

# 컴퓨터 비전공자를 위한 기초 코딩 교육에서 학습자의 시각적 문해력과 사고 유형이 프로그램 이해와 작성에 미치는 영향 분석

## Analysis of the Effects of Learners' Visual Literacy and Thinking Patterns on Program Understanding and Writing in Basic Coding Education for Computer Non-majors

박찬정<sup>†</sup> · 현정석<sup>††</sup>

Chan Jung Park<sup>†</sup> · Jung Suk Hyun<sup>††</sup>

### 요 약

최근 소프트웨어와 인공지능 교육이 점차 중요하게 다루어지면서 2019년 12월 과학기술정보통신부는 2022년까지 초·중등학교에서 소프트웨어와 인공지능 교육을 필수교육으로 확대하는 계획을 발표하였다. 초·중등학교에서는 물론, 대부분 대학교에서도 컴퓨터 비전공자들을 대상으로 한 소프트웨어 교육이 활발히 이루어지고 있지만, 컴퓨터 비전공자들을 대상으로 하는 코딩 교육에 관한 연구는 부족한 실정이다. 본 논문에서는 대학의 교양 과목으로 컴퓨터 비전공자를 대상으로 한 코딩 교육에서 효율적인 교수학습방법을 찾고자 한다. 밀레니얼 세대와 Z세대로 불리는 요즘 대학생들은 시각적 정보를 선호하고 디지털 네이티브로 컴퓨터에 익숙하다. 이런 특징을 기반으로 대학생들의 시각적 문해력과 사고 유형을 조사한 후, 기초 코딩 과목에서 학생들의 시각적 문해력과 사고 유형이 프로그램을 이해하고 작성하는 코딩 기반의 문제해결력에 영향을 미치는지 살펴보았다. 이를 토대로 코딩이 처음인 컴퓨터 비전공자 학생들을 위한 코딩 활동에 대해서 보다 효율적인 교수학습방법에 대한 대안을 제시하였다.

**주제어:** 코딩 교육, 시각적 문해력, 사고 유형, 코딩 기반 문제해결력, 교양 교육

### ABSTRACT

As software and artificial intelligence education became more and more important, in December 2019, the Ministry of Science and ICT announced plans to expand software and AI education to mandatory education in elementary and secondary schools by 2022. In addition to elementary and secondary schools, most universities are actively engaged in software education for computer non-majors, but research on coding education for computer non-majors is insufficient. The purpose of this paper is to find an efficient teaching and learning method for coding education for computer non-majors. Nowadays, college students, called Millennial and Generation Z, prefer visual information and are familiar with computers as digital natives. Based on these characteristics, this study examined the visual literacy and thinking styles of college students and then examined whether the students' visual literacy and thinking styles influenced coding-based problem solving in coding subjects. Based on this, this paper proposes an alternative to do programming education more efficiently for students who are new to coding.

**Keywords:** Coding Education, Visual Literacy, Thinking Patterns, Problem Solving Ability Based on Coding, Literal Art Education

## 1. 서론

21세기 지식정보사회를 맞이하면서 초·중등학교에서

“창의융합형 인재”를 양성하기 위한 2015 개정 교육과정  
이 고시되었다. 개정 교육과정의 큰 특징으로서 중학교  
에서 정보 과목이 필수교과로 지정되었으며, 고등학교에

<sup>†</sup> 중신회원: 제주대학교 컴퓨터교육과 교수

<sup>††</sup> 중신회원: 제주대학교 경영정보학과 교수(교신저자)

논문접수: 2020년 01월 16일, 심사완료: 2020년 02월 10일, 게재확정: 2020년 02월 28일

\* 본 논문은 2019학년도 제주대학교 교원성과지원사업에 의하여 연구되었음.

서도 정보 과목이 일반 선택과목으로 바뀌었다[1][2]. 이와 같은 변화로 최근 초·중등학교에서 진행되는 소프트웨어 교육콘텐츠 및 교육방법 등에 관한 연구가 활발하게 진행되고 있다[3][4][5].

이와 더불어 2015년도부터 대학을 위한 소프트웨어 교육에 변화가 생겼다. 4차 산업혁명의 도래로 우리나라 대학의 소프트웨어 교과와 교육의 체제를 새롭게 혁신하고 소프트웨어 전문가와 융합 전문가를 양성하기 위해 소프트웨어(SW) 중심대학 사업이 시작되었다. 2019년까지 전국적으로 40개의 대학이 소프트웨어 중심대학 사업에 참여하고 있으며 컴퓨터 비전공자들을 대상으로 하는 소프트웨어 기초 교육이 의무화되어 실시되고 있다[6].

최근 교양 과목에서 소프트웨어 교육의 하나로 코딩 교육이 진행되고 있다. 소프트웨어 교육을 효율적으로 운영하기 위한 콘텐츠와 방법들에 관한 연구가 부족한 실정이다[7]. 컴퓨터 비전공자를 위한 코딩 교육은 코딩 기술에 초점을 두기보다 문제를 해결하고 SW 전문가와 협업하기 위한 도구에 대한 교육에 치중하고 있다[8].

따라서 컴퓨터 비전공자를 대상으로 하는 코딩 교육에서 사용되는 언어는 대부분 SW 전문가와 원활한 소통과 협업에 도움을 줄 수 있는 Scratch [9]와 App Inventor [10]와 같은 블록(block) 기반의 시각적 언어를 활용하고 있다. 블록 기반의 시각적 언어들은 학생들이 원하는 블록을 찾아 맞추어가며 코딩을 진행하기 때문에 Java와 C와 같은 전통적인 프로그래밍 언어보다 쉽게 이해할 수 있다[7]. 블록 기반의 시각적 언어들은 2018년도부터 초·중·등학생 교육을 위한 교육용 프로그래밍 언어로 많이 사용되고 연구되어왔다.

최근 인공지능(Artificial Intelligence, AI)에 관한 관심이 증가하고 데이터 과학 응용 개발에 대한 요구가 증가하면서, 배우기 쉽고 시각화가 편리하며 인공지능 앱 개발이 가능한 Python 프로그래밍 언어가 대학에서 컴퓨터 비전공자를 교육하기 위한 언어로 자리매김하고 있다 [11][12][13]. Python 언어는 효율적이며, 그래픽 처리 기능이 단순하여 다양한 앱을 만들 수 있어서, 초보자들에게 교육하기 좋은 언어로 간주되고 있다[11][12][13]. 소프트웨어 품질 생산 관리 업체인 티오베(Tiobe)의 2020 프로그래밍 선호도 순위 지수(<https://www.tiobe.com/tiobe-index/>)에서 Java와 C를 제외하고 Python 언어가 3위이고 교육용 프로그래밍 언어 지수(<http://pypl.github.io/PYPL.html>)를 측정하는 사이트에서도 Scratch와 더불어 선두로 자리매김하고 있다[14].

이에 국내 여러 대학이 1학년을 대상으로 하는 교양 과목에서 Python 언어를 선택하고 필수로 가르치고 있다.

미국 대학도 마찬가지이다. 컴퓨터 전공이 개설된 대학 중에서 상위 69%가 Python 언어를 채택하여 교육하고 있다. 이처럼 국내·외에서 Python 언어는 대학의 컴퓨터 비전공자에게 적합한 언어로 판단할 수 있다[15]. 본 논문에서는 Python 프로그래밍을 다루는 한 대학의 교양 과목을 수강한 컴퓨터 비전공자 학습자들을 대상으로 프로그래밍을 이해하고 작성하는 코딩 기반의 문제해결력을 측정한다. 이를 위해 Python 프로그래밍 요소를 부분적으로 묻는 <문제해결력 측정문항군 1>과 전체 프로그래밍 요소를 포함한 <문제해결력 측정문항군 2>를 바탕으로 어떤 특성을 가진 학생들이 높은 문제해결력을 갖는지 분석하고자 한다. 문항은 Python 언어의 기본 내장 모듈인 터틀(Turtle) 모듈을 기반으로 한 기초적인 그래픽 사용자 인터페이스(Graphical User Interface: GUI) 프로그래밍에 초점을 두었다. 본 논문에서 채택한 프로그래밍 요소로는 프로그래밍 언어론[16][17]에서 다루는 변수, 연산자, 수식, 선택, 반복, 함수, 클래스를 두었다.

학생들의 개인 특성은 시각화가 소프트웨어 개발에 핵심 요소라고 주장하는 이론[18][19][20]에 근거하여 컴퓨터 비전공 학생들의 시각적 문해력(Visual Literacy)을 선택하였다. 실생활의 문제를 프로그래밍으로 해결할 때 사용하는 데이터흐름도(Data Flow Diagram), UML(Unified Modeling Language), 순서도 등은 모두 시각화된 다이어그램이다. 또한, 본 논문에서는 학생들의 추상적이거나 구체적인 사고 유형이 코딩에 영향을 미친다는 기존 연구[21][22]를 바탕으로 학생들의 사고 유형을 프로그래밍에 영향을 미치는 요소로 선택하였다. 본 논문에서는 컴퓨터 비전공자 학생들의 개인 특성 요인 중 시각적 문해력 요인을 축소화하여 시각적 자료의 분석과 사용, 시각적 자료의 제작, 시각적 자료의 평가 요인으로 분류하였다. 이 요인들에 대한 측정항목들을 분석하여 프로그램을 이해하고 작성하는 코딩 기반 문제해결력과 어떤 관계를 갖는지 분석한다. 또한, 시각적 문해력 요인과 사고 유형 요인 간 상호작용 효과가 존재하는지를 파악하여 컴퓨터 비전공자를 위한 효과적인 수업방안을 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 적용한 시각적 문해력의 개념과 설문 문항을 기술하였다. 아울러 추상적/구체적 사고수준을 측정할 수 있는 설문 문항을 제시한다. 3장에서는 설문에 사용될 측정 문항을 기술한다. 4장에서는 컴퓨터 비전공자 학습자의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력을 기반으로 개인 특성 요인과 해당 문제해결력 간의 관계 분석을 실행한다. 5장에서는 컴퓨터 비전공자를 위

한 수업 요소와 방법을 제시한 후, 결론을 맺는다.

## 2. 연구 배경

### 2.1 시각적 문해력

요즘 세대를 Z세대 또는 밀레니얼 세대라 일컫는다. 요즘 세대의 대표적인 특징 중 하나가 바로 직관적이고 빠르게 정보를 이해할 수 있는 이미지 즉 시각적 요소를 더욱 추구한다는 점이다[23].

시각적 문해력은 21세기 필수적 역량으로 자리매김하고 있다[24]. 시각적 문해력은 다양한 미디어 시대에 시각적 제품과 메시지를 올바르게 이해하고 인식하고 해석하고, 실제 생활에서 의미 있게 사용하고, 다른 사람들과 함께 제작, 분석, 평가 및 의사소통하는 능력으로 정의된다[24][25][26].

시각적 문해력에 대한 정의는 1969년 J. L. Debes[27]의 연구를 시점으로 하여, 2011년 미국 도서관 협회의 한 부서인 대학과 연구 도서관 협회(Association of College and Research Libraries: ACRL)의 정의에 이르렀다[24][27].

초기의 시각적 문해력은 시각적 메시지 이해, 읽기, 사용, 해석, 평가로 정의되었으나, ACRL은 다음과 같이 7가지 표준을 고등 교육을 위한 시각적 문해력 역량 기준(Visual Literacy Competency Standards)[30][33]으로 제시하였다. ACRL의 시각적 문해력 역량 평가 문항을 기반으로 본 연구에서 사용한 평가 문항은 다음 <표 1>과 같다.

표준 1: 필요한 시각적 자료의 성질과 정도를 결정한다.

표준 2: 시각적 기술을 사용하여 시각적 매체를 찾아내고 시각적 매체에 효과적으로 접근한다.

표준 3: 이미지와 영상 매체의 의미를 해석하고 분석한다.

표준 4: 이미지와 이미지 출처(source)를 평가한다.

표준 5: 이미지와 시각적인 미디어를 효과적으로 사용하는 시각적 감각에 친근하다.

표준 6: 의미 있는 이미지와 시각 매체를 디자인하고 제작한다.

표준 7: 시각적 매체에 대한 윤리적, 법적, 사회적 및 경제적 측면을 이해한다.

시각적 문해력은 컴퓨터 프로그래밍 영역에서 중요한 역할을 하고 있지만, 국내에서는 미술 교육에서 다루어질 뿐, 컴퓨터 프로그래밍이나 컴퓨터 과학 교육과의 관련성을 보여줄 연구가 거의 없다.

### 2.2 추상적 사고와 구체적 사고

정보 교과의 2015 개정 교육과정에서 학습자들이 지녀야 할 핵심 역량 중 하나가 컴퓨팅 사고력(computational thinking)이다. SW 중심대학에서 진행하고 있는 컴퓨터 비전공자를 대상으로 한 교육에서도 컴퓨팅 사고력 함양을 교육의 목표로 지정하고 있다. 컴퓨팅 사고력에서는 추상화(abstraction)와 자동화(automation)를 위한 추상적 사고와 구체적 사고가 모두 필요하다.

행동 개념화 이론(action identification theory) [28][29]에서는 사람들의 사고 유형에 따라서 특정 행동을 구체적이고 방법적으로 ‘어떻게(how) 할 것인가’하는 목표를 달성하기 위한 수단과 관련지어 해석하기도 하고, 추상적이고 목적과 관련지어 ‘왜(why) 행동을 해야 하는지’로 해석하기도 한다.

표 1. 시각적 문해력 문항

표준	문항
1	① 나는 일상생활에서 이미지(그림, 사진 등)의 필요성을 명확하게 말합니다.
	② 나는 다양한 이미지 자료, 자료 및 유형을 식별(구별)합니다.
2	③ 나는 가장 적절한 출처와 검색 시스템을 선택하여 필요한 이미지 및 시각적 미디어를 정확하게 찾아냅니다.
	④ 나는 일상생활에서 이미지 검색을 효율적으로 수행합니다.
	⑤ 나는 이미지와 출처 정보를 수집하고 잘 구성합니다.
3	⑥ 나는 이미지의 의미와 관련된 정보를 잘 식별합니다.
	⑦ 나는 이미지를 문화적, 사회적, 역사적 맥락에서 이해합니다.
	⑧ 나는 이미지의 물리적, 기술적, 디자인 요소를 식별합니다.
	⑨ 나는 이미지에 대한 해석을 다른 학생과 대화로 검증합니다.
4	⑩ 나는 시각적 의사소통 방법인 이미지의 효과와 신뢰성을 잘 평가합니다.
	⑪ 나는 이미지의 미적, 기술적 특성을 잘 평가합니다.
	⑫ 나는 이미지 출처의 신뢰성과 정확성에 관해 판단을 잘 내립니다.
5	⑬ 나는 다양한 목적으로 효과적으로 이미지를 사용합니다.
	⑭ 나는 기술을 효과적으로 사용하여 이미지 작업을 합니다.
	⑮ 나는 숙제를 할 때 이미지를 문서와 통합하기 위해 문제 해결력과 창의력을 사용합니다.
6	⑯ 나는 프로젝트 및 학술적 용도를 위한 다양한 시각 자료를 제작합니다.
	⑰ 나는 이미지 및 시각적 미디어 제작에서 디자인 전략과 창의력을 사용합니다.
	⑱ 나는 다양한 도구와 기술을 사용하여 이미지와 비주얼 이미지를 제작합니다.
7	⑲ 나는 이미지 및 영상 매체의 윤리적, 법적, 사회적, 경제적인 문제를 많이 이해합니다.
	⑳ 나는 서류, 프레젠테이션, 프로젝트에서 이미지와 시각적 매체를 인용합니다.

표 2. BIF 문항 사례(진술문 중 택일)

상황	진술문	
나무에 오르기	① 좋은 경치 보기	② 나뭇가지 붙잡기
집 청소하기	① 개인의 청결함을 보여주기	② 진공청소기로 방 청소하기
방을 페인트로 칠하기	① 붓칠하기	② 방을 새롭게 보이게 만들기
임대료 내기	① 살 장소를 유지하기	② 돈을 주기
실내 화초 기르기	① 식물에 물주기	② 방을 멋있어 보이게 만들기
...		

예를 들면, 같은 사건이라도 가까운 시간과 먼 시간에 따라 다르게 해석이 되는 경우가 있다. 이사를 6개월 뒤에 할 때는 “새로운 곳에서 삶을 시작한다”로 해석하는 반면에 며칠 뒤에 이사하는 경우에는 “이삿짐을 포장한다”라고 해석할 수 있다. 하나의 사건에 대해 높은 수준의 해석을 할 때는 행동의 추상적 측면인 목적과 관련된 해석을 하지만, 낮은 수준의 해석을 할 때는 행동의 구체적인 측면인 수단과 관련된 해석을 한다.


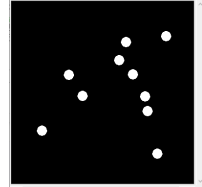
Vallacher와 Wegner는 사람들의 사고수준을 구할 수 있는 행동 개념화 척도(Behavioral Identification Form : BIF) 문항 25개를 개발하였다[30][31]. 행동 개념화 척도를 이용한 사고수준의 측정은 각 문항을 1점씩 계산하여 총 점수가 높을수록 추상적인 사고를 하고, 점수가 낮을수록 구체적인 사고를 하는 것으로 분석한다. 본 연구에서 사용한 문항의 사례는 다음 <표 2>와 같다.

한편 해석수준이론(Construal Level Theory) [30][31]은 사람들이 느끼는 심리적 거리에 따라 대상에 대한 해석 수준을 다르게 표상한다는 이론이다. 즉, 심리적 거리가 멀수록 추상적인 목적과 의사결정의 본질적인 측면에 집중하는 반면에 심리적 거리가 가까울수록 구체적인 수단과 의사결정의 실행 가능성에 초점을 둔다고 설명한다. 해석수준이론과 관련된 기존 연구들에 의하면 개인별로 추상적 사고수준에서 차이가 있는 것으로 조사되었다 [30][31]. 본 연구는 컴퓨터 비전공자들의 사고수준을 BIF로써 측정하였다[28][29]. 행동 개념화 척도 값이 컴퓨터 비전공자 학생들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 어떻게 영향을 미치는지 살펴본다면, 컴퓨터 비전공자들의 코딩 교육에 대한 유용한 시사점을 제시할 수 있다.

### 3. 시각적 프로그램을 작성하고 이해하는 코딩 기반 문제해결력 측정 문항

본 연구에서는 2019학년도 진행된 대학 교양 과목을

표 3. 코딩 기반의 문제해결력 측정 문항 예제

유형	문항 예제
<문제 해결력 측정 문항군 1>	<pre>import turtle as t t.shape('turtle') colors=["red", "pink", "orange"] for i in range(3):     t.begin_fill()     t.fillcolor(colors[i])     t.pu()     t.goto(-200+ i * 50, 0)     t.pd()     t.circle(70)     t.end_fill()</pre>
<문제 해결력 측정 문항군 1>	<p>다음 그림과 같이 계단을 만드는 프로그램을 작성하십시오. 계단의 높이는 50, 길이는 100이며, 색상은 초록색입니다.</p> 
<문제 해결력 측정 문항군 2>	<pre>import turtle as t def drawimage(x, y):     for i in range(x):         for j in range(y):             t.fd(100)             t.lt(360/y)             t.circle(50)             t.lt(360/x)  drawimage(3, 4)</pre>
<문제 해결력 측정 문항군 2>	<p>다음 그림과 같이 터틀 창안에 눈송이 10개를 임의의 위치에 그리고자 한다. 눈의 반경은 10픽셀이다. 위치는 x 좌표값과 y 좌표값 모두 -150~150 사이의 임의의 위치값을 갖게 한다. 다음 함수를 완성하십시오.</p> <p style="text-align: center;">[수행결과]</p> 

수강하면서 설문조사에 동의한 학생들 61명을 대상으로 분석하였다. 학생들의 동의하에 BIF 값과 시각적 문해력을 측정하였다. 시각적 문해력을 확인하기 위해 터틀을 이용한 코딩 기반의 문제해결력 측정 문항 예시를 <표 3>과 같이 제시하였다.

대학교의 교양 과목으로 개설된 코딩 수업에서 학습한 Python 언어를 대상으로 프로그램 이해하기와 작성하기 문항들을 가지고 2회 문제해결력을 측정하였다. <표 3>과 같이 첫번째 문항군(<문제해결력 측정문항군 1>)에서는 변수와 수식, 선택, 반복의 프로그래밍 요소를 이용하여 터틀 기반의 GUI 프로그램을 이해하고 코딩하는 문제들이었고, 두번째 문항군(<문제해결력 측정문항군 2>)에서는 첫번째 문항군의 프로그래밍 요소를 포함하고 함수와 객체를 포함하여 종합적으로 프로그램을 이해하고

코딩을 하는 문제를 활용했다. 이문제들도 마찬가지로 GUI 프로그래밍에 관한 문제들이었다.

따라서 <문제해결력 측정문항군 1>보다 <문제해결력 측정문항군 2>가 많은 프로그래밍 요소들을 포함하고 있고, 종합적인 코딩 기반의 문제해결력을 요구하여 난이도가 높은 것으로 평가할 수 있다.

## 4. 연구결과

### 4.1 시각적 문해력 요인에 대한 요인분석

이번 장에서는 시각적 문해력이 프로그램을 이해하고 작성하는 코딩 기반의 문제해결력에 어떤 영향을 미치는지를 분석한 후, 시각적 문해력 수준과 추상적/구체적 사고수준을 상하로 두어, 해당 문제해결력에 어떤 영향을 미치는지 살펴보았다. 이를 위해 본 연구에서 시각적 문해력을 측정하기 위한 ACRL의 7개 범주의 20개 항목에 대하여 신뢰성 분석과 요인분석을 하였다.

시각적 문해력의 측정항목들을 다차원 요인으로서 축소하여 시각적 문해력 요인들이 시각적 프로그램을 이해하고 작성하는 코딩 기반의 문제해결력에 어떤 영향을 미치는지에 대해서 분석하였다. 요인분석 결과는 <표 4>에 나타난 바와 같이 시각적 문해력을 구성하는 요인

표 4. 시각적 문해력 구성개념에 대한 요인분석

	성분		
	1 (분석과 사용)	2 (제작)	3 (평가)
■ 이미지 의미와 관련 정보 식별함	.84	.08	.22
■ 이미지를 정확하게 찾아냄	.74	.31	-.04
■ 이미지 출처 정보를 잘 수집함	.70	.31	.29
■ 이미지를 맥락에서 이해함	.66	.10	.41
■ 시각적 매체를 프레젠테이션에 잘 인용함	.65	.43	-.002
■ 이미지를 효과적으로 사용함	.63	.47	.10
■ 이미지를 효율적으로 검색	.62	.40	.18
■ 다양한 시각적 자료 제작함	.33	.82	.20
■ 디자인 전략과 창의력을 사용함	.33	.82	.19
■ 문제해결력과 창의력을 사용	.33	.81	.10
■ 다양한 도구와 기술을 사용하여 제작함	.29	.76	.25
■ 기술을 효과적으로 사용하여 이미지 작업을 함	.08	.74	.45
■ 이미지의 해석을 다른 학생과 대화로 검증	.03	.12	.87
■ 이미지 필요성을 말함	.41	.25	.57
■ 이미지의 효과를 잘 평가함	.17	.28	.54

요인추출 방법: 주성분 분석.  
회전 방법: Kaiser 정규화가 있는 베리맥스.

들은 분석과 사용 능력, 제작 능력, 평가 능력으로 나타났다.

신뢰성 분석을 위해 측정항목에 대한 크롬바흐  $\alpha$  값은 분석과 사용 능력 요인이 .89, 제작 능력 요인이 .92, 평가 능력 요인이 .64이었다. 세 가지 요인은 모두 크롬바흐  $\alpha$  값들이 .6보다 커서 내적 일관성이 있는 것으로 나타났다 [32][33].

### 4.2 시각적 문해력, 사고 유형, 코딩 기반 문제해결력 간의 상관관계 분석

다음 시각적 문해력 값과 사고 유형(BIF 값), 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력 간의 상관관계 분석을 통해 관계를 살펴보았다. <표 5>와 같이 상관관계 분석에서는 시각적 문해력이 높은 학생들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높은 것으로 나타났다. 반면, 사고 유형은 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 영향을 미치지 않는 것으로 나타났다.

표 5. 학습자의 시각적 문해력과 사고 유형, 코딩 기반 문제해결력 간의 상관관계 분석

요인		<문제해결력 측정문항군 1>	<문제해결력 측정문항군 2>
시각적 문해력 값	Pearson 상관계수	.38**	.36**
	유의확률 (양쪽)	.002	.004
BIF 값	Pearson 상관계수	-.09	.04
	유의확률 (양쪽)	.48	.78

\*\* 상관계수는 0.01 수준(양쪽)에서 유의

### 4.3 시각적 문해력과 코딩 기반 문제해결력 간의 회귀분석

다음은 시각적 문해력의 요인들을 독립 변수로 삼고 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력을 종속 변수로 삼아 회귀분석을 하였다. 그 결과, <표 6>에서와 같이 <문제해결력 측정문항군 1>에 대해서는 시각적 문해력 요인 중에서 분석과 사용 능력이 회귀계수 1.13( $p = .003$ )으로서 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 유의한 영향을 미치는 것으로 나타났다. 시각적 문해력의 다른 요인들은 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 유의한 영향을 미치지 않았다. 즉, 난이도가 상

**표 6.** 시각적 문해력 요인과 <문제해결력 측정문항군 1> 간 회귀분석 결과

요인	비표준화 계수		표준화 계수	t	p
	회귀 계수	표준 오차	베타		
분석과 사용 능력	1.13	.36	.37	3.10	.003
제작 능력	.70	.36	.23	1.91	.061
평가 능력	.21	.36	.07	.58	.566

\* 종속 변수: <문제해결력 측정문항군 1>, 자유도 = 3,  $F = 4.62$ ,  $p = .006$

**표 7.** 시각적 문해력 요인과 <문제해결력 측정문항군 2> 간 회귀분석 결과

모형	비표준화 계수		표준화 계수	t	p
	회귀 계수	표준 오차	베타		
분석과 사용 능력	.902	.402	.273	2.240	.029
제작 능력	.874	.402	.265	2.172	.034
평가 능력	.261	.402	.079	.647	.520

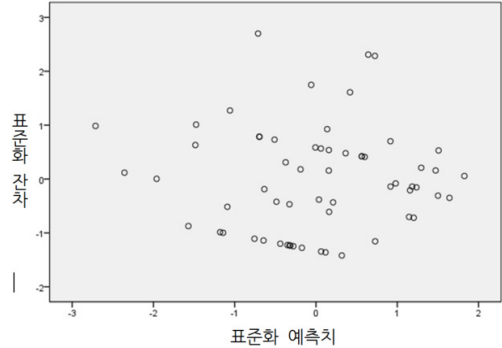
\* 종속 변수 : <문제해결력 측정문항군 2>, 자유도 = 3,  $F = 3.40$ ,  $p = 0.024$

대적으로 낮은 문항군에서는 시각적 문해력 요인에서 시각적 자료를 잘 검색하고 수집하며, 효과적으로 사용하는 능력이 도움을 주고 있음을 알 수 있었다.

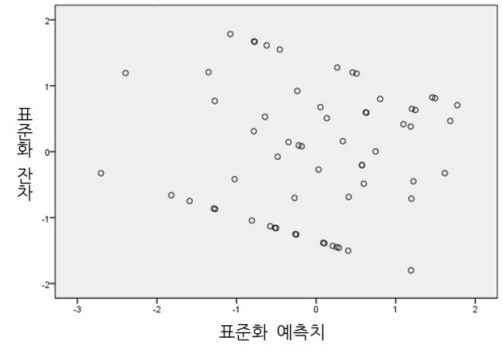
<문제해결력 측정문항군 2>에 대해서는 <표 7>과 같이 시각적 문해력 요인 중에서 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 대해 분석과 사용 능력이 회귀계수 0.902( $p = .029$ )로서 유의한 영향을 미치는 것으로 나타났다. 제작 능력이 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 미치는 영향은 회귀계수 0.874( $p = .034$ )로서 유의한 영향을 미치는 것으로 나타났다. 즉, 종합적인 프로그래밍 요소를 활용하여 문제를 잘 해결하기 위해서는 시각적 문해력 요인 중에서 분석과 사용 능력과 더불어 창의적으로 설계하고 제작해보는 능력이 함께 필요함을 알 수 있었다.

하지만, <문제해결력 측정문항군 1>과 <문제해결력 측정문항군 2>에서 평가 능력은 모두 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 유의한 영향을 미치지 못하였다. <문제해결력 측정문항군 1>과 <문제해결력 측정문항군 2>는 프로그램을 이해하고 프로그램을 작성하는 활동이 핵심으로, 평가와 관련된 활동 요소를 포함하고 있지 않다.

다음은 샘플의 수가 작아 회귀분석 결과가 회귀분석의 기본 가정을 위배하는지를 확인하기 위해 등분산성(homoscedasticity), 정규성(normality), 오차항의 독립성



(가) <문제해결력 측정문항군 1>



(나) <문제해결력 측정문항군 2>

[그림 1] 등분산성 분석

**표 8.** 정규성 분석

	평균		표준편차	
	A	B	A	B
예측값	3.47	4.95	.76	1.39
잔차	.00	.00	2.55	3.57
표준화된 예측값	.00	.00	1.00	1.00
표준화된 잔차	.00	.00	.98	.98

종속 변수: A:<문제해결력 측정문항군 1>, B:<문제해결력 측정문항군 2>

(independence of error)을 분석하였다. 첫째, 다음 [그림 1]과 같이 독립 변수값의 수준이 다를 때 종속 변수가 가질 수 있는 평균은 달라도 평균을 중심으로 분산은 동일한 분포를 보인다[34].

둘째, 회귀분석의 정규성에서 오차항의 평균은 0이며, 분산은  $\sigma^2$ 인 정규분포를 이루어야 한다. 다음 <표 8>에서와 같이 표준화된 오차항의 평균은 0이며, 분산은 0.975로 나타나고 있어, 오차항은 정규분포를 따르고 있다[34].

셋째, 오차항 간의 독립성을 분석하기 위해 상관관계를 분석하였다. 오차항 간의 상관관계가 존재한다는 것은 종속 변수 간에 체계적인 상호관계가 존재한다는 것



을 의미한다. 이는 종속 변수를 추정할 때 발생하는 오차가 추정점에서만 독립적으로 발생하는 것이 아님을 의미한다. 오차항 간의 상관관계는 Durbin-Watson test로 검증할 수 있다. <문제해결력 측정문항군 1>의 Durbin-Watson 값이 1.48보다 크면서 2에 가까운 1.7로 나왔기에 오차항의 자기 상관은 있지 않다[34]. 또한, <문제해결력 측정문항군 2>의 Durbin-Watson 값도 1.48보다 큰 값인 1.523으로 나왔기에 오차항의 자기 상관은 있지 않다[34].

넷째, 독립 변수 간의 다중공선성(multi-collinearity)이 없음을 확인하였다. 독립 변수 간의 상관관계가 높으면 특정변수의 순수한 효과가 작아지게 되어 특정변수의 유의성이 상실될 수 있다. 다음 <표 9>와 같이 공차 한계(Tolerance)는 1.0으로서 공차 한계의 최댓값을 보였다. 분산팽창요인(Variance Inflation Factor, VIF)은 1.0으로서 10보다 작았으므로 독립 변수 간의 다중공선성 문제가 존재하지 않는다.

표 9. 독립 변수 간 다중공선성 분석

독립 변수	비표준화 계수		표준화 계수	t	유의 확률
	회귀계수	표준 오차	베타		
제작능력	1.06	.47	.28	2.24	.029
사용능력	.82	.47	.21	1.74	.089
평가능력	.36	.47	.09	.76	.451

(가) <문제해결력 측정문항군 1>

독립 변수	비표준화 계수		표준화 계수	t	유의 확률
	회귀계수	표준 오차	베타		
제작능력	1.06	.47	.28	2.24	.029
사용능력	.82	.47	.21	1.73	.089
평가능력	.36	.47	.09	.76	.451

(나) <문제해결력 측정문항군 2>

#### 4.4 사고 유형과 시각적 문해력 간 상호작용 효과 분석

다음은 사고 유형을 나타내는 BIF 값과 시각적 문해력 값의 평균을 구한 후, 평균 이하인 집단과 평균을 초과하는 집단으로 구분하였다. BIF 값과 시각적 문해력 값의 평균을 기준으로 나는 집단 간 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 차이가 있는지 분산분석을 하였다. 아울러 분산분석에서 상호작용 관계가 있는지를 확인하여 학생들의 개인 특성 간 관계를 분석하였다. 분산분석에서 상호작용 효과 분석은 어떤 한 독립 변수의 효과가 다른 독립 변수의 수준에 영향을 받는지 아닌지를 조사하는 것이다.

BIF 값의 평균은 12.9였고, 시각적 문해력 값의 평균은 69.2였다. 이에 따라서, BIF 값의 경우, 0~13점은 하위집단, 14~25점은 상위집단으로 구분하였으며, 시각적 문해력의 경우, 0~70점은 하위집단, 71~100점은 상위집단으로 구분하였다. 수준 간 학생들의 수는 다음 <표 10>과 같다.

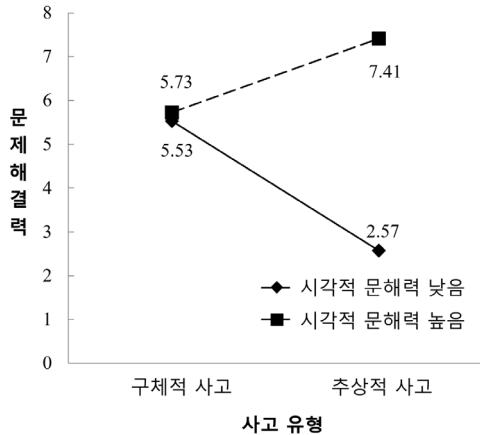
표 10. 집단 간 빈도수

항목	변숫값 설명	N
사고 유형(BLEVEL)	구체적 사고	32
	추상적 사고	29
시각적 문해력 수준(VLEVEL)	하위수준	33
	상위수준	28

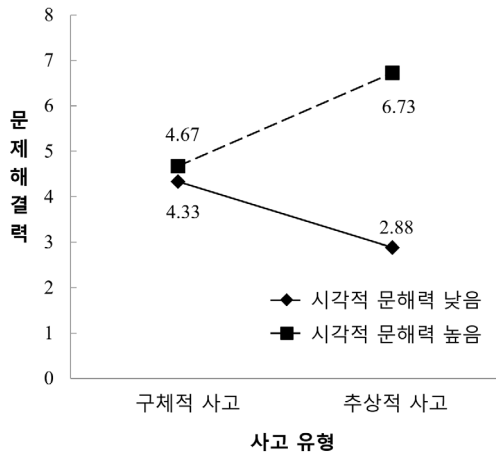
<문제해결력 측정문항군 1>과 <문제해결력 측정문항군 2>에 대한 학생들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력 점수를 사고 유형과 시각적 문해력 수준으로 구분하여 그래프를 그리면 [그림 2]와 같다. [그림 2]를 바탕으로 <문제해결력 측정문항군 1>과 <문제해결력 측정문항군 2>에 대해 사고 유형(BLEVEL)과 시각적 문해력 수준(VLEVEL) 간 상호작용 효과가 있는지 분산분석을 하였다.

<문제해결력 측정문항군 1>에 대한 평균 차이검증을 위해 사고 유형(추상적 vs. 구체적)과 시각적 문해력 수준(낮음 vs. 높음) 조건에 따른 2원 변량 분산분석을 하였다. 먼저 사고 유형과 시각적 문해력 수준 조건에 따른 집단별 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력 평균(표준편차)은 다음과 같다. 구체적 사고인 경우의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 평균은 5.62( $SD=2.68$ )이며, 추상적 사고인 경우의 문제해결력 평균은 4.91( $SD=3.44$ )이었다. 낮은 시각적 문해력인 경우의 문제해결력 평균은 4.18( $SD=2.99$ )이며, 높은 시각적 문해력인 경우의 문제해결력 평균은 6.57( $SD=2.66$ )이었다. 분산분석 결과 사고 유형에 따른 주 효과는 통계적으로 유의하지 않았다( $F(1,57)=.90, p=.346$ ). 분산분석 결과 시각적 문해력 수준에 따른 주 효과는 통계적으로 유의하였다( $F(1,57)=14.20, p<.001$ ).

마지막으로 사고 유형과 시각적 문해력의 상호작용 효과는 통계적으로 유의하였다( $F(1,57)=12.09, p=.001$ ). 즉, 추상적 사고수준이 낮은 경우에는 시각적 문해력 수준이 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 미치는 차이가 없었지만, 추상적 사고수준이 높은 경우에는 시각적 문해력이 높을수록 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높은 것으로 나타났다.



(가) <문제해결력 측정문항군 1>



(나) <문제해결력 측정문항군 2>

[그림 2] 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 대한 분산분석 결과

<문제해결력 측정문항군 1>과 마찬가지로 <문제해결력 측정문항군 2>에 대한 평균 차이검증을 위해 사고 유형(추상적 vs. 구체적)과 시각적 문해력 수준(낮음 vs. 높음) 조건에 따른 2원 변량 분산분석을 하였다. 구체적 사고인 경우의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 평균은 4.47( $SD=3.28$ )이며, 추상적 사고인 경우의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 평균은 4.74( $SD=3.37$ )이었다. 낮은 시각적 문해력인 경우의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 평균은 3.67( $SD=3.19$ )이며, 높은 시각적 문해력인 경우, 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력의 평균은 5.70( $SD=3.13$ )이었다.

분산분석 결과 사고 유형에 따른 주 효과는 통계적으로 유의하지 않았다( $F(1,57)=.15, p=.698$ ). 분산분석 결과

시각적 문해력 수준에 따른 주 효과는 통계적으로 유의하였다( $F(1,57)=6.94, p=.011$ ). 마지막으로 사고 유형과 시각적 문해력의 상호작용 효과는 통계적으로 유의하였다( $F(1,57)=4.90, p=.031$ ).

즉, 추상적 사고수준이 낮은 경우에는 시각적 문해력 수준이 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 미치는 차이가 없었지만, 추상적 사고수준이 높은 경우에는 시각적 문해력이 높을수록 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높은 것으로 나타났다.

분석 결과 <문제해결력 측정문항군 1>과 <문제해결력 측정문항군 2> 모두 <표 11>과 같이 사고 유형에 대해서는 수준 간 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력 격차가 존재하지 않았지만, 시각적 문해력 수준이 상위인 학생들이 두 시험 모두 성적이 좋았으며, 두 요인 간에도 상호작용이 존재하는 것으로 나타났다.

본 연구의 실험 결과로부터 다음과 같은 사실을 알 수 있다. 시각적 문해력 수준이 상위이면 사고 유형이 상위인 학생들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높았고 사고 유형 간 격차가 매우 컸다. 하지만, 시각적 문해력 수준이 하위이면 사고 유형이 하위인 학생들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높았다. 즉, 시각적 문해력이 높으면, 추상적 사고를 하는 학생일수록 문제를 잘 해결하는 반면, 시각적 문해력이 낮으면, 구체적 사고를 하는 학생일수록 문제를 잘 해결하는 것으로 분석되었다.

다른 각도에서 설명하면, 추상적인 사고 유형의 학습자들에 대해서 시각적 문해력 수준 간 격차가 컸고, 구체적인 사고를 하는 학습자들에 대해서는 시각적 문해력 수준 간 격차가 거의 없었다. 결국, 추상적인 사고를 하는 학습자들이 높은 수준의 시각적 문해력을 갖게 된다면, 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력에 도움을 받을 수 있다.

시각적 문해력은 현대에 와서 사람들이 시각 정보에 점점 더 많이 노출되면서 중요성이 강조되고 있다[35]. 지금까지 시각적 학습에 관한 연구는 교양이나 그림책 교육 등 미술 교육을 중심으로 이루어져 왔다. 그러나 코딩 교육과 같은 다른 영역에서 시행된 연구는 매우 드물다.

본 논문은 시각적 문해력이 컴퓨터 과학 교육 중 하나인 코딩 교육에 어떻게 영향을 미치는지를 분석하였다. 시각적 문해력을 고려하지 않았던 이전 연구[21][22]에서는 프로그래밍 초보자의 경우에는 구체적인 사고 유형을 가진 학습자가 더 유리하였다. 본 연구에서도 시각적 문해력이 낮은 집단의 경우에는 구체적인 사고를 하는 학



표 11. 분산분석 결과

원천	제곱합	자유도	평균 제곱	F값	유의 확률
BLEVEL	6.10	1	6.10	.90	.346
VLEVEL	95.78	1	95.78	14.20	.000
BLEVEL * VLEVEL	81.56	1	81.56	12.09	.001
오차	384.48	57	6.75		
합계	2262.58	61			
수정 합계	562.31	60			

\* R<sup>2</sup> = .32 (수정된 R<sup>2</sup> = .28)

(가) <문제해결력 측정문항군 1>에 대한 분산분석

원천	제곱합	자유도	평균 제곱	F값	유의 확률
BLEVEL	1.45	1	1.45	.15	.698
VLEVEL	66.15	1	66.15	6.94	.011
BLEVEL * VLEVEL	46.64	1	46.64	4.90	.031
오차	543.04	57	9.53		
합계	1941.81	61			
수정 합계	652.43	60			

\* R<sup>2</sup> = .17 (수정된 R<sup>2</sup> = .12)

(나) <문제해결력 측정문항군 2>에 대한 분산분석

습자들의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 높았다. 하지만, <표 11>에서와 같이 시각적 문해력 요인과 사고 유형을 함께 고려하였을 때, 두 요인이 모두 상위인 집단의 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력이 가장 높았음을 알 수 있다.

본 연구에서 분산분석의 분석 대상이었던 학생들의 지능지수, 학업성적, 소프트웨어 전문성과 경험, 부모의 소득 등 많은 개인적 차이는 전체 평균에서 설명되지 않는 오차항으로 반영이 되었다. 그런데도 분산분석에 사용된 집단 간 평균 차이는 통계적으로 유의한 차이를 보였다. 본 연구에서 사용된 분산분석의 절차와 측정방법과 실험 설계는 기존의 연구방법론 교재와 논문에서 사용된 방법을 그대로 따르고 있다[36][37][38].

## 5. 제언 및 결론

최근 소프트웨어 교육이 강조되면서 초·중등학교를 비롯하여 대학의 교양 과목에서 Scratch 언어를 비롯하여 AppInventor, Entry [39] 등 시각적 프로그래밍 환경을 갖는 블록 기반의 언어들이 많이 사용되고 있다. 코딩 학습 현장에서 학습자들이 문제를 해결하기 위해 코딩을 좀 더 잘 할 수 있게 도와주기 위한 교수학습 방법이 필요해졌

다. 이를 위해 학습자들의 특성을 먼저 파악해 볼 필요가 있다. 어떤 학습자 특성이 코딩에 긍정적인 영향을 미치는지 찾아 적절한 교수학습 방법을 마련해야 할 것이다.

본 논문에서는 시각적 정보를 찾는 요즘 Z세대들의 특성을 고려하여 시각적 문해력에 초점을 두고 분석하였다. 한 연구에서 시각적으로 생각할 수 있는 능력은 시각적 메시지를 정확하게 해석하고 그러한 메시지를 작성할 수 있는 능력이라 정의하였다[40]. 교육과정에서 시각적 교육 자료의 활용으로 학생들이 자료를 더 잘 활용하고, 고차원적 사고 능력을 계발시킬 수 있으며 학생들이 영상을 명확하게 보고 개념화 할 수 있는 기초적인 능력을 갖추게 해준다고 보고하고 있다[35][41].

시각적 문해력과 사고 유형을 함께 고려한 이번 연구를 통해 기존 연구들의 문제점을 해결할 수 있었다. 즉, 사고 유형과 코딩 능력 간의 관계 분석을 한 기존의 연구들[21][22]에서 어떤 경우에는 추상적 사고수준이 높은 것이 코딩 능력과 양의 상관관계를 가지고 있고, 또 다른 경우에는 구체적 사고수준이 높은 것이 코딩 능력과 양의 상관관계를 가지는 것으로 나타났다. 성별 간의 격차가 있음은 파악하였으나 더 정확한 다른 요인을 찾지 못하였다.

본 연구결과에서는 기존 연구에서 밝히지 못한 원인을 시각적 문해력으로 규명하였다. 즉, 시각적 문해력과 사고 유형 간에 상호작용이 존재하고 있음을 밝힘으로써 어떤 경우에 추상적 사고와 구체적 사고가 각각 코딩 능력에 도움이 되는지 밝힐 수 있었다. 시각적 문해력이 높은 경우, 추상적 사고수준이 함께 높다면 가장 높은 시각적 프로그램을 이해하고 작성하는 코딩 기반 문제해결력을 가지게 됨을 알게 되었고, 시각적 문해력이 낮은 경우, 구체적 사고가 코딩 능력에 도움이 됨을 알 수 있다.

한편, 본 연구에서 시각적 문해력 요인으로 추출한 분석과 사용 요인, 제작 요인, 평가 요인은 디자인 프로세스를 설명하기 위한 Lohr(2008)의 ACE(analyze, create, and evaluate) 모델[42]과 유사점이 있다. 즉, 분석(analyze)에서는 해결한 문제의 목적과 방법을 생각하고, 제작(create)에서는 시각적인 발상을 하고 제작을 위한 도구를 사용하며, 평가(evaluate)에서는 결과물의 효과와 효율성을 함께 평가하는 것이다. 시각적 문해력을 높이기 한 기존의 한 연구에 따르면, 비전공자 소프트웨어 프로그래밍 교육과정의 콘텐츠에서 학습 목표를 3단계로 두고 있다. 우선, 실세계의 문제를 컴퓨팅 문제로 바꾸고, 소프트웨어 해결책으로 시각화한 후, 협업과 대화를 하면서 코딩을 하고 문제를 해결한다[7][8]. 시각적 문해력 증진을 위해 이와 같은 수업 프로세스에 주목할 필요가 있다.

한편, 시각적 문해력 이외에 교수자들은 일반적으로 학생마다 다양한 학습 스타일로 학습한다고 주장한다[40]. 따라서 수업 내용은 다른 방식으로 학생들에게 전달되어야 하며 여러 가지 수업 방법이 활용되어야 효과적이라고 주장한다[40]. 학생들이 학습할 때에 학생들의 사고 유형도 중요한 역할을 한다. 대부분 초중등 학생의 경우에는 구체적인 사고를 하다가 성인이 되면서 성장과정 중의 경험을 바탕으로 추상적 사고를 한다고 주장한다[43].

하지만, 경험만으로는 학생들이 구체적 사고를 하다가 모두 추상적 사고를 하는 것은 아니라고도 주장한다[43]. 기존의 한 연구에서 추상적 사고를 유도하기 위해 시각장애인의 경우를 고려하여 방법을 제안한 바 있다. 연구에 따르면, 구체적 사고에서 추상적 사고로 이동하는 데 도움이 되는 몇 가지 중요한 요소는 다음과 같다[43]. 첫째, 추상적 사고수준을 높이기 위해 학생들을 문제 해결 상황에 참여시키고, 합리적인 결론에 이르기까지 일련의 작업을 계획하며 순서를 정하게 한다. 둘째, 문제에 대한 가능한 해결책을 식별하고 평가하게 한다[43].

기존의 연구가 모든 학습자를 대상으로 한 일반화된 연구는 아니었다. 하지만, 컴퓨터 비전공자인 학생들의 코딩 능력을 증진시키기 위해 학생들이 구체적으로 생각하도록 도울 필요가 있으며, 일련의 작업을 계획하고 순서를 정하여 문제를 해결할 수 있도록 돕는 것이 유익하다.

### 참고문헌

[ 1 ] 교육부 (2015). 2015 개정 교육과정 보도자료. <http://www.moe.go.kr/web/100063/ko/board/view.do?bbsId=316&boardSeq=62381&mode=view>에서 인용.

[ 2 ] 국가교육과정정보센터 (2015). 교육과정 원문 및 해설서 2015 개정시기. <http://ncic.go.kr/mobile.dwn.ogf.inventoryList.do#>에서 인용.

[ 3 ] 손병국·송상호 (2018). 초등학교 소프트웨어 교육에서 에듀테인먼트 특성에 대한 주관성 유형 탐색. **정보교육학회논문지**, 22(2), 177-193.

[ 4 ] 주현재·이현민·최성경 (2019). 4Cs 역량 증진을 위한 코딩 교양 교과목 개발 - S 전문대학 교양 교육을 중심으로-. **문화와융합**, 41(6), 817-846.

[ 5 ] 오소희·김응진·김성식 (2017). 메타인지 향상을 위한 엔트리 프로그래밍 교육 콘텐츠 개발 및 적용. **컴퓨터교육학회논문지**, 20(5), 61-68.

[ 6 ] 소프트웨어중심대학. <http://www.software.kr/um/um03/um0303/um030302/um03030203.do>에서 인용.

[ 7 ] 피수영 (2016). IT 융합교육을 위한 비전공자 코딩교육의 발전방안. **디지털융복합연구**, 14(10), 1-8.

[ 8 ] 이민정 (2017). 비전공자 SW 프로그래밍 교육과정 및 콘텐츠 개발 모형의 효과성 탐색: SW 해결안의 시각적 표현을 중심으로. **디지털콘텐츠학회논문지**, 18(7), 1313-1321.

[ 9 ] Scratch. <https://scratch.mit.edu/>에서 인용.

[ 10 ] AppInventor. <http://ai2.appinventor.mit.edu/>에서 인용.

[ 11 ] Kim, K., & Kim, H. (2014). A case study on necessity of computer programming for interdisciplinary education. *Journal of digital convergence*, 12(11), 339-348.

[ 12 ] Python Software Foundation. (2017). Python about. <https://www.python.org/about/>에서 인용.

[ 13 ] Guo, P. (2014). Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. *BLOG@CACM*, <http://cacm.acm.org/blogs/>에서 인용.

[ 14 ] <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>에서 인용.

[ 15 ] 이영석 (2018). 컴퓨터 비전공자를 위한 파이썬 기반 소프트웨어 교육 모델. **한국융합학회논문지**, 9(3), 73-78.

[ 16 ] 윤인성 (2019). 혼자 공부하는 파이썬. 한빛미디어.

[ 17 ] 원유현 (2011). 프로그래밍 언어 개념. 정익사.

[ 18 ] Koschke, R. (2003). Software Visualization in Software Maintenance, Reverse Engineering, and Re-engineering: A Research Survey. *Journal of Software: Evolution and Process*, 15(2), 87-109.

[ 19 ] Brumberger, E. (2011). Visual Literacy and the Digital Native: An Examination of the Millennial Learner. *Journal of Visual Literacy*, 30(1), 19-47.

[ 20 ] Moody, D. L. (2010). The "Physics" of Notations: A Scientific Approach to Designing Visual Notations in Software Engineering. *2010 ACM/IEEE 32nd International Conference on Software Engineering*, 2, 485-486.

[ 21 ] 박찬정·현정석 (2018). 컴퓨팅 사고력을 위한 시각적 다이어그램 구조화의 성별 및 추상적 사고 성향 차이 분석. **컴퓨터교육학회논문지**, 21(3), 11-20.

[ 22 ] 박찬정·현정석·진희란·정혜선 (2015). C와 스크래치 프로그램 이해에 대한 고등학생의 추상적 사고, 언어 친밀성, 학업성취도 관계 분석. **컴퓨터교육학회논문지**, 18(6), 1-12.

[ 23 ] 신지형 (2019). 밀레니얼세대와 Z세대의 미디어 이용. **KISDI STAT Report**, 19(4), 2-7.

[ 24 ] Hattwig, D., Bussert, K., Medaille, A., & Burgess, J. (2013). Visual literacy standards in higher education: New opportunities for libraries and student learning. *Libraries and the Academy*, 13(1), 61-89.

[ 25 ] 강인애·이재경·남미진 (2014). 융합인재교육(STEAM)을 통한 시각적 문해력 효과 연구: 인포그래픽 동영상 만들기를 중심으로. **교육발전연구**, 30(1), 89-110.

[ 26 ] Kiper, A., Arslan, S., Kıyıcı, M., & Akgün, Ö. E. (2012). Visual Literacy Scale: The Study of Validity and

Reliability. *The Online Journal of New Horizons in Education*, 2(2), 73-83.

[27] Pem, K. (2019). *Enhancing High Order Science Visual Literacy Skills in Biology Learners*. Open University of Mauritius. Online Submission.

[28] Vallacher, R. R. & Wegner, D. M. (1987). What Do People Think They're Doing? Action Identification and Human Behavior. *Psychological Review*, 94(1), 3-15.

[29] Vallacher, R. R. & Wegner, D. M. (1989). Levels of Personal Agency: Individual Variation in Action Identification. *Journal of Personality and Social Psychology*, 57(4), 660-671.

[30] Trope, Y. & Liberman, N. (2003). Temporal Construal. *Psychological Review*, 110(3), 403-421.

[31] Trope, Y. & Liberman, N. (2010). Construal-Level Theory of Psychological Distance. *Psychological Review*, 117(2), 440-463.

[32] George, D. & Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference. 11.0 update. (4th ed)*, Allyn & Bacon, Boston.

[33] 이한식·김영 (2002). **SPSS 10.0 매뉴얼**. 법문사.

[34] 남준우·이한식 (2010). **계량경제학 이론과 EViews/Excel 활용 (제3판)**. 홍문사.

[35] Daniels, S. (2019). Why Visual Learning and Teaching? <https://www.insightresources.org/2019/04/26/why-visual-learning-and-teaching/>에서 인용. (*Visual Learning and Teaching: An Essential Guide for Educators K-8*에서 발췌).

[36] 백순근 (2007). **교육연구 및 통계작성**. 교육과학사

[37] 임시혁 (2017). **통계자료의 분석의 이해와 적용**. 아카데미프레스.

[38] 김재휘·김태훈·박인희 (2010). 예방 행동의 결과를 얻는 시점에 따른 효과적인 설득 메시지 유형: 해석수준이론을 중심으로. **한국심리학회지:소비자 광고**, 11(3), 451-474.

[39] Entry. <https://playentry.org/#/>에서 인용.

[40] <https://www.thecampbellinstitute.org/wp-content/uploads/2017/09/Campbell-Institute-Visual-Literacy-WP.pdf>에서 인용.

[41] 박소라·김정선 (2011). 시각적 문해력 향상을 위한 그림책의 그림 읽기 전략 연구. **미술교육연구논총**, 30, 27-53.

[42] Lohr, L. L. (2007). *Creating graphics for learning and performance: Lessons in visual literacy*. Prentice Hall Press.

[43] <https://www.pathstoliteracy.org/blog/recognizing-and-building-abstract-thinking-child-visual-impairment#>에서 인용.

### 박 찬 정



1988년 서강대학교  
전자계산학과(공학사)  
1990년 KAIST 전산학과(공학석사)  
1998년 서강대학교 대학원 전자  
계산학과(공학박사)  
1990년~1994년 한국통신  
소프트웨어연구소 전임연구원  
1998년~1999년 한국통신 멀티미디어연구소 전임연구원  
1999년~현재 제주대학교 컴퓨터교육과 교수  
관심분야: 정보·컴퓨터 융합 교육, 코딩 교육, 에듀테크, 데이터  
마이닝

E-Mail: [cjpark@jejunu.ac.kr](mailto:cjpark@jejunu.ac.kr)

### 현 정 석



1991년 서강대학교 경영학과  
(경영학사)  
1993년 서강대학교 대학원 경영학과  
(경영학석사)  
1998년 서강대학교 대학원 경영학과  
(경영학박사)  
2002년~현재 제주대학교 경영정보학과  
교수

2008년 제주대학교 대학을 빛낸 교수상 수상

2012년 특허청장상 수상

2016년 제주대학교 강의평가 최우수 교수상 수상

관심분야: 마케팅, 행동의사결정론, 트리츠, 창의성 교육, 영재교육

E-Mail: [jshyun@jejunu.ac.kr](mailto:jshyun@jejunu.ac.kr)