

<http://dx.doi.org/10.17703/JCCT.2019.5.1.297>

JCCT 2019-2-36

컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 개념과 전략

Concept and strategy of unplugged coding for young children based on computing thinking

김대욱

Kim Dae-wook*

요약 본 연구는 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 특성과 개념, 유형, 전략을 알아보는 데 목적이 있다. 유아를 위한 언플러그드 코딩의 핵심은 컴퓨팅 사고력이다. 유아를 위한 컴퓨팅 사고력에 기초한 언플러그드 코딩은 프로그램을 사용하지 않고 알고리즘 판, s-블럭, 코딩 로봇, 스마트 기기 등을 긍정적으로 활용하여 논리적 사고를 기반으로 놀이를 통해 일상 생활 속에서 만날 수 있는 문제를 해결하고 새로운 놀이 방법을 찾는 것이다. 유아를 위한 언플러그드 코딩의 유형은 스마트 기기에 직접 입력하기, 전용 앱을 이용한 코딩 로봇 활용하기, 알고리즘을 활용한 코딩절차 연습하기, 혼합형 방법 사용하기가 있다. 전략으로 알고리즘 이해하기, 순서도 그리기, 작은 부분으로 나누기, 패턴 찾기, 벌레 잡기, 결과 예측하기가 있다.

주요어 : 유아, 컴퓨팅 사고력, 언플러그드 코딩, 개념, 유형, 전략

Abstract This study aims to investigate the characteristics, concepts, types, and strategies of unplugged coding for young children based on computing thinking. The key to unplugged coding for young children is computing thinking. Unplugged coding based on computing thinking for young children can be used to solve problems that can be encountered in everyday life through playing games based on logical thinking by positively utilizing algorithm boards, s-blocks, coding robots, and smart devices without using programs And find new ways to play. Types of unplugged coding for young children include direct input to smart devices, using coding robots with dedicated apps, practicing coding procedures using algorithms, and using hybrid methods. Strategies include understanding algorithms, drawing flowcharts, dividing into smaller parts, finding patterns, inserting, and predicting outcomes.

Key words : Young children, Computing thinking, Unplugged coding, concept, type, strategy

*정회원, 경상대학교 유아교육과
접수일: 2018년 11월 8일, 수정완료일: 2018년 12월 7일
게재확정일: 2019년 1월 11일

Received: November 08, 2018 / Revised: December 07, 2018
Accepted: January 11, 2019
*Corresponding Author: dandi@gnu.ac.kr
Dept. of Early Childhood Education, Gyeong-sang National University, Jin-Ju, Korea

1. 서론

4차 산업혁명의 시대다. 새로운 시대는 지금까지와는 다른 혁신적인 서비스가 현실화될 것이고 그것이 미래 사회의 중추가 될 것으로 보이며[1], 새로운 교육적 기기들을 유아교육에도 적용할 필요가 있다. 여전히 스마트 기기를 제공하는 것에 대해 부정적인 부모나 교사들이 있지만, 유아교육에서 긍정적인 교사-유아의 상호작용은 유아에게 중요한 영향을 미치기 때문에[2], 교사가 새로운 환경인 스마트 기기를 받아들여 제공하는 것은 중요하다.

유아들은 태어나면서부터 자연스럽게 스마트기기를 접하면서 성장한다. 유아의 주변에는 스마트폰이나 태블릿PC와 같은 스마트기기가 어디에나 있다. 이제 유아들은 한국어 언어권에서 태어나 성장하는 것처럼, 스마트 기기와 같은 디지털 언어권에서 살아가는 ‘디지털 네이티브’가 되었다. 더 이상 책상 앞에서만 컴퓨터를 사용하는 시절이 지났고, 컴퓨터를 포함한 다양한 형태와 크기를 가진 스마트 기기들은 어느 곳에서나 존재하고 있다[3]. 인간과 컴퓨터는 서로 사용하는 언어가 다르며 인간이 컴퓨터와 소통하기 위해서는 컴퓨터 언어를 이해할 필요가 있다.

컴퓨팅 사고력은 컴퓨터 과학자 뿐 아니라 누구나 배워서 활용할 수 있는 보편적인 사고이자 기술로 읽기, 쓰기, 셈하기와 같이 유아가 기본적으로 갖춰야 하는 역량과 함께 컴퓨팅 사고력을 포함하여 교육하여야 한다[4]. 컴퓨팅 사고력은 유아가 만날 수 있는 다양한 문제상황에서의 해결책을 제공한다.

컴퓨팅 사고력은 과학자 혹은 과학자를 꿈꾸는 사람에게만 필요한 것이 아니라 미래를 살아가야 할 모든 사람이 갖추어야 할 능력으로[4], 교육을 통해 습득해야 한다. 문제해결에 기반하고 있는 컴퓨팅 사고력은 성인보다 유아가 습득하기에 더 용이하다. 컴퓨팅 사고력에 기초한 언플러그드 코딩은 유아가 컴퓨터를 이해하는데 핵심적인 역할을 한다. 유아는 언플러그드 코딩을 통해 프로그램을 사용하지 않고 부담감을 적게 가지고 컴퓨팅 사고력에 기반하여 코딩에 대한 이해를 높일 수 있다.

코딩 교육의 핵심은 컴퓨터 프로그램을 활용한 능력이 아니라 프로그램을 만들기 위한 컴퓨팅 사고력과 문제해결력이다[7]. 컴퓨팅 사고력은 언플러그드 코딩을 통해 쉽게 습득할 수 있다.

이처럼 유아를 위한 언플러그드 코딩은 컴퓨팅 사고력을 갖추는 데 초점을 맞출 필요가 있다. 사실, 아직까지 유아가 스마트 기기를 사용하는 것에 대해 부정적인 시선이 많고, 스마트 기기의 올바른 사용에 대한 안내나 지침은 거의 없다. 유아기 때부터 컴퓨팅 사고력에 기초한 언플러그드 코딩을 통해 스마트 기기를 다루는 ‘스마트 리터러시(literacy)’를 갖춰야 스마트 기기를 활용하는 바른 태도를 가질 수 있게 된다.

파이썬이나 스크래치와 같은 코딩 프로그램을 이해하기 위해서 프로그램의 작동 원리를 아는 것은 유아가 코딩을 이해하는 데 큰 도움이 되고, 컴퓨터를 사용하지 않고 언플러그드 코딩을 통해 코딩을 배우는 것은 효과적인 방법이다[4]. 유아가 컴퓨팅 사고력을 갖추고 언플러그드 코딩에 대한 긍정적인 경험을 하게 되면 자연스럽게 스마트 기기를 다룰 수 있는 스마트 리터러시를 갖게 될 것이다.

본 연구에서는 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 특성과 개념, 유형, 전략을 분석해보고자 한다.

II. 컴퓨팅적 사고력과 유아 언플러그드 코딩의 특성과 개념

1. 컴퓨팅 사고력의 요소

Denning(2008)은 컴퓨팅 원리를 실행(Comutation), 통신(Communication), 협동(Cordination), 기억(Recollection), 자동화(Automation), 평가(Evaluation), 설계(Design) 7가지로 제시하였다[8]. Denning의 분류는 더 이상 컴퓨팅 원리가 컴퓨터의 운영 및 실행, 관리에 머무르지 않고 다른 학문과의 연계 가능성에 무게를 둔다. 컴퓨팅 사고력의 구성 요소를 자료수집, 자료분석, 자료표현, 문제분해, 추상화, 알고리즘과 절차, 자동화, 시뮬레이션, 병렬화로 구분할 수도 있다[12]. 박지연은 유아 컴퓨팅 사고력을 변환적 사고, 계리적 사고, 코드적 사고, 단순화 사고, 추상화 사고, 분해적 사고, 캐시적 사고, 조정적 공유 사고, 경험론 추론 사고, 절차적 사고의 10가지로 설명하였다[13].

2. 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 특성

프로그램 코딩을 소프트웨어 언어교육이며 기능적인 부분만 있다고 오해하는 경우가 있다. 코딩을 배우는 것은 컴퓨터 프로그램을 만들기 위한 기본적인 개념을 습득하고 컴퓨터처럼 생각하기 위한 문제 해결 방법을 익히는 것으로[4] 결과물을 배우는 것이 아니라 과정을 배우는 것이 더 중요하다.

코딩 프로그램은 많이 나와있고 코딩에 대해 더 어렵게 생각하는 이유가 된다, 이것을 모두 각각의 서로 다른 프로그램으로 생각하기보다 비슷한 프로그램이라고 생각할 필요가 있다. 이를 위해 유아가 코딩 프로그램의 원리를 재미있게 익힐 필요가 있다.

언플러그드 코딩의 핵심은 컴퓨팅 사고력이며, 이는 컴퓨터처럼 생각하는 것이다[6]. 언플러그드 코딩은 코딩의 원리를 쉽고 재미있게 이해할 수 있게 도와준다. 유아가 언플러그드 코딩을 놀이로 경험하면 논리-수학적 능력이 향상될 뿐 아니라, 코딩에 대한 거부감도 없어지며 흥미를 느끼게 될 것이다. 이러한 과정을 통해 유아는 컴퓨팅 사고력을 갖출 수 있게 된다.

3. 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 개념

컴퓨팅 사고력은 2006년 Jeannette M. Wing 교수가 제안한 개념으로 컴퓨팅의 기본 개념과 원리를 바탕으로 문제를 효율적으로 해결할 수 있는 사고 능력을 의미한다[7]. 또한, 컴퓨팅 사고력의 핵심은 프로그래밍이 아니라 개념화에 있다[4]. 유아가 일상생활 속에서 경험을 통해서 컴퓨팅 사고력을 향상시킬 수 있다. 이는 문제해결에 바탕한다. 컴퓨팅 사고력은 새로운 수준의 문제해결 방법을 지원해주고, 컴퓨터 과학의 이론 및 원리를 학습하는 도구가 되기도 하고, 컴퓨터과학을 대중화시켜주는 도구가 된다[9]. Wing에 의하면 컴퓨팅 사고의 핵심은 개념화에 있고, 모든 사람이 갖춰야 하는 핵심 역량이며, 인간의 사고방법이며, 수학적 사고를 보완하며, 모두를 위한 것이다. 컴퓨팅 사고력이 과학기술에 국한되는 것이 아니라 다른 학문 분야와 결합시킬 수 있으며, 더 많은 사고를 가능하게 해준다.

유아의 수준에서 컴퓨팅 사고력은 문제해결력에 초점

을 맞추어 정의되기도 하고[10], 놀이를 통한 컴퓨터 원리 이해에 주안점을 두고 정의되기도 한다[11]. 또한, 유아의 문제해결력을 증진시키기 위해서는 논리적 사고를 향상시켜야 한다. 컴퓨터는 사람보다 더 논리적이며 컴퓨터와 소통할 수 있는 컴퓨터 언어는 논리적 사고에 기초하고 있다. 유아를 위한 컴퓨팅 사고력은 논리적 사고, 문제해결력과 놀이, 컴퓨터 원리 이해 모두를 포함해야 한다.

종합하면, 유아를 위한 컴퓨팅 사고력에 기초한 언플러그드 코딩은 프로그램을 사용하지 않고 알고리즘 판, s-블럭, 코딩 로봇, 스마트 기기 등을 긍정적으로 활용하여 논리적 사고를 기반으로 놀이를 통해 일상 생활 속에서 만날 수 있는 문제를 해결하고 새로운 놀이 방법을 찾는 것이다.

III. 컴퓨팅 사고력에 기초한 유아 언플러그드 코딩의 유형

컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 유형은 스마트 기기에 직접 입력하기, 전용앱을 이용한 코딩 로봇 활용하기, 알고리즘을 활용한 코딩절차 연습하기, 혼합형 방법 사용하기의 네 가지로 분류할 수 있다.

1. 유형 1: 스마트 기기에 직접 입력하기

첫 번째 유형은 스마트 기기에 직접 입력하는 코딩 방법이다. 유아가 스마트 기기에 직접 입력하면서 쉽게 코딩을 시작할 수 있게 해 준다. 대표적인 사례인 코딩 애벌레는 앞으로, 뒤로, 왼쪽, 오른쪽, 노래와 같은 코드를 하나씩 연결하여 움직이는 기기이다.



그림 1. 코딩 애벌레
Figure 1. Coding caterpillar

2. 유형 2: 전용 앱을 이용한 코딩 로봇 활용하기

두 번째 유형은 코딩 로봇을 움직이는 방법이다. 이때 전용 앱이 있는 코딩 로봇을 이용한다. 코딩 로봇을 활용한 방법은 스마트 기기의 앱과 로봇이 직접적으로 연결된 것까지 다르다. 스마트 기기의 전용 앱이 아닌 스크래치나 엔트리를 활용한 것은 코딩의 범주로 볼 수 있으며, 유아기의 발달적 특성을 벗어난다.

전용 앱을 활용한 코딩 로봇의 예는 대시 앤 닷을 들 수 있다. 코딩 로봇은 안드로이드나 iOS 전용 앱을 활용하여 움직일 수 있는데, 로봇만 움직일 수도 있고, figure 3처럼 실로폰과 같은 부속품을 장착하여 움직일 수도 있다.

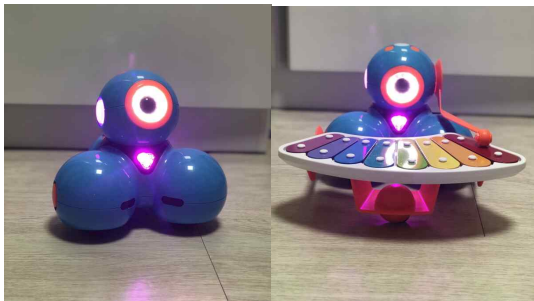


그림 2. 대시앤닷
Figure 2. Dash&Dot

그림 3. 대시앤닷 실로폰
Figure 3. with xylophone

3. 유형 3: 알고리즘을 활용한 코딩절차 연습하기

세 번째 유형인 알고리즘을 활용한 코딩절차 연습하기는 블럭을 활용한 방법과 종이카드를 활용한 방법으로 구분하여 제시할 수 있다.

블럭을 활용한 방법은 코딩 놀잇감이 제공하는 제한된 환경에서 정해진 방법으로 알고리즘을 활용하는 방법이다. 쉽게 순서도를 만들어볼 수 있고 알고리즘을 이해할 수 있는 방법이다. 큐비코는 순서도와 알고리즘의 이해를 용이하게 해 준다.



그림 4. 큐비코
Figure 4. Cubico

블럭을 이용해서 다양한 순서도를 경험한 이후에는 자유롭게 순서도를 활용하여 알고리즘 짜는 연습을 할 수 있다. 꼭 블럭을 활용할 필요는 없지만 다양한 순서도의 예시가 나와있는 코딩 장난감을 활용할 수 있고, 교사나 부모가 만들어서 활용할 수도 있다.

스카티 고는 여러 가지 순서도를 조합하여 알고리즘을 짜볼 수 있는 코딩 놀잇감이다. 이를 통해 알고리즘에 대한 이해를 높일 수 있다.

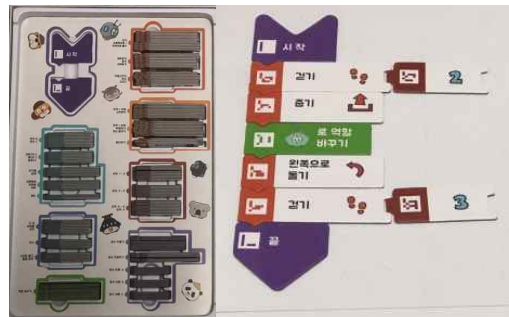


그림 5. 스카티 고
Figure 5. Scottie Go!

4. 유형 4: 혼합형 방법 사용하기

네 번째 유형인 혼합형 방법 사용하기는 s-블럭, 순서도, 종이나 블럭형 알고리즘, 전용 앱, 코딩 로봇을 모두 섞어 사용하는 언플러그드 코딩의 유형이다. 다른 유형들에 모두 익숙해진 이후에 제공해야 더욱 심화된 활동이 일어난다.

모블로는 블럭 놀잇감에 앱을 연동시켜 s-블럭, 순서도, 종이형 알고리즘, 블럭형 알고리즘, 전용 앱을 혼합시킨 코딩 놀잇감이다.



그림 6. 모블로
 Figure 6. Moblo

컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩은 이 네 가지 유형을 순서대로 제시할 때 유아가 편안하게 언플러그드 코딩에 대해 이해하고 쉽게 학습해 나갈 수 있다.

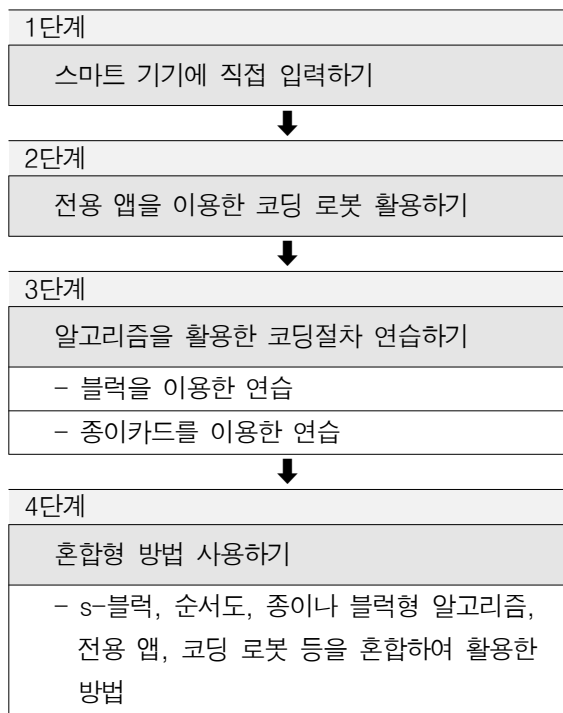


그림 7. 유아를 위한 언플러그드 코딩의 단계
 Figure 7. Steps in unplugged coding for young children

IV. 컴퓨팅 사고력에 기초한 유아 언플러그드 코딩의 전략

컴퓨팅 사고력에 기초한 언플러그드 코딩의 전략으로 Heather Lyons는 작은 부분으로 나누기, 단계별 명

령어 만들기, 버그 잡기, 정글에서 탈출하기, 패턴 찾기, 루프 이용하기, 계획 세우기, 어느 쪽인지 선택하기, 데이터 저장하기를 제시하였고[3], 채성수와 오동환은 문제 분할, 추상화, 자동화, 알고리즘의 절차화, 시뮬레이션, 병렬화를 제시하였다[6]. 노훈은 이진수 시계 놀이, 숫자 맞추기 마술, 동요 압축 놀이, 문제도 순서대로 하면 풀려요, 순서도 만들기, 동물 친구들 키 순서로 줄 세우기, 배열 연습하기, 놀이 공원 그래프 만들기 등의 전략을 제시하였다[7].

앞 절에서 분석한 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 유형과 전략에 대한 선행연구들[3][6][7]의 논의를 중심으로 유아를 위한 언플러그드 코딩에서 필요한 것을 알고리즘 이해하기, 순서도 그리기, 작은 부분으로 나누기, 패턴 찾기, 벌레 잡기, 결과 예측하기로 도출하였다.

1. 전략 1: 알고리즘 이해하기

문제를 해결하기 위한 순서와 차례, 규칙이나 방법을 알고리즘이라고 하며[7], 컴퓨팅 사고력에서만 중요한 것이 아니라 유아가 경험하는 모든 사회적 상황에서 접할 수 있다. 간단한 알고리즘부터 복잡한 알고리즘까지 모든 알고리즘에는 입력, 출력, 명확성, 유한성, 효과성의 5가지 원칙이 적용된다[7]. 유아는 알고리즘에 대해 이해하고, 논리적으로 자신이 놓인 상황들을 설명할 수 있게 된다.

2. 전략 2: 순서도 그리기

유아가 언플러그드 코딩을 시작하면서 어떤 활동을 할 것인지에 대한 계획을 세우는 것이다. 순서도 그리기는 계획 세우기에 해당한다. ‘시작/끝’, ‘처리’, ‘판단’, ‘화살표’의 의미를 이해하고 논리적으로 순서도를 그려본다. 처음에는 유아가 일상생활 속에서 경험하게 되는 작은 사건들을 가지고 순서도를 그려본다.

3. 전략 3: 작은 부분으로 나누기

유아는 언플러그드 코딩을 어렵게 느낄 수 있다. 이럴 때 유아가 문제 상황을 작은 부분으로 구분지어 살펴보는 연습을 할 필요가 있다. 유아가 수학 개념이 들

어간 새로운 놀잇감을 가지고 놀이하는 것을 어려운 다면 부분부분으로 나누어서 놀이하도록 제안할 수 있다. 놀이 방법을 순서대로 잘게 나누면 놀이 방법을 찾아낼 수 있다. 작은 부분으로 나누는 것을 코딩 용어로 ‘분해’라고 한다.

4. 전략 4: 패턴 찾기

패턴 찾기는 언플러그드 코딩이 아니더라도 유아가 일상생활 속에서 많이 하는 활동이다. ‘사과-마나나-사과->??’의 그림을 순서대로 제시하고 ‘??’에 나올 그림을 추론해보는 방법이다. 유아가 다양한 패턴을 찾아보면서 논리적 추론 능력을 향상시키게 되면 언플러그드 코딩을 하면서 많은 도움이 될 것이다.

5. 전략 5: 벌레(bug) 잡기

유아가 언플러그드 코딩을 하면서 어려워할 수 있는 부분이다. 다른 활동을 하면서 흔히 경험하지 않기 때문이다. 생활 속에서 만나게 되는 벌레는 눈에 보이지만, 언플러그드 코딩을 할 때 버그는 눈에 보이지 않는다. 유아가 인내심을 갖고 차분하게 검토하도록 한다. 명령어 하나하나가 제대로 작동하는지 살펴보는 일이다. 누구나 할 수 있는 일이지만, 꼼꼼하게 하는 일은 지루한 일이기 때문에 유아가 하기 싫어할 수 있다. 재미있게 벌레 잡기를 할 수 있도록 흥미를 유발할 수 있는 사례들을 제공해야 한다.

6. 전략 6: 결과 예측하기

유아가 알고리즘을 이해하여 결과를 예측하게 되면 언플러그드 코딩을 완성할 수 있다. 그림책에서 많이 활용하는 방법인 플랩(flap)을 적용할 수 있다. 유아의 흥미를 유발하기 위해, 결과 부분을 가려놓고 결과를 예측해보고 플랩(flap)을 열어 확인하도록 하는 방법도 있다. 예측한 결과가 맞는지 틀렸는지 확인해보고, 왜 그러한 결과에 도달했는지도 알아본다. 정답을 맞추는 것이 중요한 것이 아니라 정답에 도달하는 논리적인 과정이 더 중요하다.

이 6가지 전략은 하나씩 적용할 수도 있고, 순차적으로 적용할 수도 있다. 동기부여를 위해 결과 예측하기

부터 할 수도 있다. 하지만, 6가지 전략을 순서대로 적용시킨다면 유아의 언플러그드 코딩에 대한 이해도를 높일 수 있을 것이다.

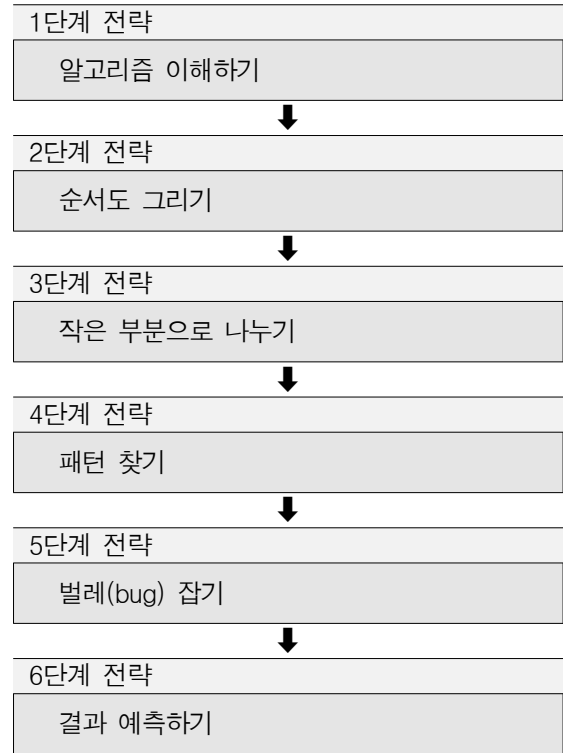


그림 8. 유아를 위한 언플러그드 코딩의 전략
Figure 8. Strategy in unplugged coding for young children

V. 결론

컴퓨팅 사고력은 언플러그드 코딩의 핵심 요소이다. 컴퓨팅 사고력은 논리적 사고이며 유아가 일상생활 속의 문제를 해결해가면서 습득할 수 있다. 유아가 언플러그드 코딩에 대해 어려워할 수 있고 일상 생활과 완전히 별개의 것이라고 느낄 수 있다. 하지만, 유아를 둘러싼 환경이 예전과는 크게 달라져 스마트 기기가 없는 사회는 상상할 수 없다. 유아가 그림책을 통해 문해 능력을 획득하듯, 컴퓨팅 사고력에 기초한 언플러그드 코딩을 통해 스마트 리터러시를 습득해야 한다.

현대 사회에서는 수많은 정보와 지식을 손쉽게 구할 수 있다. 그러나 나에게 꼭 필요한 정보와 지식을 어떻게 구하는 것은 또 다른 문제이다. 수많은 정보와 지식을 자신에게 맞게끔 찾고 가공하는 것은 현대를 살아가는 데 있어서 갖춰야할 중요한 능력이다. 컴퓨팅 사고

력에 기초한 언플러그드 코딩을 경험해보지 못한 유아는 이후에 코딩교육에서도 도태되어 디지털 디바이드(digital divide)가 되어 버릴 수가 있다. 부모의 지나친 보호 아래 문제해결력을 기를 기회를 잃어버리고 디지털 디바이드화 되어버리는 것은 좋지 않다. 유아가 언플러그드 코딩을 통해 컴퓨팅 사고력을 갖추면 시대의 빠른 변화에도 선제적으로 대처할 수 있게 된다.

유아는 컴퓨팅 사고력에 기초한 언플러그드 코딩에 흥미를 보이지 않을 수 있다. 언플러그드 코딩은 프로그래밍 툴을 다루는 기술 교육이 아니라, 문제해결에 기반한 놀이를 통한 긍정적인 코딩 경험을 추구한다. 유아는 학습이 아니라 놀이를 한다고 생각하면서 언플러그드 코딩을 통해 다양한 프로그래밍 기초 개념을 습득할 수 있다. 논리력에 기반을 한 컴퓨팅 사고력과 언플러그드 코딩을 유아기 때부터 연습한 사람은 나중에 두려움 없이 사회에 필요한 인재가 될 수 있을 것이다.

References

- [1] Seung-Hyeog, Moon, Analysis of ICT Converged Smart Factory and its Driving Strategy. *The Journal of the Convergence on Culture Technology (JCCT)*, 4(3), 235-240, 2018.
- [2] Eun-Ha, Her, & Sang-Lim, Kim, Early childhood teachers' professionalism, teaching efficacy, and teacher-child interaction. *International Journal of Advanced Culture Tehchnology (IJACT)*, 6(4), 39-44, 2018.
- [3] Heather Lyons & Elizabeth Tweeddale, Alex Westgate, *Kids get coding: Learn to program*. UK, London: Hodder and Stoughton Limited, 2016.
- [4] Wing M. Jeannett, Computational Thinking. "Communications of the ACM", 49(3), 33-35, 2006.
- [5] Kiki Prottzman, *My First Coding Book*. UK, London: Dorling Kindersely Limited, 2017.
- [6] Seong-Su, Chae, & Dong-Whan, Oh, *Computing thinking for Coding*, Seoul: Hyun Books, 2017.
- [7] Hun, Noh, *Are you new to computer-free coding?* Seoul: Jpub, 2018.
- [8] Denning, Peter, J, *Great Principles of Computing*. Calhoun: NPS Institutioanl Archive. Online Available <http://hdl.handle.net/10945/35523>, 2008.
- [9] Soo-Bum Shin, Chul Kim, Nam-Je Park, Kap-Su Kim, Young-Hoon Sung, Young-Sik Jeong, *Convergence Organization Strategies of the Computational Thinking in Informatics Curriculums*. *Journal of The Korean Association of Information Education*, 20(6), 607-616, 2016.
- [10] Sang-Un, Kim, Effectiveness of Education Program using Robot for Young Children's Computational Thinking, Kongju National University graduate school doctoral thesis, Kongju, Korea, 2018.
- [11] Jeong-Min, Kim, & Il-kyeong, Hong, Kyeong-Min, Kim, A study on teaching materials development for Computational Thinking through play. *Korea Computer Education Association Conference Bullentin*, 20, 2, 187-190, 2016.
- [12] Kyung-Sun, Oh, A study on the Contents of Computational Thinking for Programming Education. *Sung-Kyun-Kwan University graduate school doctoral thesis*, Seoul, Korea, 2016.
- [13] Ji-Yeon, Park, Analysis of the relationship between computational thinking ability and young children play, *Korea University graduate school of education master's thesis*, Seoul, Korea, 2012.