

# Comparison of Term Recognition and Awareness in Blended Classes of JavaScript and PHP Programming

Isao Miyaji<sup>1</sup>

<sup>1</sup> National Institute of Technology, Toyama College, Imizu, Toyama, Japan

Correspondence: Isao Miyaji, Ph.D. in Engineering, Toyama College Fellow, Musa 1392, Kita-ku, Okayama, 701-2141 Japan. Tel: 81-86-229-3799. E-mail: s8u3m3i2re@hotmail.co.jp

Received: December 11, 2016

Accepted: January 11, 2017

Online Published: February 3, 2017

doi:10.20849/aes.v2i1.108

URL: <http://dx.doi.org/10.20849/aes.v2i1.108>

## Abstract

A lecture course is provided by projecting slides on contents of the lecture course on a screen and the grammar in textbook and JavaScript program examples are explained in a lecture course on programming. It was then followed by explanation with materials describing program examples to practice the grammar and exercise problems. We conducted exercises to prepare programs in exercise problems with reference to the program examples. Instructions were given to complete the program during the lecture course as soon as possible, and the students were asked to submit the program and report files using the functions of e-learning. During the 14th week, each student was asked to prepare a program of voluntary subject he or she came up with, execute it and evaluate one another. With reference to the results, they were asked to correct them by the final 15th week and to evaluate one another again in the 15th week. The degree of term recognition and awareness were investigated before and after the lecture course. This article reports on the results obtained regarding this information on JavaScript programming. Then the difference between JavaScript and PHP programming is reported by comparing them with the results regarding PHP programming reported previously.

**Keywords:** JavaScript, PHP, term recognition, awareness, blended learning

## 1. Introduction

At present, blended learning is used mainly in institutions of higher education in order to make lecture course effective, efficient and attractive (Bersin, 2004, Miyaji, 2009). The author works on college education to nurture problem-solving abilities by incorporating manufacturing and evaluation in activities (Miyaji, 2005). It has been suggested that lectures should be supported with increasing the opportunities for various students to learn so that they can make preparations and do reviews “whenever and wherever” with individual support (Central Council for Education, 2012).

As a part of such support, the author reported that implementation of blended lecture combining lecture course organization note, e-learning (learning with lecture slides, learning with exercise problems, mutual learning and evaluation using materials prepared by the students), small tests and so forth was effective (Miyaji et al., 2005). We also reported that the effect could be further improved by increasing the interactions with the teacher when degree of understanding survey was incorporated (Miyaji et al., 2007).

Several methods to deepen students' understanding in programming class have been proposed. Takaoka et al. (2008) have designed and developed Java programming online course materials and conducted classes in blended style and ‘fully e-learning’ style. They found that it is important that guidance and face-to-face sessions should be conducted in a suitable fashion, in addition to well-designed coursework, to achieve ‘fully e-learning’ successfully (Takaoka et al., 2008). Taniguchi (2011) designed and implemented Visual Basic Programming so that students could collaboratively learn by sharing information during the process of creating programs. She found that there were more opportunities for students to read the code of other students, and they were able to learn collaboratively while receiving positive stimulation and influence (Taniguchi, 2011).

Methods to deepen the understanding in PHP and C programming lecture course have been proposed. Of these, blended lecture has been reported to be effective. The report discusses implementing collaborative learning and evaluating the works (Miyaji et al., 2014a, Shinkai et al. 2011). It also reports on activities that help improve the awareness (Miyaji, 2013, Miyaji, 2014b). Characteristics of various student groups have been reported as the results of cluster analysis of awareness related to programming abilities and classification of awareness and students (Miyaji et al., 2014c). It was found that awareness related to abilities improved after the lecture course.

Of all types of awareness, it was found that the awareness related to programming improved better than the awareness related to general abilities. It is necessary to learn whether the awareness related to general abilities did not improve because the assignments were difficult, or whether the awareness related to general abilities does not improve much in programming.

We examined and implemented the media necessary for JavaScript programming lecture course using e-learning as blended lecture. A class was provided by explaining the answers to the previous assignment first, followed by the lecture on the day's grammar and processing details using slides. Materials describing example exercises and exercise problems on the day's contents are then distributed, and explanation on the materials was given using slides. Exercise problems to prepare programs with reference to the grammar, processing details and example exercises were presented later to let the students exercise. JavaScript programming is easier than PHP programming. Exercise problems in JavaScript programming were easier than that in PHP. The students were asked to prepare programs of their own voluntary themes by the 14th week, and evaluate their programs one another in the 14th week as collaborative learning. Then, they were asked to repair the programs with reference the evaluation results by the 15th week and to evaluate their programs one another again at the end of the term.

In this paper, the degree of term recognition and awareness were surveyed in JavaScript programming to learn the degree of improvement. It was also analyzed by significant test whether the term recognition and the awareness related to abilities improved. We report degree of change using their results. Regarding the awareness related to general abilities, we compared JavaScript programming with PHP programming on the degree of improvement. We also show the results.

## 2. Lecture Course Design and Contents

As blended lecture, 15 of 90-minute lectures were given in an optional course on programming in junior-level in faculty of information science at University A. Table 1 shows the details about lecture course and lecture course plan. After 15 lectures, final examination was given to motivate the students and check the degree of their understanding. There were 21 students. Instructions were given from the instructor and TA during practices.

Table 1. Lecture Course Plan for JavaScript Programming

T i m e	Lesson contents	Lecture course									e-learning			
		No. of lesson slides	Distribution document	Textbook	No. of pages of the textbook	An example program and practice	Voluntary problem	Small test	Survey of term recognition	Attitude survey	Downloading	Program practice	Report	Evaluation sheet
1	About a technique to support WWW	32	Manual of the lesson plan	Chp0	14			Small test	Pre	Pre	Lecture course plan, Evaluation sheet			Pre
2	1.1 Basics of HTML, 1.2.1 HTML tag	14		Chp1	7	Exercise 1					Report for exercise	Practice 1	Practice 1	
3	1.4 Tables, 1.5 Frames, 1.6 Forms, 1.7 Receipt of data, 1.8.1.9 Basics of CSS	34		Chp1	12	Exercise 2						Practice 2	Practice 2	
4	1.10-1.13 Various styles	22		Chp1	8	Exercise 3						Practice 3	Practice 3	
5	2.1 Display of the letter in JavaScript	5		Chp2	4	Exercise 4						Practice 4	Practice 4	
6	2.2-2.6 Reference setting of the property and method in JavaScript	20		Chp3	8	Exercise 5						Practice 5	Practice 5	
7	3.1-3.5 Variables, Type of the variable, Four operations	24		Chp3	12	Exercise 6						Practice 6	Practice 6	
8	3.6-3.8 relational operators, logical operator, 4.1.4.2 Branch sentence (if sentence)	23		Chp3.4	14	Exercise 7						Practice 7	Practice 7	
9	4.3-4.6 Repetition sentence (for sentence, while sentence, switch sentence)	18		Chp4	8	Exercise 8						Practice 8	Practice 8	
10	5.1-5.3 Definition and call of the function	13		Chp5	8	Exercise 9						Practice 9	Practice 9	
11	6.1-6.5 Array, Character string, Mathematics function	23		Chp6	12	Exercise 10						Practice 10	Practice 10	
12	7.1-7.6 Window, Document, Form, Element object	49		Chp7	16	Exercise 11	Manual				Manual	Practice 11	Practice 11	
13	7.7-7.12 Location, DOM, and Object of others, Making the program of the problem	32		Chp7	10		Design				Report of voluntary problem			Self evaluation
14	An evaluation and correction of the program of the problem						Program			Voluntary problem				Peer evaluation, Others evaluation
15	The reevaluation of the program of the problem, making the report						Correction		Post	Post		Voluntary problem	Voluntary problem	Peer evaluation, Others evaluation

### *2.1 Purpose and Objective of Lecture Course*

On today's internet, web services are provided by dynamically operating programs such as CGI on the web server and making changes on the webpages. The purpose of this lecture course was to learn the JavaScript language, which is popularly used in CGI, and become proficient in basic operation of the program as well as dynamic homepage preparation.

The following objectives were set: (1) Understanding the relationship between server and client, (2) Understanding the Web services, (3) Learning how to use JavaScript, and (4) Learning how to prepare CGIs.

Activities to research, ponder, create, evaluate and correct were also included in lecture course to improve the problem-solving abilities necessary in society.

### *2.2 Structure of One Lecture*

As the structure of the class, lecture was given for approximately 35 minutes using slides based on the grammar and process details in the textbook (Anku, 2011) for the day. Then materials describing the example exercise program and exercise problems on the contents for the day were distributed. Based on these materials, explanation was given on the program using slides for approximately 10 minutes. Then with reference to the grammar, program examples in example exercise and execution examples, the students were asked to practice by presenting an exercise problem to develop programs (approximately 45 minutes). They were allowed to execute and check the process flow and execution results with reference to program example. Those who successfully develop programs for the exercise problem were asked to submit the program and report file to LMS.

### *2.3 Details of Exercise*

As exercise, 2 problems to develop JavaScript programs related to the chapter discussed in the lecture were given. One problem was similar to the program example which could be developed by making slight changes to the program example. The other was an application problem for it.

The students were instructed to submit at least one program in one class, and to develop programs for both 2 problems if they had time. Students were asked to submit the file of the program and another file which was an A4 form report, when a student finished a program. The information described in the report included the program list, execution results and discussion. The grades were given as overall evaluation on submitted materials including exercise problems and assignments as well as regular tests.

During the 13th class, an exercise to design and develop programs such as card games, horoscope, and arithmetic learning that other people can use by using control statements, array and so forth as voluntary themes was given. The flow for the process is described as follows: Voluntary theme exercise was explained during the 12th class with instruction to make preparations so that the program to prepare through data input can be designed and developed in the next class. During the 13th class, the students (1) developed programs. After distributing the forms to evaluate the others in the 14th class, they (2) executed the developed program, (3) evaluated one another, and (4) corrected the program with reference to it. During the 15th class next week, the students (5) executed the corrected programs again, (6) evaluated one another again, (7) checked if they were corrected properly, and (8) wrote and submitted reports.

### *2.4 Description of E-learning*

The following were prepared and made available as e-learning functions: ① Material downloading (lecture course plan, evaluation sheet, framework for exercise report, form for explanation of voluntary themes, framework for report on voluntary themes), ② Uploading of submitted materials (evaluation sheets, exercise reports, programs, reports on voluntary themes), ③ Bulletin board, ④ Mail.

### *2.5 Details about the Media Used*

The following media were used to give lectures: ① Textbook, ② Documents describing lecture course contents and plan, ③ Slides explaining lecture details and plan, ④ Lecture slides, ⑤ Forms describing program examples and exercise problems, ⑥ Slides describing program examples and exercise problems, ⑦ Documents describing voluntary themes, ⑧ Evaluation sheet file, ⑨ Report form file, ⑩ PC, and ⑪ e-learning.

## **3. Analysis Results**

A survey on degree of term recognition was conducted to give classes on JavaScript programming and know the changes in amount of knowledge, and a survey on awareness on abilities was conducted to know the changes in awareness. The survey data were analyzed using significance test to explain the results. The rate of exercise problem report submission was surveyed. The results on the degree of term recognition, the awareness on

abilities and the rate of exercise problem report submission are explained. They were compared with the results from PHP programming lecture course.

Hereafter, a null hypothesis is that two values to compare are equal. An alternative hypothesis is that two values to compare are different. It is considered that there is significant different with significance level 5% in the significance test results. Symbols  $\bar{m}$ , SD,  $t$ , and  $p$  indicate average, standard deviation, test static and the  $p$  value, respectively. In addition, significance levels 0.1%, 1%, 5%, and 10% are indicated as \*\*\*, \*\*, \*, and +, respectively.

### *3.1 Results of Survey on Degree of Term Recognition in JavaScript*

Survey on degree of recognition on 72 terms [6] was conducted twice before the lecture course (Week 1) and after the lecture course (Week 15). These 72 terms were selected among the index as important terms appearing in the textbook related to the JavaScript programming lecture course.

The degree of term recognition was evaluated in three stages of “1. Don’t know the term, 2. Don’t know in details but heard of it, and 3. We know the term.” The average degree of term recognition before and after the lecture course  $m_{pre}$  and  $m_{post}$  were 1.49 and 2.29, respectively. A student attending a lecture is 21 people. Three students lack response in either pre or post surveys. Analyzation is done for 18 students that replied in both pre and post surveys in the following.

The result of paired  $t$ -test on the degree of recognition of 72 terms before and after the lecture course showed significant difference as shown in lowest row of Table 2. As the overall degree of recognition increased significantly after the lecture course, it was indicated that the overall amount of students’ knowledge on JavaScript programming increased after the lecture course.

The result of paired  $t$ -test on degrees of the recognition of each term before and after the lecture course showed significant difference between values before and after the lecture course for the following 68 terms as shown in Table 2. The amount of knowledge increased after the lecture course for these 68 terms. There was tendency for significant difference for 3 terms: 23, 28 and 33. There was no significant difference for only “6. Link”. Based on these, it was found that the amount of knowledge tended to increase after the lecture course for these 3 terms. The rate at which significant difference or tendency for significant difference was observed was 98.6%, and there was improvement in degree of recognition in almost all terms.

Significant difference was not observed only for one term “Link”. The average degree of recognition for this “link” was 2.7 before the lecture course, which was highest and well-known. Though it reached 3.0 after the lecture course, the growth was small and therefore there was no significant difference.

Table 2. Significance test on degree of recognition on terms related to lecture course in JavaScript and PHP

JavaScript										PHP									
No.	Technical term	Pre		Post		Growth		t test		No	Technical term	Pre		Post		Growth		t-test	
		m	SD	m	SD	m	SD	t	p			m	SD	m	SD	m	SD	t	p
1	Hyperlink	2.0	0.6	2.3	0.7	0.4	0.6	2.1	*	1	Apache	1.2	0.5	1.9	0.8	0.8	0.9	4.1	***
2	HTML tag	2.5	0.7	2.9	0.6	0.6	0.8	2.6	*	2	array	1.4	0.7	2.6	0.6	1.2	0.8	7.5	***
3	Style sheet	1.5	0.6	2.9	0.2	1.5	0.5	8.3	***	3	break	2.0	0.9	2.6	0.6	0.7	1.0	3.3	**
4	CSS	1.7	0.6	2.9	0.2	1.4	0.7	6.6	***	4	case	1.8	0.8	2.4	0.6	0.6	0.8	3.7	**
5	CGI	1.6	0.5	2.4	0.5	0.8	0.5	4.1	***	5	CGI	1.7	0.8	2.1	0.6	0.4	1.1	1.7	+
6	Link	2.7	0.5	3.0	0.5	0.3	0.5	1.7		6	CHECKBOX	1.4	0.8	2.5	0.5	1.1	0.8	6.3	***
7	Frame	2.2	0.7	2.8	0.3	0.8	0.8	3.4	**	7	chop	1.2	0.4	1.8	0.7	0.6	0.9	3.5	**
8	Form	2.0	0.6	2.7	0.4	0.8	0.6	4.2	***	8	continue	1.6	0.7	2.2	0.7	0.6	1.0	2.8	**
9	GET	1.4	0.6	2.5	0.5	1.1	0.6	5.3	***	9	date	2.0	0.9	2.5	0.7	0.4	1.1	2.0	+
10	POST	1.6	0.6	2.4	0.6	0.7	0.7	3.3	**	10	define	2.4	0.7	2.6	0.6	0.3	0.9	1.5	
11	hover pseudo-class	1.2	0.4	2.2	0.6	0.9	0.6	4.7	***	11	do~while	2.6	0.6	2.7	0.6	0.1	1.0	0.6	
12	active pseudo-class	1.2	0.4	2.3	0.6	1.1	0.6	5.3	***	12	else	2.8	0.5	2.8	0.6	0.0	0.8	0.0	
13	Event handler	1.0	0.2	2.5	0.6	1.4	0.8	6.4	***	13	elseif	2.1	0.9	2.7	0.6	0.6	1.1	2.7	*
14	Object	2.1	0.5	2.7	0.6	0.6	0.7	3.1	**	14	endforeach	1.1	0.3	1.8	0.7	0.7	0.8	4.5	***
15	Property	2.2	0.6	2.8	0.4	0.6	0.5	3.4	**	15	endif	1.2	0.5	2.0	0.8	0.8	0.9	4.1	***
16	Method	1.7	0.6	2.5	0.5	0.9	0.5	5.1	***	16	exit	1.7	0.7	2.2	0.8	0.6	0.8	3.4	**
17	Reading of the outside JS file	1.1	0.3	2.7	0.5	1.5	0.5	8.7	***	17	fclose	1.0	0.2	2.1	0.9	1.0	1.0	5.3	***
18	Type of the variable	2.2	0.8	2.8	0.4	0.6	0.8	2.8	*	18	feof	1.0	0.2	1.9	0.8	0.9	0.9	5.0	***
19	Escape sequence	1.3	0.5	2.5	0.5	1.1	0.7	5.4	***	19	fgets	1.0	0.2	1.9	0.9	0.9	0.9	5.0	***
20	Conversion of the type	1.8	0.7	2.5	0.6	0.7	0.7	3.3	**	20	file exists	1.2	0.5	1.8	0.8	0.6	0.8	3.7	**
21	Onditional expression	2.5	0.6	2.8	0.6	0.4	0.6	2.1	*	21	filesize	1.5	0.7	2.1	0.8	0.6	0.9	3.3	**
22	Logical operator	2.6	0.5	2.9	0.4	0.4	0.6	2.1	*	22	float	1.4	0.7	2.2	0.8	0.8	1.1	3.7	**
23	Priority of the operator	2.6	0.6	2.9	0.2	0.4	0.6	1.8	+	23	fopen	1.1	0.4	2.0	0.8	1.0	0.8	5.7	***
24	Control sentence	2.3	0.6	2.9	0.3	0.6	0.7	2.8	*	24	for	2.6	0.6	2.8	0.5	0.2	0.7	1.7	
25	Method to define the function	2.2	0.7	2.8	0.4	0.8	0.8	3.7	**	25	foreach	1.0	0.2	2.2	0.8	1.2	0.8	7.3	***
26	Call of the function	2.4	0.6	2.8	0.4	0.6	0.7	3.1	**	26	form	1.6	0.7	2.4	0.7	0.8	1.2	3.5	**
27	Call by value and call by reference	1.9	0.7	2.8	0.4	1.1	0.7	5.0	***	27	fputs	1.0	0.2	2.2	0.8	1.1	0.8	6.7	***
28	Return value	2.4	0.6	2.7	0.5	0.4	0.8	1.9	+	28	FTP	1.8	0.8	2.4	0.7	0.6	0.9	3.1	**
29	Local variable	1.5	0.7	2.5	0.6	1.2	0.7	5.6	***	29	function	1.9	0.8	2.6	0.6	0.7	0.8	4.0	***
30	Global variable	1.5	0.7	2.4	0.7	0.9	0.9	4.0	***	30	GET	1.6	0.6	2.2	0.7	0.6	0.7	3.9	***
31	Array object	1.3	0.6	2.5	0.7	1.3	0.7	6.0	***	31	global	1.5	0.6	2.1	0.7	0.6	0.9	3.4	**
32	Associative array	1.9	0.6	2.4	0.7	0.5	0.7	2.5	*	32	GUI	1.5	0.7	2.1	0.7	0.6	1.0	2.9	**
33	Two-dimensional array	2.0	0.7	2.4	0.7	0.4	0.8	1.8	+	33	HTML	2.6	0.6	2.8	0.4	0.2	0.7	1.3	
34	slice() method	1.1	0.3	2.0	0.8	0.9	0.8	4.1	***	34	HTTP	2.4	0.7	2.7	0.4	0.4	0.8	2.1	*
35	unshift() method	1.1	0.3	1.9	0.8	0.8	0.8	3.7	**	35	if	2.7	0.5	2.8	0.4	0.2	0.7	1.2	
36	pop() method	1.1	0.5	2.0	0.8	0.9	0.9	4.0	***	36	include	2.3	0.8	2.5	0.7	0.2	0.8	1.2	
37	shift() method	1.1	0.5	1.9	0.8	0.8	0.8	3.7	**	37	MySQL	1.3	0.7	2.1	0.8	0.8	0.9	4.2	***
38	string object	1.0	0.2	2.1	0.8	1.0	0.8	4.4	***	38	NULL	2.0	0.8	2.6	0.6	0.6	0.9	3.2	**
39	charAt() method	1.0	0.2	1.8	0.8	0.8	0.7	3.6	**	39	print	2.4	0.8	2.8	0.4	0.4	0.9	2.3	*
40	split() method	1.0	0.2	1.8	0.7	0.8	0.7	3.6	**	40	require	1.1	0.3	2.1	0.7	1.0	0.8	6.2	***
41	indexOf() method	1.0	0.2	1.9	0.7	0.9	0.8	4.1	***	41	return	2.6	0.7	2.6	0.6	0.0	0.9	0.0	
42	Math object	1.0	0.2	2.1	0.7	1.0	0.7	4.8	***	42	round	1.4	0.6	2.2	0.8	0.8	0.9	4.5	***
43	Date object	1.0	0.2	2.1	0.7	1.1	0.7	5.0	***	43	SELECT	1.8	0.8	2.5	0.6	0.7	0.9	3.8	***
44	Meta letter	1.5	0.7	2.3	0.7	0.8	0.8	3.4	**	44	STDIN	1.0	0.2	1.8	0.6	0.8	0.6	6.2	***
45	Character class	1.4	0.6	2.3	0.7	1.1	0.7	5.0	***	45	STDOUT	1.1	0.3	1.8	0.6	0.8	0.6	6.4	***
46	Grouping	1.4	0.5	2.0	0.7	0.6	0.7	3.1	**	46	SUBMIT	1.3	0.6	2.1	0.7	0.8	0.7	5.2	***
47	RegExp object	1.1	0.3	1.6	0.7	0.6	0.7	2.8	*	47	switch	1.8	0.7	2.5	0.6	0.7	1.0	3.4	**
48	replace() method	1.0	0.2	1.7	0.7	0.6	0.8	3.0	**	48	TEXT	1.8	0.8	2.3	0.7	0.5	1.0	2.5	*
49	Window object	1.1	0.5	2.2	0.7	1.1	0.6	5.3	***	49	URL	2.7	0.7	2.8	0.5	0.1	0.9	0.6	
50	open() method	1.3	0.5	2.4	0.7	1.1	0.8	5.1	***	50	VALUE	1.8	0.8	2.5	0.6	0.8	1.0	3.8	**
51	close() method	1.3	0.5	2.3	0.7	1.0	0.7	4.8	***	51	Web server	2.4	0.8	2.8	0.4	0.4	0.9	2.1	*
52	confirm() method	1.1	0.3	1.9	0.7	0.8	0.7	3.9	**	52	WWW	2.8	0.5	2.8	0.5	0.1	0.7	0.5	
53	prompt() method	1.0	0.2	1.9	0.7	0.8	0.7	3.9	**	53	XAMPP	1.1	0.3	2.8	0.4	1.8	0.6	14.7	***
54	setInterval() method	1.0	0.2	1.9	0.7	0.8	0.7	3.9	**	54	Regular expression	1.2	0.4	2.5	0.6	1.3	0.6	10.5	***
55	Document object	1.2	0.6	2.4	0.7	1.2	0.7	5.8	***	55	Session	1.4	0.6	2.4	0.6	1.0	0.8	6.1	***
56	Form object	1.2	0.5	2.5	0.6	1.3	0.6	6.9	***	56	Transmission button	2.0	0.7	2.7	0.5	0.7	1.1	3.3	**
57	Element object	1.1	0.5	2.2	0.6	1.1	0.6	5.9	***	57	Here document	1.2	0.4	1.9	0.7	0.7	0.7	4.9	***
58	Select object	1.1	0.3	2.1	0.6	1.0	0.6	5.2	***	58	File handle	1.2	0.4	2.0	0.8	0.9	0.8	5.3	***
59	Option object	1.0	0.2	2.0	0.7	0.9	0.7	4.4	***	59	Radio button	1.4	0.7	2.8	0.4	1.4	0.8	8.8	***
60	Location object	1.0	0.2	2.1	0.7	1.0	0.7	4.8	***	60	Associative array	1.1	0.3	2.0	0.8	0.9	0.9	5.1	***
61	DOM	1.1	0.3	1.8	0.7	0.6	0.7	3.1	**	Average		1.7	0.8	2.4	0.7	0.7	1.0	3.5	**
62	Event object	1.0	0.2	2.1	0.6	1.0	0.7	4.8	***	*** p<.001, ** p<.01, * p<.05, + p<.1									
63	Style object	1.0	0.2	2.1	0.6	1.0	0.7	4.8	***										
64	History object	1.0	0.2	1.9	0.7	0.9	0.7	4.3	***										
65	Image object	1.0	0.2	2.1	0.7	1.0	0.8	4.6	***										
66	Anchor object	1.0	0.2	1.6	0.7	0.6	0.6	3.0	**										
67	Navigator object	1.0	0.2	1.6	0.6	0.6	0.6	3.0	**										
68	Ajax	1.1	0.3	1.8	0.7	0.6	0.8	3.0	**										
69	XML	1.8	0.5	2.1	0.7	0.4	0.5	2.3	*										
70	XHTML	1.8	0.5	2.2	0.6	0.6	0.5	3.4	**										
71	DynamicHTML	1.2	0.4	1.7	0.6	0.5	0.5	3.0	**										
72	XMLHttpRequest object	1.1	0.3	1.6	0.6	0.5	0.5	2.7	*										
Average in total		1.5	0.5	2.3	0.6	0.8	0.3	5.7	***										

### 3.2 Comparison between Degrees of Recognition for Common Term in Two Kinds of Classes

The class of PHP programming (25 respondents) was carried out. The effect for the class is reported (Miyaji et al., 2014a, Miyaji et al., 2014c). The awareness related to a programming in the report was prolonged as a whole. But it was stated that general awareness did not lengthen very much. As the reason it is presumed that this is caused by the fact that the issue of practice was difficult.

It was projected that the results of the term recognition in the class of PHP programming were compared with that in the JavaScript programming to confirm it. The class in the JavaScript programming was designed to present easier problems than that in class in the PHP programming. The common terms in two kinds of classes were examined for the term recognition investigation. They were seven items as shown in Table 3. Table 3 shows average and standard deviation for recognition degrees of these seven common terms. The results of t-test about the elongation that is the difference are shown to the right of Table 3.

It was admitted that the prior rating level of the whole of PHP was significantly higher than JavaScript. In addition, it was admitted that the overall elongation of the JavaScript was significantly higher than PHP.

T-test was conducted about the recognition of each seven terms. It was admitted that the prior rating level of two items of date and SELECT in PHP was significantly higher than JavaScript. It was admitted that the prior rating level of the associative array in JavaScript was significantly higher than that in PHP. The subsequent rating level of Date object and Select object in JavaScript was the tendency that was lower than SELECT in PHP. It was admitted that the growth of date in the JavaScript was significantly higher than PHP. The tendency that the growth of GET in the JavaScript was significantly higher than that in PHP was recognized.

The prior rating level of the PHP tends to be higher in the whole than that of JavaScript. The difference was not seen in post rating level. The elongation of JavaScript was the tendency which was higher in the whole than that in PHP. It is thought that the elongation that is a difference between the prior rating level and the post rating level appeared greatly, because the prior rating level of JavaScript is lower in the whole than that of PHP as one of the causes. Even though there is difference between JavaScript and PHP in prior rating level, it may be said that quantity of knowledge in the JavaScript increases more generally than that in PHP.

Table 3. Results of significant difference about the recognition degrees of the common terms for two kinds of classes

JavaScript									PHP												Comparison between JavaScript and PHP					
No.	Technical terms	Pre		Post		Elongation		t-test		No.	Technical terms	Pre		Post		Elongation		t-test		Pre		Post		Elongation		
		m	SD	m	SD	m	SD	t	p			m	SD	m	SD	m	SD	t	p	t	p	t	p	t	p	
5	CGI	1.6	0.5	2.4	0.5	0.8	0.5	4.1	***	5	CGI	1.7	0.8	2.1	0.6	0.4	1.1	1.7	+	0.2		1.5		1.2		
43	Date object	1.0	0.2	2.1	0.7	1.1	0.7	5.0	***	9	date	2.0	0.9	2.5	0.7	0.4	1.1	2.0	+	4.5	***	1.6		2.0	*	
24	Control statement	2.3	0.6	2.9	0.3	0.6	0.7	2.8	*	24	for statement	2.6	0.6	2.8	0.5	0.2	0.7	1.7		1.4		0.3		1.5		
9	GET	1.4	0.6	2.5	0.5	1.1	0.6	5.3	***	30	GET	1.6	0.6	2.2	0.7	0.6	0.7	3.9	***	0.8		1.5		2.0	+	
21	Codition equation	2.5	0.6	2.8	0.6	0.4	0.6	2.1	*	35	if statement	2.7	0.5	2.8	0.4	0.2	0.7	1.2		1.2		0.5		1.2		
58	Select object	1.1	0.3	2.1	0.6	1.0	0.6	5.2	***	43	SELECT	1.8	0.8	2.5	0.6	0.7	0.9	3.8	***	3.1	**	1.9	+	1.1		
32	Associative array	1.9	0.6	2.4	0.7	0.5	0.7	2.5	*	60	Associative array	1.1	0.3	2.0	0.8	0.9	0.9	5.1	***	4.9	***	1.3		1.5		
Average		1.7	0.7	2.4	0.6	0.8	0.7	3.7	**	Average		1.9	0.9	2.4	0.7	0.5	0.9	2.5	*	3.0	**	0.2		3.0	**	

\*\*\* p<.001, \*\* p<.01, \* p<.05, + p<.1

### 3.3 Results of Survey on Awareness Related to Abilities in JavaScript

For 55 items of awareness related to abilities, survey was conducted for before lecture course during the 1st class and after lecture course during the 15th class in JavaScript. The evaluation was made in 9 stages of "1. None, 3.Slightly aware, 5.Somewhat aware, 7.Highly aware, 9.Extremely aware." The number of students responding to both surveys before and after lecture course was 18. For all of the 55 items, the average evaluation values before and after the lecture course were 4.3 and 5.4, respectively. The result of paired t-test on evaluation values before and after the lecture course showed significant difference for all of the 55 items as shown in the bottom line of Table 4. Based on these, it was indicated that the overall awareness related to abilities improved after the lecture course.

The paired t-test on evaluation values before and after the lecture course in JavaScript was conducted for each item of awareness related to abilities. There was significant difference between values before and after the lecture course for 40 items as shown in JavaScript column of Table 4 as a result of t-test on evaluation values for each of the awareness items related to abilities. Based on these, it was found that the awareness for these 40 items improved after the lecture course. Tendency for significant difference was observed for the following 4



items: (2) Understanding of computers, (8) Ability to study by oneself, ability to learn, (21) Ability to pursue matters deeply, ability to explore matters, (24) Sense of accomplishment, sense of satisfaction. Based on these, it was indicated that awareness for these 4 items tended to increase after the lecture course.

There were 44 items of awareness related to abilities with significant difference or tendency for significant difference, indicating that awareness for approximately 69% of the 55 items improved after the lecture course. Neither significant difference nor tendency for significant difference was observed for only 11 awareness items.

Table 4. Significance test on awareness related to abilities

Evaluation item	JavaScript						PHP						Comparison		
	Pre		Post		Growth		Pre		Post		Growth		Pre	Post	Growth
	m	SD	m	SD	m	SD	t	p	m	SD	m	SD	t	p	t
(1) Interest in and curiosity about computers	6.7	0.9	6.6	1.7	-0.1	1.4	0.3		7.0	2.0	6.0	2.2	-0.6	2.9	1.0
(2) Understanding of computers	4.7	1.4	5.2	1.5	0.6	1.2	2.0	+	4.4	1.6	5.1	1.7	1.0	2.2	2.3
(3) Computer operation skills	4.3	1.5	5.2	1.4	0.8	1.1	3.2	**	4.5	1.7	5.5	1.8	1.3	2.3	2.7
(4) Computer usage methods and broadening of situations	5.1	1.8	5.1	1.8	0.1	1.5	0.1		5.1	1.9	5.8	2.1	1.0	2.5	1.8
(5) Ability to set challenges, ability to discover problems	4.3	1.7	5.2	1.5	0.8	1.3	2.6	**	3.9	2.2	4.8	1.7	1.1	2.5	2.2
(6) Ability to plan, to do things in a planned manner	4.4	1.6	4.9	1.6	0.5	1.5	1.3		3.9	1.8	4.5	2.1	0.9	2.5	1.7
(7) Cultivation of understanding of knowledge learned	4.5	1.4	5.4	1.5	0.9	1.4	2.7	**	4.2	1.4	4.6	2.0	0.6	2.3	1.3
(8) Ability to study by oneself, ability to learn	5.2	1.9	5.9	1.5	0.7	1.6	1.9	+	4.3	1.8	4.9	1.7	0.9	2.1	2.0
(9) Ability to gather information, ability to conduct research	5.4	1.9	5.8	1.7	0.4	2.2	0.7		4.7	2.0	5.0	2.0	0.6	2.4	1.3
(10) Ability to sort through related information or data	4.8	1.7	5.3	1.9	0.6	2.1	1.1		4.5	1.5	4.8	1.7	0.7	2.1	1.5
(11) Ability to analyse information	3.9	1.4	5.2	1.7	1.3	1.4	3.9	**	4.2	1.9	4.8	1.7	0.8	2.0	1.9
(12) Ability to express thoughts in writing	3.4	1.3	4.7	1.8	1.3	1.3	4.0	***	4.2	2.1	4.6	2.1	0.7	2.5	1.3
(13) Ability to express thoughts through media other than writing	3.6	1.5	4.7	2.1	1.1	1.5	2.8	**	4.7	1.9	4.9	1.9	0.5	2.5	0.9
(14) Ability to talk to and explain to others comprehensively	3.5	1.3	4.6	2.0	1.1	1.6	2.8	**	4.4	2.0	4.2	1.8	0.1	2.1	0.2
(15) Ability to make presentations	3.8	2.0	4.3	1.8	0.5	2.3	0.9		4.0	1.9	4.2	1.8	0.5	2.7	0.8
(16) Ability to listen to others and to ask questions to others	4.2	1.5	5.3	1.4	1.1	1.4	3.1	**	4.3	1.8	5.3	2.1	1.2	2.5	2.4
(17) Communication ability	4.2	1.6	5.2	1.7	1.1	1.2	3.6	**	4.3	2.4	4.7	2.1	0.6	3.3	0.9
(18) Ability to appropriately self-evaluate one's thoughts	4.9	1.8	5.8	1.8	0.9	1.5	2.5	*	4.2	1.7	4.7	2.1	0.7	2.2	1.5
(19) Ability to appropriately evaluate other people's thoughts	5.0	1.8	6.1	1.6	1.1	1.5	2.8	**	5.1	1.9	4.9	1.6	0.2	2.4	0.3
(20) Ability to correct and improve on one's own thoughts	4.6	1.4	5.9	1.5	1.3	1.5	3.7	**	4.6	1.8	4.5	1.9	0.2	2.5	0.4
(21) Ability to pursue matters deeply, ability to explore matters	4.9	1.8	5.6	1.5	0.7	1.6	1.9	+	4.5	1.5	4.9	1.9	0.7	2.1	1.6
(22) Ability to execute, ability to practice, ability to put into action	4.7	1.9	5.6	1.3	0.8	1.5	2.2	*	4.6	1.5	4.8	1.9	0.5	2.4	0.9
(23) Ability to cooperate and to learn concertedly	4.8	1.7	5.9	1.4	1.1	1.8	2.6	**	5.2	1.4	4.7	1.6	-0.1	2.2	0.3
(24) Sense of accomplishment, sense of satisfaction	5.4	1.9	6.1	1.4	0.7	1.5	1.9	+	5.2	2.0	4.8	2.0	-0.1	3.3	0.1
(25) Sense of fulfilment, sense of achievement	5.2	1.7	5.9	1.5	0.7	1.6	1.7		5.2	1.9	5.0	2.0	0.2	3.2	0.3
(26) Ability to solve problems	4.7	1.4	5.3	1.7	0.6	1.8	1.4		4.5	1.8	4.9	1.7	0.7	2.0	1.7
(27) Ability to construct and create knowledge	4.3	1.5	5.3	1.6	1.1	1.9	2.3	*	4.0	1.7	5.0	1.9	1.2	2.5	2.3
(28) Ability to think, consider and come up with ideas by oneself	4.5	1.8	5.3	1.5	0.8	2.0	1.6		5.0	2.1	5.1	1.9	0.5	3.0	0.7
(29) Creativity/ability to create	4.3	1.7	4.9	1.8	0.7	2.2	1.3		4.6	1.8	4.9	2.1	0.6	2.5	1.1
(30) Interest in and curiosity about this field	5.8	1.7	5.8	1.5	0.1	1.7	0.1		5.7	1.8	5.2	1.8	-0.1	2.5	0.1
(31) Interest in programming	6.3	1.5	5.9	1.8	-0.4	1.7	1.1		6.2	1.8	5.5	2.2	-0.2	2.9	0.4
(32) Knowledge of programming	3.8	1.4	5.3	1.5	1.5	1.8	3.4	**	4.2	1.6	4.7	1.7	0.8	1.6	2.4
(33) Desire to learn about programming	5.8	1.5	5.6	1.7	-0.2	1.9	0.4		5.8	1.5	5.3	2.2	-0.1	2.6	0.2
(34) Desire to try problems	5.6	1.9	5.8	1.7	0.2	2.3	0.4		5.6	1.9	5.0	1.7	-0.3	2.8	0.5
(35) Ability to think about a problem in stages	4.9	1.5	5.9	1.7	1.0	1.5	2.8	**	4.5	1.7	4.8	1.7	0.6	2.3	1.2
(36) Ability to express an idea as an algorithm	3.8	2.0	5.0	1.7	1.2	1.8	2.6	**	3.6	1.9	4.6	1.8	1.2	2.2	2.7
(37) Ability to think about algorithms	3.4	1.8	5.0	1.9	1.6	1.9	3.5	**	3.7	1.9	4.5	1.6	1.1	2.4	2.1
(38) Ability to review the flow of an algorithm	3.3	1.9	5.2	1.9	1.9	1.7	4.6	***	3.6	1.8	4.5	1.6	1.1	2.1	2.6
(39) Ability to improve algorithms	3.3	1.8	4.8	1.8	1.5	1.9	3.2	**	3.3	1.9	4.8	1.8	1.7	2.3	3.6
(40) Ability to express ideas with JavaScript	2.3	1.5	5.5	1.8	3.3	2.4	5.5	***	2.6	1.9	5.1	1.9	2.6	2.7	4.5
(41) Ability to debug JavaScript programs	2.1	1.5	5.2	2.0	3.1	2.3	5.4	***	2.2	1.5	4.8	1.9	2.6	2.5	5.1
(42) Ability to configure test data	2.7	2.0	4.6	1.9	1.9	1.9	4.2	***	2.5	1.9	5.0	1.8	2.7	2.2	5.7
(43) Ability to work to improve a program	3.3	1.7	5.3	1.4	2.0	2.0	4.0	***	4.0	2.2	5.2	1.7	1.4	2.7	2.5
(44) Ability to write reports about programs	3.0	1.5	5.4	1.4	2.4	1.6	6.0	***	3.3	2.2	4.5	1.6	1.4	2.3	2.8
(45) Ability to understand other people's ideas	3.9	1.6	5.9	1.5	2.0	2.0	4.2	***	4.2	1.6	4.8	1.7	0.9	2.3	1.8
(46) Ability to read other student's programs	3.4	2.0	5.6	1.4	2.3	2.1	4.5	***	3.5	1.9	4.9	1.7	1.6	2.4	3.1
(47) Ability to read other people's reports	3.7	1.5	5.8	1.2	2.1	1.6	5.6	***	3.5	2.2	5.2	1.8	1.9	2.8	3.3
(48) Ability to express personal ideas using a computer	3.0	1.9	5.2	1.6	2.2	1.9	4.7	***	3.4	1.8	4.9	1.5	1.7	2.2	3.7
(49) Ability to collaborate on problems	4.7	1.9	5.5	1.2	0.8	2.2	1.5		5.0	1.7	5.0	1.8	0.3	2.5	0.7
(50) Ability to learn for a problem positively	5.1	2.1	5.5	1.6	0.4	2.1	0.7		4.9	1.7	5.1	1.6	0.6	2.0	1.3
(51) Ability to keep working on a problem until it is finished	5.4	2.2	5.9	0.9	0.4	2.3	0.8		5.4	1.9	5.1	1.8	0.1	2.6	0.2
(52) Knowledge of JavaScript syntax	2.2	1.5	5.3	1.5	3.1	1.7	7.3	***	2.9	1.8	5.0	1.9	2.3	2.6	4.2
(53) Knowledge for running JavaScript	2.1	1.5	5.3	1.3	3.2	1.7	7.5	***	3.6	1.6	4.7	1.4	1.3	2.1	3.0
(54) Knowledge of correcting program errors	2.7	1.9	5.2	1.5	2.5	2.1	4.9	***	3.4	1.9	5.1	1.6	1.9	2.4	3.7
(55) Knowledge of programming techniques	2.7	1.9	5.0	1.3	2.3	1.6	6.2	***	3.0	1.9	4.8	1.6	2.0	2.1	4.6
Average in total	4.3	1.7	5.4	1.6	1.2	1.1	4.6	***	4.3	2.0	4.9	1.9	0.9	1.8	2.3

\*\*\* p<.001, \*\* p<.01, \* p<.05, + p<.1

### 3.4 Comparison of Awareness Related to Abilities Regarding Programming Lecture Courses for Two Kinds of Classes

The lecture course in this report was on programming with Java Script which was given in Fiscal 2014. Class in PHP programming was given in Fiscal 2011 and the effects of the class have been reported (Miyaji et al. 2014a, Miyaji et al. 2014c). Of these, it was described that the general awareness did not improve much while the

awareness related to programming improved, and it was considered that one of the reasons might have been because the exercise problems were too difficult.

Therefore in this study, which was the same programming class even though the language was Java Script, to check this was decided by making the exercise problems easier. Here, the description Java Script was replaced with PHP in evaluation items. The change in awareness on programming by PHP is shown at the center of Table 4. The changes in awareness on programming by PHP and the changes in awareness in this report are compared on the right of Table 4. The numbers of students responding to the surveys on lecture courses in PHP and JavaScript programming were 23 and 18, respectively.

The result of significance test on average evaluation values in each lecture course before the lecture course, after the lecture course and growth (after – before) showed that the number of items with significant difference (for general awareness and awareness related to programming) in PHP and JavaScript programming were (5, 17) and (15, 19), respectively. Based on these, it is assumed that the awareness improved in more general items in JavaScript programming after the lecture course. In addition, it seems that there was little difference in awareness related to programming.

Next, t-test between average evaluation values from PHP and JavaScript before the lecture course, after the lecture course and growth (after – before) was conducted. As a consequence, significant difference was observed only for Item (53) before the lecture course. Tendency for significant difference was observed for Item (13). For average evaluation values after the lecture course, there was significant difference for Items (19), (20), (23), (24), (35), and (45). There was tendency for significant difference in Items (8), (18), and (44). Regarding average evaluation value for growth, significant difference was observed only for Item (53). Tendency for significant difference was observed for Item (23).

Based on this, it was found that there was little difference in the growth of average evaluation values between PHP and JavaScript programming. Items with significant difference or tendency for significant difference were “(53) Knowledge to execute Java Script (PHP)” and “(23) Ability to cooperate and to learn cooperatively.” For Item (53), nearly all students were able to complete the program for exercise problems using JavaScript as the problems are relatively easier compared to PHP. It is therefore assumed that they felt that Item “(53)” in JavaScript improved better. In addition, it is assumed that Item (23) improved as students did voluntary themes, executed, evaluated one another’s work, and also stated comments. The students seemed especially to feel more strongly that mutual evaluation helped to improve awareness since they were able to complete nearly all programs in JavaScript programming.

### *3.5 Comparison of Exercise Problem Submission Rate*

The students were asked to develop programs as exercise in PHP and JavaScript programming. Details about the problems and submission rates are shown in Table 5. The submission rate was calculated by dividing the number of students who submitted the exercises by the total number of students. In class, they were asked to submit at least one of the two problems. In both classes, Exercise 2 was slightly more difficult and the rate of submission tended to be lower. As their own voluntary themes, the students developed programs with contents they decided on using what they had learned until then. In PHP lecture course, voluntary theme assignments were given twice. It was given once in JavaScript programming. Therefore, Exercise Number 12 is blank for JavaScript programming in Table 4. Exercise 1 and 2 with voluntary themes with Exercise Numbers 12 and 13 indicate first submission and re-submission after correction.

The result of unpaired t-test on average submission rates in PHP and JavaScript programming showed that the t value for Exercise 1 was  $t=2.1$ , showing tendency for significant difference with significance level 10%. For Exercise 2, it was  $t=0.8$ , showing no significant difference with significance level 5%. In addition, when the maximum value and minimum value for submission rates in two classes are compared, there was difference in the minimum value for the problem in Exercise 1 while there was little difference in maximum value. The minimum value for PHP programming was smaller than that of JavaScript programming class. In questionnaire for PHP programming class, the cause of the evaluation being low seems to be due to the low submission rate for the first exercise problem. The average submission rate was also low in Exercise 2 of JavaScript programming class, with the minimum value for submission rate being the smallest. However, the evaluation in class questionnaire was not low.



Table 5. Comparison of submission rate on exercise problems in PHP and JavaScript programming

No.	JavaScript			PHP		
	Problem contents	Practice 1	Practice 2	Problem contents	Practice 1	Practice 2
1	Display of the character string by the HTML	0.86	0.62	Display of the character string by PHP	0.96	0.96
2	Use of the CSS	0.71	0.38	Outbreak of the random number and replace of the numerical value	0.88	0.80
3	Class designation and reading an outside style sheet	0.90	0.52	if sentence, else sentence and branch by the discriminant	0.96	0.80
4	Display of letter and image by a JavaScript	0.95	0.67	Display of the sum of n data	0.48	0.28
5	Property of the Window object and the warning dialogue	0.95	0.81	Display of average, total, the highest score, and the lowest score for five subjects	0.88	0.36
6	Outbreak of the random number, four operations, use of document.write	0.86	0.48	Display of outbreak of n random numbers and product less than 2500	0.72	0.64
7	Display of bigger value after comparing two integer values	0.95	0.90	Display of products for two numerical values using a function	0.68	0.48
8	Display of the sum for n data	0.95	0.76	for sentence, if sentence, judgment of true or false for the phone number memorized in array	0.72	0.72
9	for sentence, if sentence, a definition and the use of the function and stop of the repetition	0.86	0.43	Display by inserting contents memorized in array into a sentence	0.80	0.36
10	for sentence and indication of the contents memorized in array	0.81	0.14	Reading and writing of the file, while sentence and judgment of true or false for the addition of the arithmetic	0.88	0.88
11	Display of an answer by the radio button into text box	0.95	0.29	Making of a database and the table	0.76	0.76
12	—	—	—	Voluntary problem	0.92	0.96
13	Voluntary problem	0.95	0.81	Voluntary problem	0.96	0.56
m	Average	0.89	0.57	Average	0.82	0.66
SD	Standard deviation	0.07	0.22	Standard deviation	0.14	0.22
max	Maximum	0.95	0.90	Maximum	0.96	0.96
min	Minimum	0.71	0.14	Minimum	0.48	0.28

## 4. Discussion

### 4.1 About Change of the Term Recognition and Its Reason

The paired t-test has been performed about the whole growth of the recognition of 72 terms in the JavaScript. As the result, post degree of term recognition became significantly high in total. It was found that quantity of knowledge of the student for the programming increased as a whole after the class. Next, the paired t-test was conducted for the pre and the post degree of each term recognition. The ratio that significant difference or a tendency to significant difference was recognized was 99%.

Recognition degree of other almost all terms improved except one term. Recognition degree of only "6. link" was not recognized. An experience of hearing a lecture, hearing the explanation of the exercise program, creating programs and evaluating programs each other brought to know many terms and to understand them.

Significant difference for one term "6. link" in the class of JavaScript was not admitted. Students have already learned this term in the class of C language before this class. As the prior average recognition degree of this term was 2.7, this means that students understood it to some extent. Because the average recognition degree is almost nearly 3 beforehand in this way and there was little lengthening room, we believe that significant difference was not shown.

Paired t-test was conducted about growth of the recognition of all 60 terms in the class of PHP. It was recognized that post degree of term recognition became significantly high in total. As a whole, quantity of knowledge about the programming increased after the class. As the results of paired t-test about growth of the recognition of each term, the ratio that significant difference or a tendency to significant difference was recognized was 83.3%. Recognition about ten terms did not improve.

Since the ratio of the items with significant difference in the class of JavaScript is larger than that in the class of PHP, low degree of difficulty of the issue of practice seems to improve the term recognition. The quantity of knowledge about the programming increases so that difficulty of the issue of practice is low.

Unpaired t-test about the recognition of seven common terms in both classes of PHP and JavaScript was conducted. The growth of the recognition in the JavaScript was the higher tendency in the whole than that in PHP. It may be said that quantity of knowledge in the class of JavaScript increases generally than that in PHP. This may be because the practice problem having a low degree of difficulty is easy to acquire knowledge as the reason.

## 4.2 About Changes in Students' Awareness toward their Abilities and Reasons for these Changes

### 4.2.1 Awareness toward General Abilities

Awareness can be categorized into awareness toward general abilities (1–30) and awareness toward programming abilities (31–55). In the category of awareness toward general abilities, a significant difference in PHP was observed for five items and a trend toward significance for three items. The ratio of items that improved was  $8/30=0.27$ . Significant differences were observed for the following five items: “(2) Understanding of computers,” “(3) Computer operation skills,” “(5) Ability to set challenges,” “(16) Ability to listen to others,” and “(27) Ability to construct and create knowledge.” Trends toward significance were observed for the following three items: “(4) Computer usage methods and broadening of situations,” “(8) Ability to study by oneself, ability to learn,” and “(11) Ability to analyze information.” Items (2), (3) and (4) likely improved because students created programs on a computer in class of PHP. Items (8) and (16) likely improved because students had to carefully listen to explanations of example programs and syntactic elements in the lectures in order to complete assignments. Items (5), (11), (16) and (27) likely improved because students performed exercises where they created programs based on example programs and assigned programs. As just described, abilities related to participating in the class appeared to improve.

In the category of awareness toward general abilities, a significant difference in JavaScript was observed for fifteen items and a trend toward significance for four items. The ratio of items that improved was  $19/30=0.63$ . Significant differences were observed for the following 15 items: (3), (5), (7), (11)–(14), (16)–(20), (22), (23), (27), (32), (35)–(48), (52)–(55). Trends toward significance were observed for the following four items: (2), (8), (21), (24). Items (2) and (3) likely improved because students created programs on a computer in class of JavaScript. Items (7), (8) and (16) likely improved because students had to carefully listen to explanations of example programs and syntactic elements in the lectures in order to complete assignments. Items (5), (11), (12), (13), (14), and (27) likely improved because students created assignments by referring example programs and example assignment. Items (17), (18), (19), (20), (21), (22), and (23) likely improved because students did peer evaluation and correction of assignment. As just described, abilities related to participating in the class appeared to improve.

By comparing PHP with JavaScript, it was suggested that the practice problem having a low degree of difficulty improves more awareness as the number of items improved in the class of JavaScript was more than that in the class of PHP. Difference of two classes is only one item “(4) Computer usage methods and broadening of situations”. Degree of awareness of Item (4) in the class of PHP improves. On the one hand, it in the class of JavaScript does not improve. However, significant difference between elongations in both classes was not recognized. Awareness in the class of JavaScript improved more than that in the class of PHP by evaluating mutually the programs, correcting them, and feeling sense of satisfaction after completing them.

### 4.2.2 Awareness toward Programming Abilities

In the category of attitudes toward programming abilities, a significant difference in the class of PHP was observed for 17 items and a trend toward significance for one item. The ratio of items that improved was  $18/25=0.72$ . The mean increase in score was 0.57 for attitudes toward general abilities and 1.24 for attitudes toward programming abilities. Analysis of the mean increase in these two scores with paired t-tests revealed a significant difference. This indicates that awareness toward programming abilities improved more than awareness toward general abilities ( $t(23)=3.6$ ,  $p<0.001$ ). Therefore, the mean increase in score in their awareness toward their programming abilities in the class of PHP was greater than that in awareness toward general abilities.

In the category of attitudes toward programming abilities, a significant difference in the class of JavaScript was observed for 19 items. The ratio of items that improved was  $19/25=0.76$ . The mean increase in score was 0.76 for attitudes toward general abilities and 1.68 for attitudes toward programming abilities. Analysis of the mean increase in these two scores with paired t-tests revealed a significant difference. This indicates that awareness toward programming abilities improved more than awareness toward general abilities ( $t(17)=3.4$ ,  $p<0.001$ ). Therefore, the mean increase in score in their awareness toward their programming abilities in the class of JavaScript was greater than that in awareness toward general abilities.

It can be concluded that awareness toward general abilities in the class of JavaScript improves more than that in the class of PHP through the course methods described in Section 2. On the other hand, the classes of JavaScript and PHP on the awareness toward their programming abilities brought to improve in the same scale. It is suggested that degree of difficulty of exercise problems have an influence on awareness toward general abilities and do not affect the awareness toward their programming abilities. If you want to improve awareness toward

general abilities, it should be careful for the degree of difficulty of exercise problems.

#### *4.3 About Comparison of Exercise Problem Submission Rate*

The result of unpaired t-test on average submission rates in PHP and JavaScript programming showed that the t value for Exercise 1 showed tendency for significant difference. For Exercise 2 was showed no significant difference. In addition, when the maximum value and minimum value for submission rates in two classes are compared, there was difference in the minimum value for the problem in Exercise 1 while there was little difference in maximum value. The minimum value for PHP programming was smaller than that of JavaScript programming class. In questionnaire for PHP programming class, the cause of the evaluation being low seems to be due to the low submission rate for the first exercise problem. The average submission rate was also low in Exercise 2 of JavaScript programming class, with the minimum value for submission rate being the smallest. However, the evaluation in class questionnaire was not low. Based on these, it was indicated that it was better to give a problem for which nearly everyone can submit as the first problem when at least one of two problems is to be submitted.

### **5. Conclusion**

In JavaScript programming education in a university, we gave lectures and exercises to have the students submit programs and reports for exercise problems. In the final 2 weeks, programs for voluntary themes were developed for mutual evaluation, correction with reference to the evaluation and another mutual evaluation to check if they could correct properly. Surveys on degree of term recognition and awareness were conducted before and after the course. The survey data were analyzed in significance tests. It is surmised that this finding can be used to help in other types of lectures.

The following were found in this study:

- (1) The overall degree of recognition of terms related to JavaScript programming improved, and the overall amount of knowledge on them also increased after the lecture course.
- (2) The degree of recognition in JavaScript programming improved for 99% of the 72 terms after the lecture course.
- (3) The overall awareness related to abilities in JavaScript programming improved after the lecture course.
- (4) Awareness for approximately 78% of the 55 items in JavaScript programming improved after the lecture course.
- (5) Items with significant difference or tendency for significant difference in awareness related to general abilities corresponded to 77% in JavaScript programming, and those in awareness related to programming corresponded to 80%.
- (6) From t-test about the recognition of seven common terms in both classes of PHP and JavaScript programming, the growth of knowledge quantity in the JavaScript increases generally more than that in PHP.
- (7) There was little difference in the growth of average evaluation values according to awareness between PHP and JavaScript programming.
- (8) The mean increase in score in their awareness toward their programming abilities in both classes of PHP and JavaScript was greater than that in awareness toward general abilities.
- (9) It is better to give a problem for which nearly everyone can submit the first problem when at least one of two problems is to be submitted.

We would like to make further examination in the future through multivariate analysis to classify the students as well as evaluation items in order to identify helpful activities and difference in instruction methods depending on the student group. We would also like to conduct analysis that may lead to findings other than those from this study.

### **Acknowledgements**

The author appreciates the support of the Grant-in-Aid for Scientific Research, foundation study (C: KAKENHI Grant Number 25350364) provided by the Ministry of Education, Culture, Sports, Science and Technology, Japan for this research. The author would like to express appreciation to the students who were surveyed and who helped collect educational information.

## References

- Anku. (2011). *Illustrated Book of JavaScript*. Shoeisha, Tokyo, Japan.
- Bersin, J. (2004). *The Blended Learning Book: Best Practices, Proven Methodologies, and Lessons Learned*. John Wiley & Sons, Inc., San Francisco, USA.
- Central Council for Education. (2012). Toward qualitative change of the university education to build a new future -aiming at the university which can nourish the ability to continue learning throughout the life and think independently. (Report). Retrieved from [http://www.mext.go.jp/b\\_menu/shingi/chukyo/chukyo0/toushin/1325047.htm](http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo0/toushin/1325047.htm).
- Miyaji, I., & Yoshida, K. (2014a). Improvement of the attitudes and their familiarity with terminology of a programming course with a blended learning structure. *Proceedings of the Canada International Conference on Education. CICE 2014*. Sydney, Canada. pp.301-307.
- Miyaji, I., & Yoshida, K. (2014c). Categories of attitude and student determined by cluster analysis of the attitudes toward programming abilities in a blended class. *International Journal for Cross-Disciplinary Subjects in Education*, 5(4), 1845-1853. <http://dx.doi.org/10.20533/ijcdse.2042.6364.2014.0257>
- Miyaji, I. (2009a). Effects on blended class which incorporates e-learning inside the classroom, *Proceedings of E-learn2009. The 20th World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education*. Vancouver, Canada. pp.1818-1826.
- Miyaji, I. (2011). Comparison between effects in two blended classes which e-learning is used inside and outside classroom. *US-China Education Review, USA*, 8(4), 468-481.
- Miyaji, I. (2013). Consciousness and recognition degree of terms in programming through blended classes. *Proceedings of Japanese Society for Science Education, Chugoku Branch Symposium "Toward Blended Learning from E-learning (part 5)"*, pp.24-28.
- Miyaji, I. (2014b). Useful activities for improving the attitudes in a programming course with a blended class. *Proceedings of the 4th Annual International Conference Education & E-Learning. EeL 2014*, Bangkok, Thailand. pp.81-89.
- Miyaji, I. (Ed.) (2009b). *Toward Blended Learning from E-learning*, Kyouritu-Shuppan, Tokyo, Japan.
- Miyaji, I., & Yoshida, K. (2005). The practice and learning effect of education by blending of lecture and e-learning. *Transactions of Japanese Society for Information and Systems in Education*. 22(4), 230-239.
- Miyaji, I., Yoshida, K., & Naruse, Y. (2007). The effects of blending e-learning and lectures utilizing a structured notebook. *Transactions of Japanese Society for Information and Systems in Education*, 4(3), 208-215.
- Shinkai, J., & Miyaji, I. (2011). Effects of blended instruction on C programming education. *Transactions of Japanese Society for Information and Systems in Education*, 28(2), 151-162.
- Takaoka, E., & Ishii, W. (2008). Fully E-learning Java Programming Course: Design, Development and Assessment. *Transactions of Japanese Society for Information and Systems in Education*, 25(2), 214-225.
- Taniguchi, R. (2011). Collaborative Learning through Sharing Students' Work Information and Evaluation of This Learning Environment. *Transactions of Japanese Society for Information and Systems in Education*, 28(4), 283-291.

## Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).