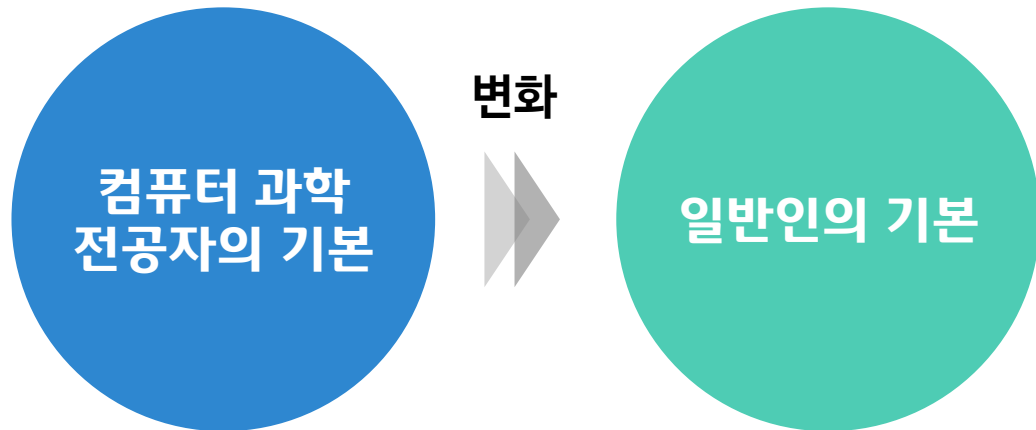
### 3 | 컴퓨팅 사고의 중요성과 코딩과의 관계

#### 컴퓨팅 사고에서 고려해야 할 주요사항

- 1 문제를 해결했을 경우 정확한 답인가?
- 2 그 문제를 가장 효율적인 방법으로 해결할 수 있는 방법인가?
- 3 가장 빠르고 논리적인 해결방법인가?
- 4 시간이나 메모리 등 최소한의 자원을 사용하여 해결이 가능한가?
- 5 그 해결 방법이 다른 문제들을 해결하는 데도 쓰일 수 있는가?

### 3 | 컴퓨팅 사고의 중요성과 코딩과의 관계

#### 컴퓨팅 사고에서 고려해야 할 주요사항



### 3 | 컴퓨팅 사고의 중요성과 코딩과의 관계

#### 컴퓨팅 사고의 장점



C 언어를 마스터하면 다른 프로그램 언어를 배우는 데에도 도움이 됨



프레임을 익히면 다양한 분야에 적용할 수 있음



3 | 컴퓨팅 사고의 중요성과 코딩과의 관계



**컴퓨팅 사고는 일반적으로 우리가  
생활할 때 하는 사고와 다를까?**

소프트웨어  
개발 시  
프로젝트 기획

VS.

신제품 개발 시  
프로젝트 기획

### 3 | 컴퓨팅 사고의 중요성과 코딩과의 관계

#### 최근 코딩과 교육과의 관계



최근 코딩 교육에 대한 관심이 높아지고 있음



미래의 세대에 반드시 제공해야 하는 기본적인 교육임



교육자들은 SW, 코딩, 컴퓨팅 사고, 알고리즘과의  
관계를 명확하게 이해하고 가르치는 것이 필요함

#### 컴퓨팅 사고의 발상

컴퓨팅 능력  
(Power of  
Computing)



깊은 사고력  
(Critical  
Thinking)



컴퓨팅 사고  
(Computational  
thinking)

### 3 | 컴퓨팅 사고의 중요성과 코딩과의 관계

#### 최근 코딩과 교육과의 관계

#### 소프트웨어 교육의 목적

- 코딩을 잘 할 수 있는 프로그래머의 육성도 중요하지만,  
**컴퓨팅 사고를 갖춘 많은 인재들을 양성**하는 데  
초점을 맞추어야 함

**“코딩을 하기 전에 컴퓨팅 사고를 통해  
문제를 효율적으로 해결할 수 있는 방법을  
찾는 것이 매우 중요함”**



03

## 컴퓨팅 사고의 단계 모델과 응용





#### 1 | 컴퓨팅 사고의 7단계 모델



## 1 | 컴퓨팅 사고의 7단계 모델

구 분		설 명
문제분석		<ul style="list-style-type: none"> <li>주어진 문제나 시스템에 대한 논리적 분석을 통해 핵심적인 사항을 구체적으로 점검하고 분석함</li> </ul>
데이터 수집과 표현		<ul style="list-style-type: none"> <li>문제해결과 관련된 정보들을 컴퓨터를 통하여 수집하며, 데이터를 적절한 그래프, 차트, 단어, 영상 등의 형태로 표현함</li> </ul>
컴퓨터 사고	분 해	<ul style="list-style-type: none"> <li>주어진 복잡한 문제를 보다 쉽게 다룰 수 있도록 여러 개의 작은 부분들로 쪼개어 분해함</li> </ul>
	추상화	<ul style="list-style-type: none"> <li>주어진 문제에서 필요 없는 부분들을 걸러내고 꼭 필요한 것만 분리해 내어 집중하는 과정임</li> <li>복잡한 문제나 아이디어를 단순화하고 핵심적인 개념에 초점을 맞추어 일반적인 원리를 찾아냄</li> </ul>

## 1 | 컴퓨팅 사고의 7단계 모델

구 분		설 명
컴퓨터 사 고	패 턴 인 식	<ul style="list-style-type: none"> <li>문제를 보다 효율적으로 해결하기 위해 유사성과 패턴들을 발견함</li> <li>문제나 데이터 내에서의 패턴, 경향, 규칙성 등을 관찰하여 인식하거나 문제 사이에서의 유사성을 살펴 찾아냄</li> </ul>
	알 고 리 즘	<ul style="list-style-type: none"> <li>알고리즘은 주어진 문제 또는 그 문제와 유사한 문제들을 해결할 수 있는 일련의 논리적인 지시인데, 문제에 대한 단계적인 해결, 설명, 지시사항을 설계하게 됨</li> </ul>
평 가		<ul style="list-style-type: none"> <li>알고리즘이 제대로 작동하는지, 문제를 해결하는지, 조건을 만족시켰는지 최종 점검함</li> </ul>

#### 2 | 최근 컴퓨팅 사고 모델과 7단계 모델의 비교

##### 윙 교수의 컴퓨팅 모델과의 비교

개 념 화	패턴인식 및 추상화에 대응되는 개념
재귀적 사고	해법을 알게 되면 그것을 반복적으로 적용시키는 개념
병 렬 화	하나의 일을 여러 개의 업무로 나누어 동시에 병렬로 처리하는 개념

#### 2 | 최근 컴퓨팅 사고 모델과 7단계 모델의 비교

##### 구글 모델과의 비교

<b>시뮬레이션</b>	실제와 비슷한 모델을 만들어 모의로 실험함으로써 문제점을 미리 발견하여 수정할 수 있는 기회를 가짐
<b>병렬화</b>	하나의 일을 여러 개의 업무로 나누어 동시에 병렬로 처리하는 개념
<b>자동화</b>	반복적인 일을 컴퓨터를 이용하여 자동적으로 신속하게 처리하도록 함

➡ 그 이외 7단계 모델은 동일함

### 3 | 컴퓨팅 사고와 문제해결



#### 3 | 컴퓨팅 사고와 문제해결

컴퓨팅 사고 → **콘텐츠** 개발

음악, 방송, 영화, 포탈, 논문,  
게임, 만화 등

컴퓨팅 사고의 4가지 핵심적 개념



### 4 | 컴퓨팅 사고의 응용

#### 분해

#### 분 해

주어진 문제를 여러 개의 작은 크기나 과정들로 나눔





#### 4 | 컴퓨팅 사고의 응용

##### 분해

##### 1 미술

- 쿠카부라 새를 어떻게 그릴 수 있을까?



#### 4 | 컴퓨팅 사고의 응용

##### 분해

#### 1 미술

아이디어  
구상

밑그림  
그리기

색칠하기

마무리  
칠하기

#### 2 문학

- 시를 분석할 때는 그 시를 구조, 음율, 이미지, 어조, 의미 등 다양한 영역으로 분석함

### 4 | 컴퓨팅 사고의 응용

#### 분해

### 3 앱만들기

- 처음 앱을 만들어 본 사람의 경우는 여러가지 상황을 고려해야 함
  - 만들고자 하는 앱의 종류는 무엇인가?
  - 앱의 핵심기능은 무엇인가?
  - 어떤 형태의 앱인가?
  - 앱을 사용할 주된 대상은 누구인가?
  - 앱을 만들기 위해 어떤 소프트웨어를 사용할까?
  - 어떤 그래픽을 사용할까?
  - 어떤 오디오를 포함시킬까?
  - 사용자가 어떻게 앱을 사용할까?
  - 앱을 어떤 방법으로 테스트할 것인가?
  - 앱을 출시하여 어떤 방법으로 수익을 올릴까?

**“ 복잡한 일을 분해해서 처리할 때에는 역할을 명확하고  
효율적으로 나누어 일을 처리할 수 있다는 장점이 있음”**

### 4 | 컴퓨팅 사고의 응용

#### 패턴 인식

#### 패턴 인식

**패턴**이나 경향을 발견하고 인식하여 적재적소에 활용함



일정한 특징, 양식, 유형, 틀



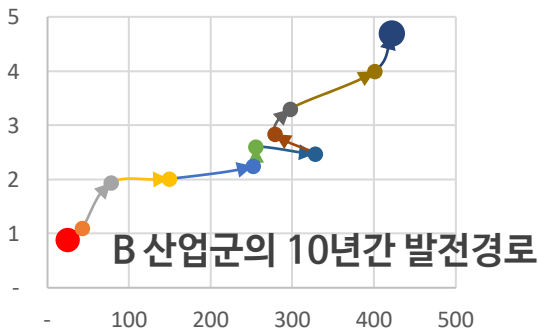
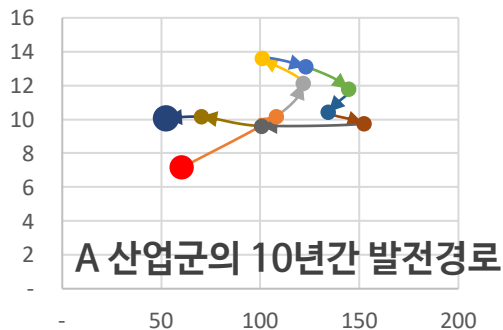
4 | 컴퓨팅 사고의 응용

패턴 인식

1 경영학

- 특정 산업/기업과 관련된 경영지표의 오르고 내리는 경향을 발견하고 그 패턴을 인식하여 경영에 유용하게 활용함

[ 어느 산업군이 바람직한 방향으로 성장하고 있는가? ]



X축 : R&D 규모, Y축 : 매출액, ● : 시작년도, ● : 종료년도

#### 4 | 컴퓨팅 사고의 응용

## 패턴 인식

### 2 경제학

- 주식 거래에서 오르고 내리는 패턴을 발견하고 그 패턴을 인식하여 실제 투자에 반영함



## 4 | 컴퓨팅 사고의 응용

### 패턴 인식

#### 3 달력 규칙 찾아보기

- 빨간색 테두리 안의 숫자를 모두 더하면?
- 파란색 테두리 안의 숫자를 모두 더하면?

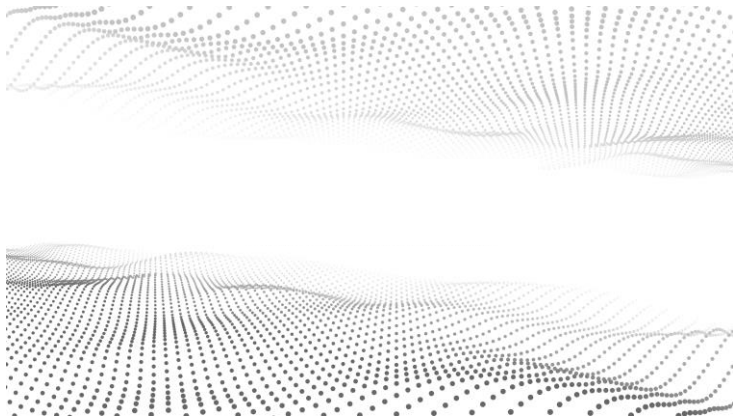
일	월	화	수	목	금	토
30	31	1	2 음 7.15	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17 음 8.1	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3

### 4 | 컴퓨팅 사고의 응용

#### 추상화

#### 추 상 화

패턴과 경향을 규칙, 원리, 일반화, 통찰력을 통해 추상화함





#### 4 | 컴퓨팅 사고의 응용

추상화

#### 1 화학

- 화학반응식

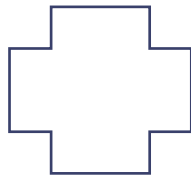
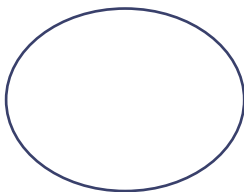
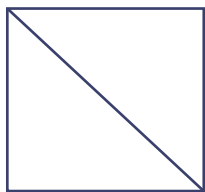
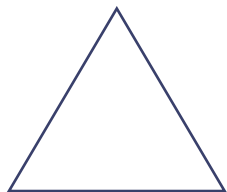


#### 4 | 컴퓨팅 사고의 응용

##### 추상화



**다음 도형 중 특징 추출을 통해서 서로 다른  
한 가지를 찾아내고 그 이유를 밝히시오.**

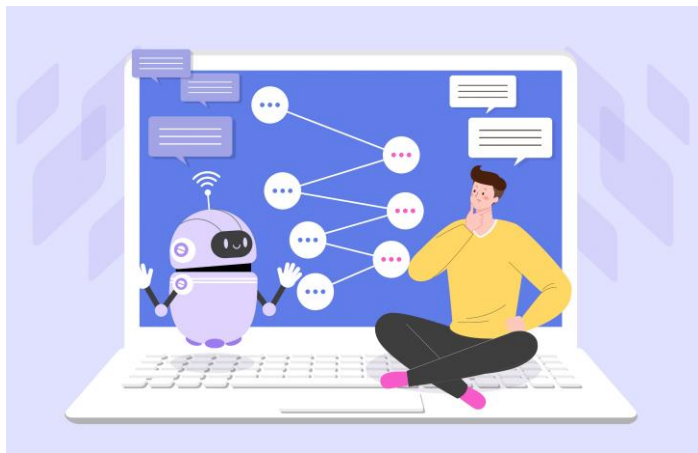


### 4 | 컴퓨팅 사고의 응용

#### 알고리즘

#### 알고리즘

문제를 해결하는 과정이나 어떤 업무처리과정에서의  
순서를 단계적으로 서술함



#### 4 | 컴퓨팅 사고의 응용

##### 알고리즘

##### 1 요리

점선 부근까지  
포장지 뜯으세요.

전자레인지에서  
2분 30초간  
데워주세요.

남은 포장지를  
뜯어 버리고,  
맛있게 드십시오.

이 요?

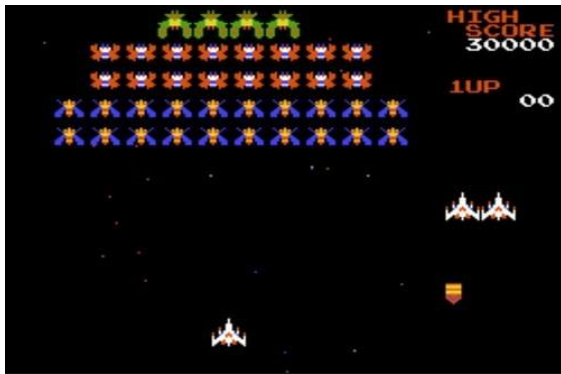


### 4 | 컴퓨팅 사고의 응용

#### 알고리즘

#### 2 게임

- 게임에 있어서 목표지점까지 도달하기 위한 과정을 알아내어 차례로 표현함



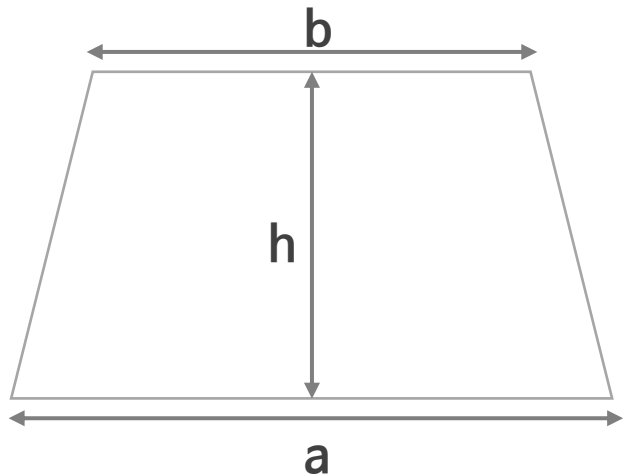
[ 갤러그(GALLAG) 게임 갤러가(GALAGA, ギャラガ) ]

#### 4 | 컴퓨팅 사고의 응용

##### 알고리즘



**사다리꼴 면적을 구하는 알고리즘을 정리해  
봅시다.**





## Q1

다음 중 지넷 웡 교수가 설명하는 ‘Computational Thinking’과 거리가 **먼** 것은?

- 1 컴퓨팅 사고는 누구에게나 일반적으로 적용되는 사고방식이다.
- 2 컴퓨팅 사고는 컴퓨터 사이언티스트에게 최적화된 기술로 일반인이 배우기 어렵다.
- 3 컴퓨팅 사고 과정은 문제해결을 위한 데이터 수집과 적절한 논리를 만들기까지 통찰을 얻는 과정을 말한다.
- 4 컴퓨팅 사고를 읽고, 쓰고 셈하는 것과 같은 수준으로 교육해야 한다.



## Q1

다음 중 지넷 윙 교수가 설명하는 ‘Computational Thinking’과 거리가 **먼** 것은?

- ① 컴퓨팅 사고는 누구에게나 일반적으로 적용되는 사고방식이다.
- ② 컴퓨팅 사고는 컴퓨터 사이언티스트에게 최적화된 기술로 일반인이 배우기 어렵다.
- ③ 컴퓨팅 사고 과정은 문제해결을 위한 데이터 수집과 적절한 논리를 만들기까지 통찰을 얻는 과정을 말한다.
- ④ 컴퓨팅 사고를 읽고, 쓰고 셈하는 것과 같은 수준으로 교육해야 한다.

정답

2

해설

컴퓨팅 사고는 컴퓨터 전문가 뿐만 아니라 모든 사람에게 필요한 읽기, 쓰기, 셈하기와 같은 근본적인 기술입니다.





## Q2

다음 중 통상의 사고를 뛰어넘는 반짝이는 아이디어를 적용하는 사고를 설명하는 단어는 무엇인가?

- ① 감각적 사고
- ② 학습적 사고
- ③ 컴퓨팅 사고
- ④ 창의적 사고





## Q2

다음 중 통상의 사고를 뛰어넘는 반짝이는 아이디어를 적용하는 사고를 설명하는 단어는 무엇인가?

- ① 감각적 사고
- ② 학습적 사고
- ③ 컴퓨팅 사고
- ④ 창의적 사고

정답

4

해설

창의적 사고란 통상의 사고를 뛰어넘는 반짝이는 아이디어를 적용하는 사고를 말합니다.



# Q3

다음 컴퓨팅 사고에서 고려해야 할 내용 중 거리가 먼 것은?

- ① 문제를 해결했을 경우 유일한 답인가?
- ② 그 문제를 가장 효율적인 방법으로 해결할 수 있는 방법인가?
- ③ 가장 빠르고 논리적인 해결방법인가?
- ④ 그 해결 방법이 다른 문제들을 해결하는 데도 쓰일 수 있는가?





## Q3

다음 컴퓨팅 사고에서 고려해야 할 내용 중 거리가 먼 것은?

- ① 문제를 해결했을 경우 유일한 답인가?
- ② 그 문제를 가장 효율적인 방법으로 해결할 수 있는 방법인가?
- ③ 가장 빠르고 논리적인 해결방법인가?
- ④ 그 해결 방법이 다른 문제들을 해결하는 데도 쓰일 수 있는가?

정답

①

해설

문제를 해결했을 경우, 정확하거나 만족도가 높은 답이 바람직합니다.



## Q4

다음 보기에서 설명하는 컴퓨팅 사고의 핵심적인 개념 요소는 무엇인가?

### 〈보기〉

영문을 국문으로 번역할 때 콤마, 문맥, 주요 단어들을 중심으로 문장을 나누고, 전체적인 관점에서 이해하고 해석한다.

① 분해

② 추상화

③ 패턴인식

④ 알고리즘



## Q4

다음 보기에서 설명하는 컴퓨팅 사고의 핵심적인 개념 요소는 무엇인가?

### 〈보기〉

영문을 국문으로 번역할 때 콤마, 문맥, 주요 단어들을 중심으로 문장을 나누고, 전체적인 관점에서 이해하고 해석한다.

1 분해

3 패턴인식

2 추상화

4 알고리즘

정답

1

해설

4분해는 주어진 복잡한 문제를 보다 쉽게 다룰 수 있도록 여러 개의 작은 부분들로 쪼개어 분해하는 것을 의미합니다.



# 정리하기

## 1 컴퓨팅 사고의 배경 및 이해

### ■ 컴퓨팅 사고 개요

- 컴퓨팅 사고는 누구에게나 일반적으로 적용되는 사고방식임
- 컴퓨팅 사고과정은 문제해결을 위한 데이터 수집과 적절한 논리를 만들기까지 ‘통찰을 얻는 과정’을 총칭함
- 컴퓨팅 사고를 읽고, 쓰고, 셈하는 것과 같은 수준으로 교육해야 함



# 정리하기

## 1 컴퓨팅 사고의 배경 및 이해

- 지넷 웡 교수가 제시한 컴퓨팅 사고의 5가지 요소

분 해

개념화

추상화

재귀적  
사 고

병 렬  
처 리





# 정리하기



## 2 컴퓨팅 사고의 단계

