

# 探討 Code.org 程式學習平台對於國小四年級學童程式自我

## 效能、運算思維態度之影響：以 Hour of Code 課程為例

### The Effect of Code.org Program Learning Platform on 4th Grade Elementary School Students' Programming Self-efficacy and Computational Thinking Attitude: An Example of Hour of Code Course.

林令惠<sup>1</sup> 陳素芬<sup>2</sup>

LIN, LING HUI<sup>1</sup> CHEN, SU FEN<sup>2</sup>

<sup>1</sup> 國立臺灣科技大學 數位學習與教育研究所 研究生

<sup>1</sup> National Taiwan University of Science and Technology  
Graduate Institute of Digital Learning and Education Student

E-mail : [tf\\_10729@hotmail.com](mailto:tf_10729@hotmail.com)

<sup>2</sup> 國立臺灣科技大學 數位學習與教育研究所 教授

<sup>2</sup> National Taiwan University of Science and Technology  
Graduate Institute of Digital Learning and Education Professor

E-mail : [sufchen@mail.ntust.edu.tw](mailto:sufchen@mail.ntust.edu.tw)

#### 摘要

本研究旨在探討 Code.org 對於國小四年級學童的程式自我效能與運算思維態度之影響。本研究採預實驗研究的單組前後測設計，以基隆市地區兩所公立小學之四年級學生為研究對象，研究對象共 48 人。研究工具為 Code.org 中 Hour of Code 課程，實驗期間共 6 週，每週一次，每次 40 分鐘，並於實驗課程前後施測「程式自我效能問卷」與「運算思維態度信念問卷」。研究結果指出全體學生在「程式自我效能」與「運算思維態度」的全構面得分上，後測平均分數皆高於前測平均分數，且都達到顯著差異；若從不同性別差異來看，雖然男學生在兩份問卷皆達顯著差異，女學生僅有程式自我效能達顯著差異，運算思維態度未達顯著差異，但是不同性別的後測平均分數皆高於前測。最後整體而言，學生透過 Code.org 平台學習，對程式自我效能與運算思維態度皆有提升。

**關鍵字：**Code.org、自我效能、運算思維

## Abstract

This study aims to investigate the effect of ‘Code.org’ on fourth grade elementary school students’ programming self-efficacy and computational thinking attitude: An Example of Hour of Code Course. One-group pretest-posttest design was adopted in this study. The research participants were fourth grade students from two public elementary schools in Keelung City. The experiment lasted for six weeks, with one 40-minute-long session every week. ‘Code.org’ website, ‘Programming Self-efficacy Questionnaire’ and ‘Computational Thinking Attitude Questionnaire’ are the research tools for this study. Research participants were asked to fill in the questionnaire before and after the experiment. According to the paired sample t-test results, it was found that all students’ mean value of post-questionnaire scores is higher than pre-questionnaire scores, there was a significant difference between pre-test and post-test. In terms of genders, male students were a significant difference in all questionnaire scores. On the other hands, female students were a significant difference of scores in ‘Programming Self-efficacy Questionnaire’ only but not significant difference of scores in ‘Computational Thinking Attitude Questionnaire’. However, both genders showed that the mean of post-questionnaire scores is higher than pre-questionnaire scores. In conclusion, ‘Code.org’ learning platform has improved both their programming self-efficacy and computational thinking attitude.

**Keywords : Code.org, Self-efficacy, Computational Thinking**

# 壹、前言

## 一、研究背景與動機

我國自 2019 年實施十二年國民基本教育課程綱要(簡稱 108 新課綱)，將程式設計納入國高中課程當中，以培養學生運算思維為主要課程理念，希望學生能有效利用運算思維與資訊科技工具解決問題、合作共創與溝通表達(國家教育研究院，2018)。至於國小階段原本安排從三年級培養學生科技與資訊應用的基本素養，將改以資訊素養融入在各個領域中，結合其他領域之課程內容來落實，不再安排固定課程，或是建議各縣市及各校的資訊科技課程延後至五年級才開始進行每週 1 或 2 節的彈性學習課程。因此國小階段的資訊科技課程沒有審定過的課程教材，也沒有各年級能力指標，一切都依照各縣市、各校來訂定課程內容。

目前六都已經完成了自訂版的國小資訊教育課程，大部分的縣市大多是參考六都的課程架構，加入地方需求編寫而成，導致各個縣市自訂版課程的完成進度落差甚大。即便是同一縣市內的學校，在師資上也有很大的落差，根據本研究者在國小擔任資訊教師的經驗，部分兼任資訊教師並沒有具備足夠的資訊概念。如果由各縣市政府、各校去制定不同課程，造成原有的數位落差更明顯。另外在 108 新課綱訂定之後，許多現場教師紛紛投入教學活動設計及教材開發，但是多數教師缺乏運算思維的訓練、資訊融入教學的經驗，甚至是自身資訊能力不足，使得新課程架構的撰寫多以舊課程內容修改而成，無法引發學生更高層次思考。

因此選擇運算思維學習工具就顯得重要，Code.org 是視覺化程式積木的平台網站，學生不需透過複雜的程式語言就能學習，在遊戲中培養運算思維、想像力以及解決問題能力，同時也熟悉程式設計的基本概念(Code.org, 2016)。

不過 Liang 與 Sedig(2007)研究指出，對於剛接觸電腦軟體教學的兒童，不容易透過視覺化程式軟體掌握複雜的程式概念，在操作上容易產生認知負荷。Code.org 有完整的引導系統整合到平台中，在適當的情況下給予提示，讓學生能運用程式相關知識(Buwalda, Jeurig, & Naus, 2018)，並根據自身評估學習狀況與困難，自行決定是否使用系統給予的提示(Piech, Sahami, Huang, & Guibas, 2015)。

## 二、研究目的與問題

基於上述背景與動機，過去研究很少探討 Code.org 融入國小資訊課程中，對於程式自我效能與運算思維態度影響高低，故本研究旨在探討 Code.org 對於國小四年級學童的程式自我效能與運算思維態度之影響。此外希望透過 Code.org 平台，讓不同年齡層的學生皆可透過平台上安排的課程，學習程式設計概念與運算思維核心素養，更提供給非資訊相關背景的教師教學上的參考。

基於上述研究目的，具體研究問題詳述如下：

1. 國小四年級使用 Code.org 平台學習後，是否能提升程式自我效能？
2. 不同性別的程式自我效能是否有差異？
3. 國小四年級使用 Code.org 平台學習後，是否能提升運算思維態度？
4. 不同性別的運算思維態度是否有差異？

## 貳、文獻探討

### 一、Code.org 程式學習平台

Code.org 於 2013 年成立，是由 Hadi Partovi 兄弟建立的一個非營利性教育機構網站，成立宗旨是為了鼓勵美國學生學習電腦科學，以促進學校的電腦科學課程。網站免費提供編程，且大部分支援繁體中文，少部分尚未翻譯完全。課程內容有不插電程式課程與程式積木課程，其中最著名的就是「一小時玩程式(Hour of Code)」，積木塊背後有對應好的 JavaScript 程式語言，即使沒有程式相關背景的教師，也可以和學生一同學習，相當容易上手。

此外，Code.org 具備完整的學習管理系統，教師可以透過後台輕鬆設置學生帳號，監控學生的進度，課程皆有明確的學習範圍和流程，教師除了可以將課程分配給不同級別的學生，教師也可以對學生進行特定程式概念考核。

本研究將採用 Code.org 的 Hour of Code 課程中經典迷宮(Classic Maze)一內容有憤怒鳥(Angry Birds)、植物大戰殭屍(Plant vs. Zombies)、冰原歷險記(Ice Age)，此活動共有 20 個關卡，學習重點有透過拖拉程式積木來寫程式，學習「循序」、「迴圈」、「條件」概念。

### 二、自我效能理論與相關研究

「自我效能(self-efficacy)」理論，由著名美國心理學家 Bandura(1986)提出，指個體對自己是否能夠從事或完成某種行動的信念強弱或意志程度。簡單來說就是，個體在面對各種情況時能夠產生特定行為的自信能力(李尹暘、林曉佩、林君怡，2007)。自我效能較低的人在行動中容易卻步，自我效能較高的人則會積極參與且比較願意投入及堅持努力。

近年來，由於資訊科技快速發展，電腦與個人這兩者的探討與研究逐漸增多。Hill、Smith 與 Mann(1987)將自我效能理論運用在資訊科技中，發展出「電腦自我效能(Computer self-efficacy)」。Compeau 與 Higgins(1995)定義，電腦自我效能是代表個體對於自己能運用電腦與其相關知識去完成一特定任務的能力，並非只是反應一項單純的電腦知識與技能。簡而言之，就是電腦使用者自我覺察在使用電腦時所期望的信心程度(Murphy, 1989)。

許多研究皆說明自我效能的重要性(Matsui & Ohnishi, 1990; Lent, Lopez, & Bieschke, 1993)，指出自我效能與學習成效呈現正相關。吳文雄(2002)和韓宜娣(2011)指出電腦自我效能愈高者，其學習成效、學習態度以及程式設計學習表現愈高，所以電腦自我效能可以視為學習成效的一種指標。

綜合上述，本研究欲探討學生在 Code.org 學習程式時，是否能藉由 Code.org 完善的引導提示系統輔助，提升程式學習的自我效能。量測工具為蔡雯欣(2018)所編製的「程式自我效能」問卷，量測學生在程式積木課程中學習的自信程度。

### 三、 運算思維定義與相關研究

「運算思維(Computational Thinking)」一詞，由美國卡內基美隆大學學者Wing(2006)提出，運用電腦科學基本概念來解決問題、設計系統和理解人類的行為。除了閱讀、寫作和算數基本素養能力之外，運算思維是目前每個人該具備素養能力之一。運算思維將看似困難且複雜的問題，透過簡化，嵌入，轉換或仿真，轉化為我們知道的方式來解決，可增進資訊科技應用、程式設計、或資訊科學的學習成效，具備運算思維能更善用運算來解決日常生活中的問題。

運算思維是從提出問題到解決方案的思考過程，但是運算思維的觀念建立並不是只能藉由寫程式培養，可以透過不插電的實作、體驗方式建立思維模式，只是程式語言是最快速且清楚培養運算思維的工具之一(江其勳，2017)。

自運算思維一詞被提出之後，許多學者也分別對運算思維定義提出不同見解(Wing, 2006; CSTA 2011, 2015; ISTE, 2011; Barr & Stephenson, 2011; Grover & Pea, 2013; Selby & Woollard, 2014)。為了統整諸多學者定義，讓全世界教學者有所依據，美國 Google 公司提供運算思維線上資源 Google for Education(2015)，網站中定義運算思維是一種利用運算來解決問題，包含問題拆解、模式識別、抽象化、演算法設計等。

除了具備運算思維能力之外，Barr 與 Stephenson (2011)建議，在設計中小學運算思維課程，如果要找到學生在運算思維中的價值觀、動機、情感、觀念和態度方面，需要包含處理複雜問題的信心、處理困難問題的耐心、處理歧義的能力、處理開放式問題的能力以及團隊工作中的溝通技巧。學生可以透過程式設計中學習到運算思維的態度與技能(Lee, 2015)，因為程式設計不只是學程式撰寫而已，更可以立即知道運行後的結果，增強學生對運算思維的概念。視覺化程式設計會比傳統的程式語言教學，更能幫助學生理解運算思維的概念與提升學習態度(Lye & Koh, 2014)。

綜合上述，本研究將透過 Code.org 讓學生學習運算思維的概念以及提升學習程式語言的態度。量測工具為自編「運算思維態度信念」問卷，而自編問卷的構面依照 Barr 等人對中小學運算思維課程設計建議，量測學生在面對問題時的解決問題之信心程度、評估相關可行性、態度看法。

## 參、 研究實施與設計

### 一、 研究方法

#### (一) 研究設計

本研究採預實驗研究的單組前後測設計，因為同所學校無法找到 2 個班級皆為相同教師授課，所以無法做到隨機取樣和隨機分配，故採單組前後測設計。

#### (二) 研究流程

本研究實驗時間為期五週，每週上一節課，每節課 40 分鐘。在第一週教學

實驗之前會進行「程式自我效能問卷」、「運算思維態度信念問卷」前測。在第五週教學實驗之後會進行「程式自我效能問卷」、「運算思維態度信念問卷」後測。第二到第四週為實驗介入期間。

### (三) 研究對象

本研究的研究對象為基隆市兩所國小四年級學生，每所學校選擇 1 個班級，共 2 個班級。學生從三年級開始接觸學校的彈性資訊課程，已具備基礎電腦操作能力、基礎文書處理能力、利用瀏覽器瀏覽網頁、搜尋相關資料、未上過運算思維相關課程、無接觸過其他程式積木。

## 二、 研究工具

### (一) 程式自我效能問卷

本問卷採用蔡雯欣(2018)所編製的「程式自我效能」問卷，主要是量測學生對於軟體使用與完成關卡之個人能力信心程度。題目總共 10 題，計分方式採 Likert Scale 五點量表，依序為 5 代表非常同意、4 代表同意、3 代表沒意見、2 代表不同意、1 非常不同意，計分方式為回答 1 者給 1 分，回答 2 者給 2 分，以此類推，若得分越高代表程式自我效能程度越高，於實驗課程前後分別進行施測，以了解研究對象的程式自我效能之差異。正式實驗前對此量表進行信度內部一致性考驗，全量表各向度 Cronbach's Alpha = 0.93，符合理想(0.7 以上)。

### (二) 運算思維態度信念問卷

本問卷為研究者參考 Standl(2017)、Yadav 等人(2011)的問卷，翻譯修改編制。Standl(2017)問卷構面來自於 Barr 與 Stephenson (2011)提出的中小學運算思維課程設計須具備的態度。Yadav 等人(2011)發展問卷來評估學生對於運算思維的態度與理解。題目總共 20 題，計分方式採 Likert Scale 五點量表，依序為 5 代表非常同意、4 代表同意、3 代表沒意見、2 代表不同意、1 非常不同意，計分方式為回答 1 者給 1 分，回答 2 者給 2 分，以此類推，若得分越高代表問題解決與運算思維態度越高，於實驗課程前後分別進行施測，以了解研究對象的問題解決與運算思維態度之差異。

本問卷正式實驗前進行預試，預試之後進行項目分析與因素分析，結果顯示 KMO = 0.944 (>0.6)、球形檢定 Bartlett's test = 0.00 (<0.05)符合標準，最後分析信度內部一致性考驗，全量表各向度 Cronbach's Alpha = 0.93，符合理想(0.7 以上)。

本問卷共有三個構面，構面 1：處理複雜問題的信心程度(第 1 題至第 4 題)；構面 2：評估處理複雜問題的可行性(第 5 題至第 12 題，第 5 題為反向題)；構面 3：對於運算思維的態度(第 13 題至第 20 題)。

## 三、 資料分析

為了瞭解學生在實驗課程前後，程式自我效能與運算思維態度之變化情況，本研究者採用 SPSS 22 版本軟體，並透過相依樣本 T 檢定來檢驗前、後側之差異，分析所得結果請見下個章節。

## 肆、結果與討論

### 一、程式自我效能

由表 4-1 的結果顯示，在接受 Code.org 的 Hour of Code 課程之前，全體學生的程式自我效能問卷前測的平均分數為 3.28，標準差為 0.67，實施課程之後，程式自我效能問卷後測的平均分數為 3.83，標準差為 0.80。表示經過 Hour of Code 課程實驗之後，程式自我效能的平均分數有所提升。

進一步觀察相依樣本 t 檢定摘要表，如表 4-2，得知 t 值為-4.533，p 值為  $0.000 < 0.001$ ，已達顯著差異，表示程式自我效能前、後測間的得分有顯著的差異存在，且後測平均分數(M=3.83)顯著高於前測平均分數(M=3.28)。

表4-1 程式自我效能問卷前、後測之平均數、標準差

分析項目	N	前測 Pre-test		後測 Post-test	
		平均數	標準差	平均數	標準差
		M	SD	M	SD
程式自我效能	48	3.28	0.67	3.83	0.80

表4-2 程式自我效能問卷前、後測之相依樣本t檢定摘要表

分析項目	平均數	標準差	T 值	自由度	P 值
	M	SD		df	
程式自我效能	-0.55208	0.84	-4.533	47	.000***

\* $p < .05$ ; \*\* $p < .01$ ; \*\*\* $p < .001$

為了更深入了解不同性別的程式自我效能前、後測之差異，表 4-3 為男、女學生的平均分數與標準差，男學生的程式自我效能問卷前測的平均分數為 3.43，標準差為 0.64；女學生的程式自我效能問卷前測的平均分數為 3.10，標準差為 0.68，實施課程之後，男學生的程式自我效能問卷後測的平均分數為 4.04，標準差為 0.14，女學生的程式自我效能問卷後測的平均分數為 3.58，標準差為 0.85。

表示透過此期間的實驗課程，不論是男學生或是女學生平均分數皆提高，且根據男、女學生前、後測之相依樣本 t 檢定摘要表顯示，男學生( $t = -.61154$ ,  $p = 0.002 < 0.01$ )和女學生( $t = -.48182$ ,  $p = 0.011 < 0.05$ )兩者皆達顯著差異，意味著不同性別學生在參與 Hour of Code 課程實驗之後，其程式自我效能皆有顯著成長。

表4-3 程式自我效能問卷不同性別的前、後測之平均數、標準差

分析項目	男生			女生		
	N	前測 M(SD)	後測 M(SD)	N	前測 M(SD)	後測 M(SD)
	程式自我效能	26	3.43(0.64)	4.04(0.14)	22	3.10(0.68)

表4-4 程式自我效能問卷男學生的前、後測之相依樣本t檢定摘要表

分析項目	平均數 M	標準差 SD	T 值	自由度 df	P 值
程式自我效能	-.61154	0.88	-3.550	25	.002**

\*p&lt;.05; \*\*p&lt;.01; \*\*\*p&lt;.001

表4-5 程式自我效能問卷女學生的前、後測之相依樣本t檢定摘要表

分析項目	平均數 M	標準差 SD	T 值	自由度 df	P 值
程式自我效能	-.48182	0.82	-2.771	21	.011*

\*p&lt;.05; \*\*p&lt;.01; \*\*\*p&lt;.001

## 二、 運算思維態度

由表 4-6 的結果顯示，在接受 Code.org 的 Hour of Code 課程之前，全體學生的運算思維態度信問卷「處理複雜問題的信心程度」構面前測的平均分數為 3.13，標準差為 0.85；「評估處理複雜問題的可行性」構面前測的平均分數為 3.62，標準差為 0.67；「對於運算思維的態度」構面前測的平均分數為 3.52，標準差為 0.61，實施課程之後，程式自我效能問卷「處理複雜問題的信心程度」構面後測的平均分數為 3.79，標準差為 0.92；「評估處理複雜問題的可行性」構面後測的平均分數為 3.87，標準差為 0.78；「對於運算思維的態度」構面後測的平均分數為 3.95，標準差為 0.81。表示經過 Hour of Code 課程實驗之後，運算思維態度的三個構面平均分數皆有所提升。

進一步觀察相依樣本 t 檢定摘要表，如表 4-7，得知「處理複雜問題的信心程度」構面 t 值為 -4.461，p 值為  $0.000 < 0.001$ ；「評估處理複雜問題的可行性」構面 t 值為 -2.486，p 值為  $0.017 < 0.05$ ；「對於運算思維的態度」構面 t 值為 -3.986，p 值為  $0.000 < 0.001$ ，三個構面皆達顯著差異，表示運算思維態度前、後測間的得分有顯著的差異存在，且三個構面後測平均分數皆顯著高於前測平均分數。

表4-6 運算思維態度信念問卷前、後測之平均數、標準差

分析構面	N	前測 Pre-test		後測 Post-test	
		平均數	標準差	平均數	標準差
		M	SD	M	SD
處理複雜問題的信心程度	48	3.13	0.85	3.79	0.92
評估處理複雜問題的可行性	48	3.62	0.67	3.87	0.78
對於運算思維的態度	48	3.52	0.61	3.95	0.81

表4-7 運算思維態度信念問卷前、後測之相依樣本t檢定摘要表

分析項目	平均數	標準差	T 值	自由度	P 值
	M	SD		df	
處理複雜問題的信心程度	-.65625	1.02	-4.461	47	.000***
評估處理複雜問題的可行性	-.25260	0.70	-2.486	47	.017*
對於運算思維的態度	-.43229	0.75	-3.986	47	.000***

\*p<.05; \*\*p<.01; \*\*\*p<.001

表 4-8、4-9、4-10 呈現不同性別對運算思維態度各構面前、後測之差異。在「處理複雜問題的信心程度」構面(男學生：前測 M=3.04, 後測 M=4.00, p<0.001; 女學生：前測 M=3.24, 後測 M=3.53, p>0.05), 男學生達顯著差異, 女學生未達顯著差異。在「評估處理複雜問題的可行性」構面(男學生：前測 M=3.50, 後測 M=4.02, p<0.001; 女學生：前測 M=3.76, 後測 M=3.69, p>0.05), 男學生達顯著差異, 女學生未達顯著差異。在「對於運算思維的態度」構面(男學生：前測 M=3.65, 後測 M=4.19, p<0.001; 女學生：前測 M=3.37, 後測 M=3.68, p>0.05), 男學生達顯著差異, 女學生未達顯著差異。

由上述不同性別的分析結果來看, 男學生在三個構面上皆達顯著差異, 女學生則未達顯著差異, 其原因可能有兩個, 原因一, 因為本次 Hour of Code 選擇經典迷宮關卡, 人物大多為電玩遊戲主角(例如: 憤怒鳥、植物大戰殭屍), 所以在進行闖關時, 男學生會比女學生多些自信與耐心去解決問題, 也讓多數的男學生對程式設計產生更濃厚的興趣, 認為自己未來可以設計出如同電玩的程式。原因二, 雖然女學生皆未達顯著差異, 但是可以發現女學生在前測時, 大部分的構面平均分數高於男學生, 所以在後測時平均分數提升幅度沒有男學生高, 假設此次實驗選擇的關卡是女學生偏好的卡通人物(例如: 冰雪奇緣), 或許後測平均分數有機會再往上提高。

表4-8 運算思維態度信念問卷不同性別的前、後測之平均數、標準差

分析構面	男生			女生		
	N	前測 M(SD)	後測 M(SD)	N	前測 M(SD)	後測 M(SD)
處理複雜問題的信心程度	26	3.04 (0.74)	4.00 (0.92)	22	3.24(0.97)	3.53(0.87)
評估處理複雜問題的可行性	26	3.50(.66)	4.02(0.76)	22	3.76 (0.68)	3.69(0.78)
對於運算思維的態度	26	3.65(0.63)	4.19(0.62)	22	3.37(0.57)	3.68(0.92)

表4-9 運算思維態度信念問卷男學生的前、後測之相依樣本t檢定摘要表

分析構面	平均數 M	標準差 SD	T 值	自由度 df	P 值
處理複雜問題的信心程度	-.96154	0.88	-5.575	25	.000***
評估處理複雜問題的可行性	-.52404	0.55	-4.868	25	.000***
對於運算思維的態度	-.53846	0.59	-4.668	25	.000***

\*p<.05; \*\*p<.01; \*\*\*p<.001

表4-10 運算思維態度信念問卷女學生的前、後測之相依樣本t檢定摘要表

分析構面	平均數 M	標準差 SD	T 值	自由度 df	P 值
處理複雜問題的信心程度	-.29545	1.07	-1.291	21	.211
評估處理複雜問題的可行性	.06818	0.74	.431	21	.671
對於運算思維的態度	-.30682	0.91	-1.588	21	.127

\*p<.05; \*\*p<.01; \*\*\*p<.001

## 伍、未來展望

綜合第四章的結果與討論，簡單歸納出結論，研究對象全體學生在「程式自我效能」與「運算思維態度」的「處理複雜問題的信心程度」、「評估處理複雜問題的可行性」、「對於運算思維的態度」得分上，後測平均分數皆高於前測平均分數，且都達到顯著差異；若從不同性別差異來看，雖然男學生兩者平均得皆達顯著差異，女學生僅有程式自我效能達顯著差異，運算思維態度未達顯著差異，但是不同性別的後測平均分數皆高於前測(除了「評估處理複雜問題的可行性」構面除外，但是此構面的前後測平均分數相差甚微，且都超過3.5分以上)。可以認為兩性刻板印象在逐漸減少，男女生不分領域上發展皆應該平等對待，即便在許多學科或領域上，性別還是有些微的差異，但是在思維的能力上(例如：運算思維、設計思維等)，目前尚未有研究提出男女有顯著差異(Taylor & Baek, 2019)。

本研究對於未來的建議，可以增加對照組來提升研究的信效度，對照組的工具可以採用其他程式積木相關軟體比較(例如：Scratch、mBot、micro:bit、Blocky...)。如果要更深入探討性別和運算思維的關聯，建議分組上可以研究分為單一性別分組或男女混和分組。本研究由於受到新型冠狀病毒肺炎影響，因此實驗介入時間受到限制，未來若想要做國小程式教育相關之研究，應該至少將課程拉長至半學期到一學期，希望長時間的思維培養，可以讓學生達到更高層次的思維能力。

## 參考文獻

### 一、中文部分

- 教育部(2018)。十二年國民基本教育課程綱要國民中學暨普通型高級中等學校-科技領域。臺北：國家教育研究院。
- 李尹暘、林曉佩、林君怡(2007)。自我效能理論之分析與應用。《澄清醫護管理雜誌》，3(2)，46-52。
- 吳文雄(2002)。電腦技能學習者過去的績效、目標認同、電腦自我效能及電腦績效因果關係之驗證—社會認知理論與目標設定理論的整合。《師大學報：科學教育類》，47(1)，39-54。
- 韓宜娣(2011)。鷹架支持與自我效能對國小學生程式設計學習表現與學習態度之影響。國立臺灣師範大學資訊教育學系碩士論文。
- 蔡雯欣(2018)。「不插電的資訊科學」課程對國小五年級學生於電腦程式設計課程之學習態度、自我效能與運算思維能力之影響。國立臺灣科技大學數位學習與教育研究所碩士論文。

## 二、英文部分

- Bandura, A. (1994). Self-efficacy. *Encyclopedia of human behavior*, 4, 71-81.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community. *Inroads*, 2(1), 48-54.
- Buwalda, M., Jeurig, J., & Naus, N. (2018). Use expert knowledge instead of data: Generating hints for hour of code exercises. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, 1-4.
- Compeau, D. R., & Higgins, C. A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS quarterly*, 189-211.
- CSTA, K. (2015). Computer Science Standards: Revised 2011.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational researcher*, 42(1), 38-43.
- Hill, T., Smith, N. D., & Mann, M. F. (1987). Role of efficacy expectations in predicting the decision to use advanced technologies: The case of computers. *Journal of applied psychology*, 72(2), 307.
- CSRA & ISTE. (2011). Computational Thinking in K-12 Education leadership toolkit. <http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershipToolkit-SP-vF.pdf>
- Lent, R. W., Lopez, F. G., & Bieschke, K. J. (1993). Predicting mathematics-related choice and success behaviors: Test of an expanded social cognitive model. *Journal of vocational behavior*, 42(2), 223-236.
- Liang, H. N., & Sedig, K. (2007). Presentation of feedback in interactive learning environments: Some dimensions. In *EdMedia+ Innovate Learning*, 972-977. Vancouver, Canada: Association for the Advancement of Computing in Education (AACE).
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. *Computers in Human Behavior*, 41, 51-61
- Matsui, T., Matsui, K., & Ohnishi, R. (1990). Mechanisms underlying math self-efficacy learning of college students. *Journal of Vocational Behavior*, 37(2), 225-238.
- Murphy, K. M., Shleifer, A., & Vishny, R. W. (1989). Industrialization and the big push. *Journal of political economy*, 97(5), 1003-1026.
- Piech, C., Sahami, M., Huang, J., & Guibas, L. (2015). Autonomously generating hints by inferring problem solving policies. In *Proceedings of the Second (2015) ACM Conference on Learning at Scale*, 195-204.

- Selby, C., Dorling, M., & Woollard, J. (2014). Evidence of assessing computational thinking.
- Standl, B. (2017). Solving everyday challenges in a computational way of thinking. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, 180-191.
- Taylor, K., & Baek, Y. (2019). Grouping matters in computational robotic activities. *Computers in Human Behavior*, 93, 99-105.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, 465-470.

