

## 루브릭의 인지여부가 코딩교육의 학습효과에 미치는 영향

이 애 리\* · 김 창 희\*\*

### *The effect of recognition of rubrics on the learning effect of coding education*

Lee Aeri · Kim Changhee

#### 〈Abstract〉

This study provided a rubric prior to the performance of a task in a coding class and aimed to verify its effect. As for the research contents, this study redesigned the rubric developed in the preceding studies for the purpose of this study, presented that to the students and conducted a verification of its effects on the computer efficacy and academic achievement of the students who participated in the experiment, conducting a class. For this purpose, with 41 students in an experimental group, to which the rubric was presented to conduct the class and 39 students in a control group to which the rubric was not presented to conduct the class, this study conducted a six-week class and conducted a survey on their self-efficacy in the computer and a test to measure their academic achievement before and after the experiment. As a result, there were more positive effects on both computer efficacy and academic achievement in the group to which a rubric was presented and in which a rubric evaluation was conducted than in the control group. Through the result of this study, it was noted that the evaluation method presenting more specific and more detailed standards for performance and the level of the standards would have positive impacts on the learners' learning.

Key Words : Coding Education, Rubric, Performance Assessment, Programming

### I. 서론

코딩 교육은 프로그래밍 언어를 통해 사물이나 실생활에서 일어나는 사건이나 문제를 논리적으로 분석해서 문제점을 발견하고 해결하는 과정에서 창의적 사고력, 논리적 사고력, 문제해결력 등을 키우

는 것을 목적으로 한다. 코딩 수업을 진행해보면 코딩을 처음 접하는 학생들은 비교적 쉬운 교육용 언어를 사용하는 수업에서도 중도에 수업을 포기하거나 수업의 어려움을 호소하는 경우가 많이 나타난다[1].

또한 코딩교육은 이론적인 학습보다 실기적인 학습이 더 중요하고 수업시간마다 학생들이 직접 창의적인 산출물을 제작하고 이결과물은 평가의 대상이

\* 가톨릭관동대학교 VERUM교양대학 교양과 조교수

\*\* 서울기독대학교 국제경영정보학과 조교수(교신저자)

된다. 이러한 학습결과물에 대한 평가는 교수자의 주관적인 판단 하에 평가되며, 평가하는 사람에 따라 또는 평가하는 환경에 따라 그 결과가 달라질 수 있어 평가의 신뢰도에 대한 불신이 있을 수 있다. 학습자에게 부여된 점수에 대하여 객관적이고 신뢰할 수 있고 일관성이 있는 평가의 근거를 제시하고 설명할 수 있어야 한다. 루브릭(rubric)은 이러한 점들을 보완하기 위해 최근 학습자의 평가를 위한 부분에서 활발하게 연구되고 있다[2].

루브릭(rubric)의 장점은 교수자 입장에서는 학생들이 수행해야 할 과제에 대해 정확하게 평가 준거를 제시하고 각 수행 수준에 대한 지표를 상세하게 제시하여 학생이 수행한 과제의 결과물을 보다 객관적으로 평가할 수 있는 점다[3].

루브릭(rubric)은 학습자의 입장에서는 학생들이 해야 하는 활동에 대한 구체적인 설명이 있기 때문에 수행해야 할 과제에 대한 명확한 단서를 제공 해 주며, 학습자가 수행해야 할 과제에 대해 객관적이며 결과물을 예측할 수 있는 힌트를 부여하므로 과제 수행의 기대 수준을 높일 수 있다. 이를 통해 학습자 자신의 학습 수준을 자각함으로써 학생들에게 자기 평가의 역할을 하게 도와준다[4, 5].

따라서 평가 루브릭(rubric)을 코딩 수업의 과제 수행 전에 제시한다면 코딩 수업에서 어려움을 겪는 학생들에게 수행할 과제에 대한 단서를 제시할 수 있고 학습 결과물의 기대 수준 또한 높아져 학습에 도움을 줄 수 있을 것이다. 또한 학습결과물에 대한 객관적이고 일관적인 평가를 제공 할 수 있을 것이다.

본 연구는 효과적인 평가체계로서 루브릭이 컴퓨터 비전공자를 대상으로 실시되는 교양 코딩 수업의 효과에 어떠한 영향을 미치는지 고찰 하고자 한다. 이를 위해 미리 설계된 루브릭을 수업 전에 학생들에게 제시하고 과제 수행과 평가에 적용하여 학

습자들의 컴퓨터 자기효능감과 학업 성취도에 어떤 영향을 미치는지 분석하였다.

## II. 관련연구

### 2.1 대학에서의 교양 코딩 교육

컴퓨터 정규교육은 1990년에 시작되어 2008년도에 완전 폐지된 후 특수 고등학교이외의 일반 고등학교에서는 선택적으로 '정보'교과를 채택하여 시행해왔으나 2015년 7월 미래창조과학부와 교육부는 초·중등을 대상으로 2018년부터 코딩 교육 필수화 계획을 발표하였다. 이를 위해 초·중등교육에서는 현재 약 1200여개의 소프트웨어 선도학교가 선정되어 운영 중이며, 학교 현장에서 근무하는 교사나 교육전문가들이 소프트웨어 중심의 교육 강화를 위한 교육과정 수립 및 지원체계를 마련하는 등 다양한 활동을 하고 있다[6].

반면 대학교육에서의 코딩 교육은 대학 소프트웨어 교육 혁신을 위해 '소프트웨어 중심대학 사업'에서 2015년 8개, 2016년 6개, 2017년 6개의 소프트웨어 중심 대학이 선정되어 운영 중이다. 이를 시작으로 대학 내 설치된 기초교양교육원등을 활용하여 컴퓨터과학 관련 비전공자 학생들을 대상으로 코딩 교육을 확대하고 있다[7].

대학에서 비전공자 학생들을 대상으로 운영되어 온 컴퓨터 관련 교양 교육과정은 문서작성, 프레젠테이션 등 단순 컴퓨터 응용프로그램 사용법 중심과 IT 관련 자격증의 취득을 목적으로 하는 교육에 한정되었다는 문제점이 지적되면서 대학교육에서 정보기술 유창성 측면에서의 지적 역량이 크게 강조되고 있는 추세이다[8].

현재는 2015년 미래부에서 소프트웨어교육 인재

양성을 목적으로 소프트웨어교육 중심대학 선정되지 않은 대학에서도 비전공자 학생들의 창의적 아이디어를 소프트웨어로 구현하는 사고력 증진을 목적으로 코딩교육이 확대되고 있다. 미래부가 발표한 소프트웨어 중심대학의 운영방안에 따르면 대부분 대학의 코딩 교육은 세가지로 분류할 수 있다. 예비 신입생을 대상으로 운영되고 있는 기초 소프트웨어 프로그램, 비전공자를 대상으로 학기 중에 진행되고 있는 소프트웨어 교육 프로그램, 소프트웨어 교육 프로그램이 발전한 융합 소프트웨어 교육이다. 예비 신입생을 대상으로 실시하는 코딩 교육은 소프트웨어의 기초 교육을 온라인으로 입학 전 예비 신입생에게 제공한다. 학습자는 컴퓨터과학의 관점에서 문제해결을 위한 컴퓨팅적 사고력의 주요 개념을 학습하고 문제해결의 방법론을 학습 할 수 있다. 학기 중 비전공자를 대상으로 실시하는 코딩 교육은 컴퓨팅적 사고력의 주요 핵심개념을 활용하여 학습자의 논리적 사고력의 향상을 목표로 소프트웨어 기초 개념과 교육용 프로그래밍 언어를 활용하여 프로그래밍 실습을 하게 되어 문제해결을 위해 단순히 프로그램을 코딩하는 것뿐만 아니라, 전공과 연계된 주제중심 학습활동으로 문제를 해결하기 위해 자신의 선수 지식과 경험, 사고를 바탕으로 문제를 해결해 나가는 과정을 절차 화시킬 수 있도록 구성된다. 융합 코딩 교육은 소프트웨어 분야의 융합형 인재양성을 목표로 비전공자에게 소프트웨어 전공 교육을 확대할 수 있는 기반을 조성하고, 코딩 교육을 전공에 접목 시킬 수 있는 기회를 제공할 수 있다[7].

본 연구의 강원도 소재 C대학교에서는 2015년까지 컴퓨터 활용 중심의 정보화과목을 교양필수과목으로 선정하여 운영해 왔는데, 2016년도부터 컴퓨터 활용 중심의 교육에서 코딩 교육 중심으로 교육과정을 개편하여 전체 1학년 학생을 대상으로 1년간 총 2학점을 필수 이수학점으로 정하여 운영하고 있다.

## 2.2 루브릭(rubric)

루브릭(rubric)에 대한 개념은 학자마다 다양한 관점에서 정의하고 있으며, 먼저 용어에 대한 의미를 살펴보면 다음과 같다. 루브릭(rubric)은 학생들의 과제 수행 과정과 그 수행의 결과를 분석할 수 있도록 안내해 주고, 과제의 질을 판단할 수 있도록 학생들이 이해하기 쉬운 언어로 준거와 수행 수준을 제시한 준거척도이다[9]. 또는 교수자가 과제를 채점하거나 평정하기 위해 설정된 일련의 준거로써 척도와 산출물이나 수행의 특성에 대한 설명과 다양한 평가 수준의 예로 구성된 채점 기준표 라고 정의하기도 한다[10]. 또한 평가를 위한 채점 리스트라고 할 수 있으며, 일반적으로 각 항목에 해당하는 수준을 측정하고 이를 합산하는 방식으로 사용된다. 루브릭은 평가도구이긴 하지만 주된 사용자는 교수자가 아니라 학습자가 될 수 있으며, 과제 수행과 관련한 평가지표로서 루브릭을 학생들에게 제공하면 결과적으로 학습자들의 과제 수행 능력을 향상시킬 수 있다고 정의하기도 한다[3].

이와 같이 루브릭(rubric)의 정의는 연구하는 사람의 관점에 따라 학습자들이 학습 목표나 수행해야 할 과제의 기대 수준을 정확히 인식하도록 제시해주는 도구로써 정의한 것과 학습자의 산출물의 질을 판단하기 위한 준거로써의 정의한 것으로 크게 두 가지로 구분할 수 있다. 결국 루브릭(rubric)은 학습자에게는 이해하기 쉽고, 명확한 평가의 기준을 바탕으로 가장 높은 성취수준을 도달할 수 있도록 학습동기의 부여가 될 것이고, 채점 결과에 대한 피드백을 제공함으로써 채점 결과에 대한 신뢰성을 갖게 할 것이다. 또한 교수자는 채점에 소요되는 시간과 노력을 줄일 수 있고 공정하고 객관적인 채점을 할 수 있을 것이다[5].

루브릭(rubric)의 요소는 채점영역, 범주의 수, 수

행 판단 기준에 대한 기술이다. 학생의 과제 수행의 결과를 판단하는 기준이 되는 것이 채점영역이다. 이는 과제수행의 결과나 산출물을 어떤 특성이나 영역에 따라 평가할 것인가에 관한 것이다. 각 영역이나 특성을 평정하는 수치나 범주로 구분해야 할 때 평가자는 몇 개의 범주로 구분하여 점수를 부여할 것인가를 결정해야 한다. 범주의 갯수를 정하는 규칙은 없으나 [11]에 의하면 일반적으로 세 개의 범주를 가장 적절하다고 하였다. 각각의 범주에 대한 수행 판단할 기준을 기술해야 한다[12].

### 2.3 컴퓨터 자기 효능감

자기효능감이란 주어진 문제를 해결할 때 자신의 생각 특히 구체적인 자신감을 말한다. 어떤 결과를 얻기 위해 하는 행동을 성공적으로 수행해 낼 수 있는 능력에 대한 개인의 신념이라 할 수 있다. 자기효능감은 자기 자신을 얼마나 유능하고 능력 있는 사람으로 인식하는가를 의미한다. 이런 자신에 대한 신념은 행동에 큰 영향을 주어 자신이 얼마나 그 행동을 잘할 수 있다고 생각하는가에 따라서 행동의 수준이 결정되기 때문에 학습에서는 아주 중요한 요소이다. 자기 효능감에 대해서 높게 지각할수록 그 행동을 보다 자주 선택하고 지속하며 반복하게 되고 그 기대가 높을수록 동기수준이 높아진다[13].

컴퓨터 사용에 대해 자기효능감이 적용된 것을 컴퓨터 자기효능감이라고 한다. 이는 문제를 해결할 때 컴퓨터를 활용하여 잘 해결할 수 있다는 자신감으로 정의할 수 있고 컴퓨터 관련 지식이나 기술을 이용해 과제를 수행할 때 컴퓨터 사용을 조직화 하고 문제를 해결해나갈 수 있는 역량을 가지고 있다고 스스로 믿는 정도라고 할 수 있다[14-16].

Bandura[13]는 개인이 어떤 과제를 쉽다고 생각하면 자기 효능감 기대가 높아진다고 보고 있으며,

능력에 성공을 귀인시킬 때 자기효능감이 높아진다고 보고 있다. 루브릭(rubric)은 학생들이 해야 하는 과제에 대한 구체적인 설명을 제공하므로 수행해야 할 과제에 대한 명확한 단서를 제공하고 결과를 예측할 수 있는 힌트를 부여한다. 따라서 루브릭의 인지여부는 컴퓨터 자기효능감에 영향을 미칠 것이다.

### 2.4 학업 성취도

학업성취도는 학습자가 교육목표를 어느 정도 달성하고 있는가를 다양한 학습과제 및 평가도구들을 이용하여 확인하는 것이다. 학습자가 수업의 일련 과정을 통해서 학습에서 의도한 목적을 이루었는지 이런 수준이 어떠한지를 평가한 결과를 의미한다. 이는 학습자, 교수자, 교육기관이 교육적인 목표를 성취하기 위한 교육의 결과라고 할 수 있다. 학업성취도는 일반적으로 시험이나 계속평가(continuous assessment)등의 방법을 통해 측정할 수 있다.

이에 본 연구에서는 루브릭의 인지여부에 대한 학습효과를 컴퓨터 자기효능감과 학업 성취도 측면에서 분석하였다.

## III. 교과목 운영

### 3.1 교과목 개요

본 연구는 C대학교 1학년 대상의 교양필수 교과목인 '창의적사고와 코딩'을 수강하는 학생들을 대상으로 진행되었다.

본 과목은 교육용 프로그래밍언어를 이용한 코딩 교육을 통해 학생들에게 논리력, 창의력, 문제해결력을 키울 수 있도록 하기 위한 교과목이다. 본 연구에서는 코딩 수업을 위해 초보자도 이해하고 배

우기 쉬운 교육용 프로그래밍 언어인 스크래치를 활용하여 수업을 진행하였다. 스크래치 같은 교육용 언어는 코딩 교양 수업에서 프로그래밍을 처음 접하는 학생들이 코딩 교육에 대해 느끼는 어려움을 최소화하고 코딩 교육에 대한 긍정적 감정을 형성할 수 있도록 하고 이를 통해 본인이 생각하는 바를 어려움 없이 디지털 기술을 통해 표현해 볼 수 있게 하고 자신의 생각을 구체적이고 논리적으로 정리할 수 있으며 이 정리된 것을 구체화한 대상으로 만들 수 있게 한다[1].

### 3.2 교과목 운영

코딩교육의 특성에 맞게 15주 수업을 3개의 모듈로 나누어 조직하였다, 프로그래밍 언어의 기본 문법과 기능을 배우는 2주~7주까지의 기초모듈, 기초 모듈에서 학습한 기초 문법들을 통해 스스로 프로그램을 계획하고 작성해보는 9주~11주까지의 발전모듈, 팀 프로젝트가 이루어지는 12주~14주까지의 심화모듈의 순서로 15주 동안 수업이 진행된다. 실험 집단과 비교집단 모두 기초 모듈과 발전 모듈에 플립드 러닝 수업이 적용 된다. 수업의 방식과 과제는 모두 동일하며 실험 집단에게만 과제 수행 전 루브릭을 제시하고 발전 모듈과 심화 모듈에서 수업을 진행 하였다

전체 수업 15주 수업 중 첫 주와 중간/기말고사를 제외하고 12주의 수업을 기초, 발전, 심화의 세 모듈로 그룹 화하여 각 모듈별 약간 상이한 방식으로 수업을 설계하여 적용하였다. 심화단계는 팀 프로젝트로 운영되기 때문에 플립드 러닝을 적용하지 않았다. 첫 주는 수업에 대한 오리엔테이션과 컴퓨터 효능감 설문을 실시하고, 8주에는 중간고사를 실시하고, 마지막 주에는 동일한 질문의 설문과 사후 진단평가를 실시하였다.

<표 1> 모듈 별 수업 운영

학습 단위	주차	플립드러닝 적용	Rubric 제공	내용
기초	2	○	×	기본적인 프로그래밍 문법
	3	○	×	
	4	○	×	
	5	○	×	
	6	○	×	
	7	○	×	
	발전	9	○	
10		○	○	
11		○	○	
심화	12	×	○	팀 프로젝트
	13	×	○	
	14	×	○	

기초 모듈은 스크래치의 블록과 프로그래밍 언어적인 요소 등 기본 개념에 대해 익힌다. 교수자의 시연을 따라 코딩해 본 후 학생들은 이를 더 확장하여 새로운 결과물을 만들어 내게 된다.

발전 모듈에서는 기초 모듈에서 학습한 기초 문법들을 통해 스스로 프로그램을 계획하고 탐구하고 작성해보는 과정을 거치게 된다. 이때 평가 루브릭이 제시되고 학생 스스로 개발하고자하는 프로그램에 대한 기본적인 탐구과정과 함께 코딩을 위한 기초 설계의 과정을 거쳐 자신만의 프로그램을 완성하게 된다.

심화 모듈에서는 평가 루브릭이 제시되고 팀 단위 수행과제가 주어져 팀 프로젝트가 이루어진다.

### 3.3 적용된 루브릭(rubric)

<표 2> 루브릭의 평가요소와 기준

단계	평가요소	평가기준
설계	문제의 구조화	적절하게 문제가 분해되고 프로그램의 주요 부분을 논리적인 단위로 나누어 처리하였는가?
	프로그램의 절차 및 알고리즘	프로그램의 절차 및 알고리즘이 효율적으로 설계되었는가?
	추상화와 모듈화	추가블록을 통해 반복적으로 사용되는 핵심 요소가 객체로 정의되었으며 객체 간 관계가 명확하게 정의되었는가?
구현	초기화	프로그램의 초기상태와 최종상태가 정의가 잘 되어있는가?
	병행성과 동기화	동시적 처리나 동기화처리가 적절하게 이루어지는가?
	자료표현	자료의 구성 방법이 효율적인가?
	프로그래밍 요소를 사용하는 능력	프로그래밍요소(구조적, 변수, 조건문, 논리구조, 반복구조, 함수)의 적절한 사용
실행	프로그램 결과	프로그램의 동작 결과가 정확한가? 프로그램이 올바르게 동작하는가?
	기능의 적정성	전체적으로 구현된 프로그램의 기능이 적절하게 설계되었는가?
	실행의 효율성	프로그램의 처리속도와 사용자 입력에 대한 반응속도가 적절한가? 자원들이 적절하게 선택되어 사용되었는가?

학생들에게 제시된 평가 루브릭(rubric)은 <표 1> 과 <표 2>의 내용과 같다. <표 1>에서는 각 단계별 평가요소와 평가 기준을 나타내고, <표 2>에서는 각 평가기준별 수행준거를 나타낸다. 실험집단에게 제공한 루브릭(rubric)은 [17-19]에서 제안한 평가 시트를 참조하여 실험 목적에 맞도록 연구가 재조직하고 재작성한 것이다.

## IV. 연구방법

### 4.1 연구대상

본 연구에서는 C대학의 ‘창의적 사고와 코딩’ 교양 필수 과목을 수강하는 1학년 학생들을 대상으로 실험을 실시하였다. 동일 교수자가 교수하는 동일단과대의 2개 분반을 대상으로 선택하여 집단 차이를 최소화하였다. 통제집단 참여자 수는 39명 실험 집단은 41명으로 연구 대상을 구성하였다.

### 4.2 연구 설계 및 절차

실험은 프로그램언어의 기본 문법 교육이 끝난 중간고사 이후인 9주차 수업부터 6주간에 걸쳐 시행되었다.

컴퓨터 자기효능감을 평가하기 위한 사전 평가는 1주차 수업에 설문을 통해 진행 하였다. 컴퓨터 자기효능감을 진단하기 위해 [15]에서 개발한 컴퓨터 프로그래밍 언어 교육에서 자기 효능감 척도 문항 중에서 프로그래밍 언어 학습에서 자기 효능감 척도 영역을 연구의 목적에 맞게 재구성하여 설문에 이용하였으며 Likert 5점 척도로 측정하였다.

학업성취도의 사전 사후 평가를 위해 동일교과를 담당하는 교수2인과 관련 박사 3인이 공동으로 평가 문제를 제작하여 구성하였다. 사전 평가는 두 그룹 모두 중간고사 이전까지는 동일한 방식으로 수업이 진행되기 때문에 학업 성취도를 위한 사전 평가는 중간고사에 실시하였다. 이를 통해 두 그룹의 동질성을 확인하였다. 9주 이후 수업 진행 과정에서 실습을 진행 할 때마다 실험집단에게만 루브릭을 제공하였다. 실험 집단에게 제공된 루브릭은 표2, 표3과 같다. 학기말에는 두 집단을 대상으로 컴퓨터 자기 효능감에 대한 설문과 학업성취도에 대한 사후

<표 3> 평가 루브릭의 수행수준[17-19]

평가요소	수행수준		
	상	중	하
문제의 구조화	적절하게 문제가 분해되고 프로그램의 주요 부분을 논리적인 단위로 나누어 처리함	문제 분해가 되었으나 적절하지 않고 일부를 논리적인 단위로 나누어 처리함	문제 분해가 이루어지지 않고 논리적 부분이 혼재됨
프로그램의 절차 및 알고리즘	프로그래밍 요소가 매우 적절하게 사용되었다.	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 다소 보인다.	프로그램의 절차 및 알고리즘의 효율성에 개선할 부분이 많이 보인다.
추상화와 모듈화	전체 프로그램에서 분해된 모든 논리적 단위가 새 블록으로 구현됨.	전체 프로그램에서 분해된 논리적 단위가 부분적으로 새 블록으로 구현됨.	사용자 자신이 만든 블록으로 구현된 것이 없음.
초기화	프로그램의 초기상태와 최종상태가 정확하게 정의가 되었다.	프로그램의 초기상태와 최종상태가 어느 정도 정의가 되었다.	프로그램의 초기상태와 최종상태가 정의가 거의 되어있지 않다.
병행성과 동기화	동시적 처리가 적절히 이루어지고 객체간의 메시지 교환을 통해 프로그램 흐름의 동기화가 적절하게 이루어졌다.	동시적 처리는 적절히 이루어졌으나 객체간의 메시지 교환처리가 없거나 적절히 이루어지지 않았다.	동시적 처리나 동기화처리가 이루어지지 않거나 적절하지 않다.
자료표현	변수나 리스트를 적절히 사용하였다. 입출력 부분의 적용을 적절히 해결할 수 있다.	변수 또는 리스트를 사용하였으나 적절하지 못하다. 입출력 부분의 적용을 하였으나 부적절하다.	변수나 리스트가 전혀 사용 되지 않았다. 입출력 부분의 적용이 거의 되지 않았다.
프로그래밍 요소를 사용하는 능력	프로그래밍 요소가 매우 적절하게 사용되었다.	프로그래밍 요소가 어느 정도 적절하게 사용되었다.	프로그래밍 요소가 적절하게 사용되지 않았다.
프로그램 결과	프로그램 결과가 성공적으로 도출되었다.	프로그램 결과가 성공적으로 도출되지 않고 오류가 발생하였으나 프로그램 코드는 전체적으로 잘 구성되어 있다.	프로그램 결과가 성공적으로 도출되지 않고 오류가 발생하며 프로그램 코드도 적합하게 구성되지 않았다.
기능의 적정성	구현된 프로그램의 기능이 적절하게 설계되었다.	구현된 프로그램의 기능이 일부 적절하지 못하다.	구현된 프로그램의 기능이 적절하지 못하다.
실행의 효율성	프로그램의 처리속도와 사용자 입력에 대한 반응속도가 적절하며 자원의 적절하게 사용되었다.	프로그램의 처리속도와 사용자 입력에 대한 반응속도가 적절하며 자원의 사용이 적절하지 않다.	프로그램의 처리속도와 사용자 입력에 대한 반응속도가 느리며 자원의 사용도 적절하지 않다.

검사를 진행하여 루브릭이 코딩 수업의 효과에 미치는 영향을 확인하였다.

### 4.3 연구결과 분석 및 논의

#### 4.3.1 컴퓨터 자기 효능감

실험집단과 비교집단이 동질집단임을 검증하기

위하여 컴퓨터 자기효능감 사전 설문결과에 대한 t 검증을 실시한 결과 두 집단 간에 컴퓨터 자기 효능감의 사전 설문에서는 통제집단의 평균이 실험집단보다 약간 높게 나타났다. 그러나 이는 통계적으로 유의미한 차이는 없었다. 따라서 실험집단과 통제집단은 컴퓨터 자기 효능감에 있어서 실험 처치 전 동일한 집단이라고 할 수 있다.

<표 4> 컴퓨터 자기효능감 사전 평가

구분	n	M	SD	t	p
실험집단	41	9.8	3.67	.484	.630
통제집단	39	10.3	4.04		

루브릭을 제시하여 인지하고 있던 집단의 사후 검사 평균은 19.73점이고, 통제집단의 사후 검사 평균은 16.73점으로 실험 집단의 평균이 3점 더 높게 나타났다. t검정한 결과는 <표 5>와 같으며 이들 사이에는 평균 점수 사이에 유의미한 차이가 있는 것으로 나타났다( $p < .05$ ). 따라서 학업 성취도에서는 루브릭을 제시하여 인지하고 있는 것이 효과적인 것을 알 수 있었다.

<표 5> 컴퓨터 자기효능감 사후평가

구분	n	M	SD	t	p
실험집단	41	19.73	3.3	3.151	.03
통제집단	39	16.53	4.47		

#### 4.3.2 학업 성취도

실험집단과 비교집단이 동일집단임을 검증하기 위하여 학업성취도에 대한 사전검사를 실시하였다. 이를 t검정한 결과 두 집단 간에 학업성취도의 사전 평가는 통계적으로 유의미한 차이를 보이지 않았다. 따라서 실험집단과 통제집단은 학업 성취도에 있어서 실험 처치 전 동일한 집단이라고 할 수 있다.

<표 6> 학업성취도 사전평가

구분	n	M	SD	t	p
실험집단	41	78.16	15.51	-.257	.798
통제집단	39	79.13	14.65		

루브릭을 제시하여 인지하고 있던 집단의 사후 검사 평균은 84점이고, 통제집단의 사후 검사 평균은 75.56점으로 실험 집단의 평균이 4.44점 더 높게 나타났다. 이와 같은 평균의 차이가 통계적으로 의미 있는지 알아보기 위해 t검정한 결과는 표5에 나타나 있다. 이 결과 두 집단 간에는 평균 점수 사이에 유의미한 차이가 있는 것으로 나타났다( $p < .05$ ). 따라서 학업 성취도에서는 루브릭을 제시하여 인지하고 있는 것이 효과적인 것을 알 수 있었다.

<표 7> 학업성취도 사후평가

구분	n	M	SD	t	p
실험집단	41	84	15.19	2.142	.036
통제집단	39	79.56	16.30		

#### 4.3.2 루브릭에 대한 학생 반응

학기말에 실험집단에게만 간단한 설문 문항으로 루브릭에 대한 반응을 살펴보았다. 설문 문항은 아래와 같다.

문항1. 루브릭이 과제를 이해하는데 도움이 되었는가?

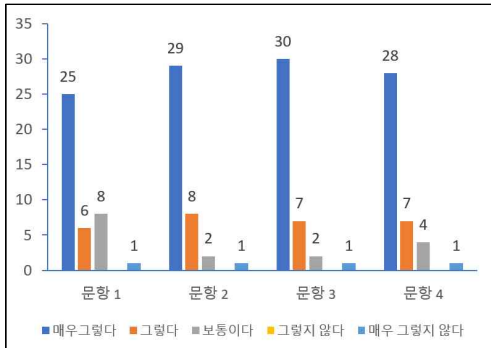
문항2. 루브릭이 과제를 수행하는 과정을 이해하는데 도움이 되었는가?

문항3. 루브릭이 평가기준을 이해하는데 도움이 되었는가?

문항4. 채점된 결과에 대해 신뢰 할 수 있는가?

설문에 대한 응답은 아래의 그림1과 같다. 각 문항에 대한 '그렇다'와 '매우 그렇다' 의견의 평균은 87.5%로 학생들은 루브릭 제시와 평가 기준으로 사용하는 데 대해 긍정적으로 응답하였다.





<그림 1> 루브릭에 대한 학생 의견

## V. 결론

본 연구의 목적은 컴퓨터 비전공자를 대상으로 실시된 코딩 수업에서 루브릭의 인지 여부가 학생들의 학업성취도와 컴퓨터 자기효능감에 어떠한 영향을 미치는가를 알아보는 것이었다. 연구 결과 다음과 같은 결론을 얻을 있었다.

연구 결과 루브릭을 제시하고 인지하여 과제 수행을 실시한 집단이 그렇지 않은 집단에 비해 학업성취도면에서 긍정적인 영향을 미치는 것으로 나타났다. 또한 코딩수업에서 학습자가 루브릭을 인지하는 것은 컴퓨터 자기 효능감 측면에서도 유의미한 효과가 있는 것으로 나타났다.

이는 루브릭의 제시로 학생들은 루브릭에서 설명된 수행준거를 고려하여 과제를 수행하면서, 수행해야 할 과제에 대해 더 잘 이해 할 수 있게 되고 과제를 수행하는 과정에서도 각 각의 준거를 참고할 수 있어 과제와 학습에 대한 성취감을 경험하게 되고 학습의욕과 자신감을 증진시켜 컴퓨터 자기 효능감도 상승하는 것으로 분석된다.

루브릭 사용에 대한 학생들의 반응을 조사한 결과에서 학생들은 대부분 긍정적인 응답을 보였다.

과제수행 전에 학생들에게 루브릭을 제시하는 것이 과제를 수행하는 과정과 그것을 평가하는 기준에 대해 이해하는데 많은 도움이 되었다고 답했다. 즉 학습자 측면에서 루브릭은 어떤 기준에 의하여 수행과제를 평가하는지를 구체적으로 설명하기 때문에 학생이 수행할 과제에 대한 조건과 수행의 수준을 더 명확하게 이해하도록 도와주어 학습의 효과를 더욱 높일 수 있다. 따라서 과제를 수행하기 전에 학생들에게 루브릭을 제공한다면 코딩 수업의 학습을 더욱 촉진할 수 있는 도구로 활용될 수 있을 것이다.

최근 코딩 수업의 도입이 확대되면서 학생들의 학습 성과 향상을 위한 전략으로 루브릭 활용의 중요성이 증가됨에 따라 본 연구와 같은 결과는 의미가 있다. 코딩 수업에서 루브릭은 과제에 대한 구체적인 학습방향을 제시해 주어 비전공 학생들이 전공과 무관한 코딩이라는 과목을 학습하면서 교육환경에 더 빨리 적응하고 컴퓨팅적인 내용을 체계적으로 학습하는데 유용한 도구임을 확인할 수 있었다.

현재 루브릭이 많이 개발, 연구 되고 있으나 비전공자의 코딩 교육에 대한 루브릭의 연구와 개발은 많지 않다. 학습자의 코딩 능력을 신장시킬 수 있도록 학생들의 학습 수행 결과에 대해 평가하고 학생들에게 제공할 수 있는 교양 코딩 교육의 여러 영역을 위한 타당도와 신뢰도가 높은 루브릭이 개발되어야 한다.

대학 코딩 수업의 평가에서 개발된 루브릭을 사용하고 이에 대해 지속적으로 보완하고 업데이트할 수 있는 작업을 통해 개발된 루브릭의 효과를 명확히 하여 코딩 교과목 관련 루브릭의 표준화를 마련할 필요가 있다.

## 참고문헌

- [1] 이애리, “코딩수업을 위한 학습자 중심의 플립드 러닝 적용 사례 연구,” 실천공학교육논문지, 제9권, 제1호, 2017, pp.23-30.
- [2] 김기훈·엄정호, “사이버보안 전문인력 획득을 위한 사이버보안 훈련생에 특화된 훈련성과 측정 모델에 관한 연구,” 디지털산업정보학회논문지, 제12권, 제4호, 2016, pp.59-69.
- [3] Goodrich, “H. Understanding Rubrics. Educational Leadership,” Vol.54, No.4, 1997, pp.14-17.
- [4] Josson, A. & Svingby, G., “The use of scoring rubrics: Reliability, validity and educational consequences,” Educational Research Review, Vol.2, No.2, 2007, pp.130-144.
- [5] 이인영, “루브릭이 대학생들의 글쓰기 능력 신장에 미치는 효과,” 우리말글학회, 제48권, 2010, pp.29-53.
- [6] 오미자, “스크래치 프로그램을 활용한 프로그래밍 교육에 대한 비전공자의 인식 연구,” 컴퓨터교육학회논문지, 제20권, 제1호, 2017, pp.1-11.
- [7] 박지현, “플립러닝기반 소프트웨어 교육이 창의적 문제해결력 및 수업만족도에 미치는 효과 -대학교 교양 소프트웨어 수업 중심으로-,” 한국방송통신대학교 대학원 석사학위 논문, 2017.
- [8] 박성희, “컴퓨팅 사고력(Computational Thinking) 함양을 위한 대학에서의 SW교육에 관한 고찰,” 한국디지털정책학회, 제14권, 제4호, 2016, pp. 1-10.
- [9] 한지영, “기술과 교육 평가에서 학습자 중심 루브릭이 학습과정 및 학습성취에 미치는 영향,” 서울대학교 대학원 박사학위논문, 2004.
- [10] 남명호·김성숙·지은림, “수행평가 이해와 적용,” 문음사, 2000.
- [11] Judith Arter and Jay McTighe, “Scoring rubrics in the classroom: using performance criteria for assessing and improving student performance,” Thousand Oaks, Calif: Corwin Press, 2001.
- [12] 범선화·채정현, “중학교 가정교과 수행평가를 위한 루브릭(rubric) 개발 - 실험·실습법에 적용,” 한국가정교과교육학회지, 제20권, 제3호, 2008, pp.85-105.
- [13] Bandura, A, “Self-efficacy: Toward a unifying theory of behavioral change,” Psychological Review, Vol.84, No.2, 1977, pp.191-215.
- [14] 전병호, “대학 수업에서의 블렌디드 러닝 만족에 영향을 미치는 학습자 변인 연구,” 디지털산업정보학회 논문지, 제13권, 제3호, 2017, pp.105-113.
- [15] 김갑수, “컴퓨터 프로그래밍 언어 교육에서 자기 효능감 척도 개발 및 적용,” 정보교육학회논문지, 제18권, 제1호, 2014, pp.111-120.
- [16] 이용규, “정보기술 사용 전유에 관한 변수들 간의 관계에서 컴퓨터자기효능감의 조절효과,” 정보시스템연구, 제18권, 제1호, 2009, pp.25-46.
- [17] 최형신, “Computational Thinking역량 개발을 위한 수업 설계 및 평가 루브릭 개발,” 정보교육학회논문지, 제18권, 제1호, 2014, pp.57-64.
- [18] 김민자·유길상·김현철, “비전공자 프로그래밍 수업 창의적 산출물의 컴퓨팅 사고력 기반 평가 루브릭 개발,” 컴퓨터교육학회논문지, 제20권, 제2호, 2017, pp.1-11.
- [19] 안성훈, “초·중학교 SW교육을 위한 프로그래밍 평가지표 개발,” 컴퓨터교육학회논문지, 제19권, 제4호, 2016, pp.11-20.

■ 저자소개 ■



이 에 리  
(Lee Aeri)

2014년 3월~현재  
가톨릭관동대학교 VERUM교양대학  
교양과 조 교수  
2007년 2월 명지대학교 컴퓨터공학과(공학박사)  
관심분야 : 네트워크 보안, IOT 보안, 옛지 컴퓨팅,  
코딩 교육,  
E-mail : allee@cku.ac.kr



김 창 희  
(Kim Changhee)

2008년 3월~현재  
서울기독대학교 국제경영정보학과  
교수  
2007년 2월 명지대학교 컴퓨터공학과(공학박사)  
2000년 2월 명지대학교 컴퓨터공학과(공학석사)  
1998년 2월 명지대학교 컴퓨터공학과(공학사)  
관심분야 : 컴퓨터 교육, Social Computing,  
e-learning  
E-mail : area88@scu.ac.kr

논문접수일 : 2017년 11월 20일  
수 정 일 : 2017년 11월 30일  
게재확정일 : 2017년 12월 01일