Subject: Arrays

Overview

In our new lesson, our users will learn arrays with the ability to keep multiple variables within them. Intended purpose of the arrays is that they have the feature of keeping multiple variables at the same time within them. When we say multiple numbers it can also be big values such as 100 or 1000. Instead of declaring a large number of variables with the same name one by one, we can keep them together by means of arrays.

When we have one variable, array and variables may be considered as the same thing, but we shouldn't use variables instead of arrays for multiple variables.

Activity

Let's start to understand arrays by giving an example of the fruits after shopping from the market. Suppose that we bought 5 different fruits. Apple, Banana, Orange, Apricot and Peach. Let's call all of these "Fruits". We transferred multiple values into the “Volunteers” variable distinctly from variable samples we made in our previous lesson. Because of keeping multiple variables within it, "Volunteers" became an array. Items within this array have different values but have a common name. Now let's give some commands respectively to the items of the array that we formed for you to comprehend this better.

- First item of the “Fruits" array is “Apple”
- Second item of the “Fruits" array is “Banana”
- Third item of the “Fruits" array is “Orange”
- Fourth item of the “Fruits" array is “Apricot”
- Fifth item of the “Fruits" array is “Peach”

In this manner, we can differentiate the variables belonging to the same type thanks to the naming we use. By virtue of the array we use, we could keep multiple variables of a type together and assign them different values.

We can do this activity by coding as below.

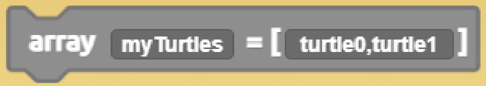
Fruits = [Apple, Banana, Orange, Apricot, Peach]

Now let's continue to our document for better understanding of our subject.

Commands to be Used

Commands which are used with the array in the form of turtle[1].forward() and the commands which has been learned before.

Blocks and Button icons to be used in the training about this subject

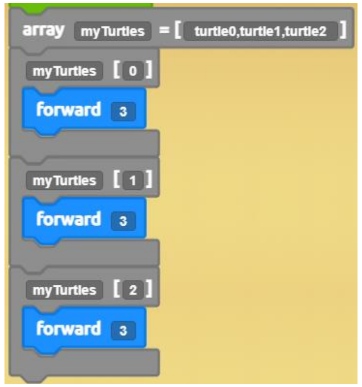
Block	Explanation
 array myTurtles = [turtle0,turtle1]	Arrays which can keep multiple variables within them enable us to assign separate tasks for each of our characters in case there are multiple characters having the same name in our task scenes. We can transfer our characters into an array with their common names and recall our characters according to the numbers we gave. *Sample usage* (**Python**): myTurtles = [turtle0,turtle1] myTurtles [0].forward(3) myTurtles [1].forward(3) forward(5) *Sample usage* (**Javascript**): var myTurtles = [turtle0,turtle1] myTurtles [0].forward(3) myTurtles [1].forward(3) forward(5)
Text	
[]	

WARNING

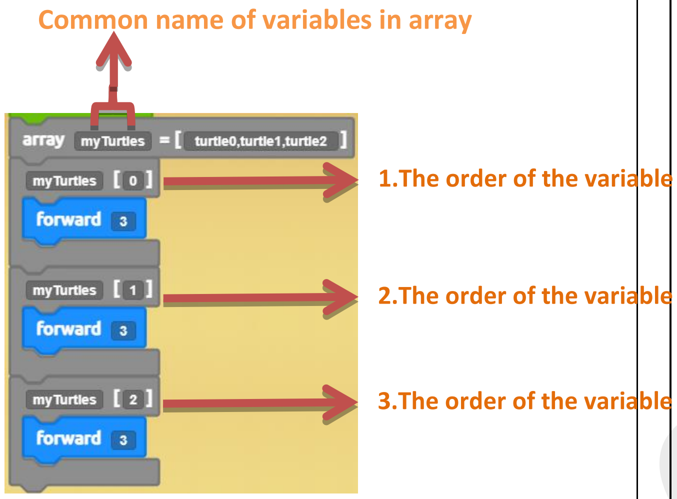
The numbering of arrays starts with 0 (zero). It means the number of the first variable is 0, the number of the second variable is 1.

BLOCK

Usage



Explanation



Common name of variables in array

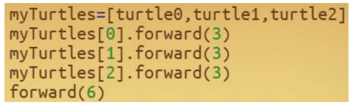
1.The order of the variable

2.The order of the variable

3.The order of the variable

PYTHON

Usage

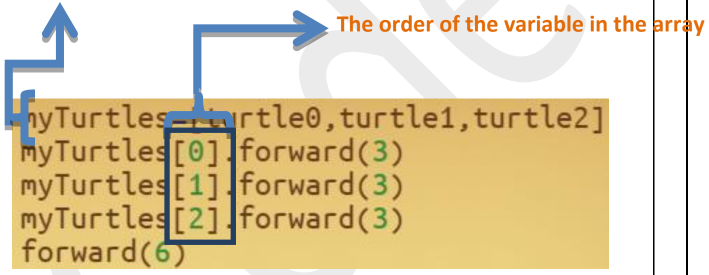


```
myTurtles=[turtle0,turtle1,turtle2]
myTurtles[0].forward(3)
myTurtles[1].forward(3)
myTurtles[2].forward(3)
forward(6)
```

Explanation

Array definition

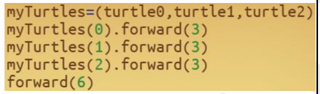
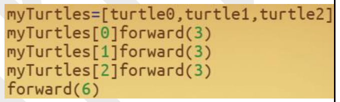
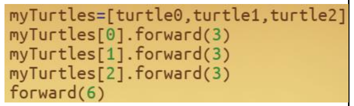
The order of the variable in the array



```
myTurtles=[turtle0,turtle1,turtle2]
myTurtles[0].forward(3)
myTurtles[1].forward(3)
myTurtles[2].forward(3)
forward(6)
```

Erraneous writing forms

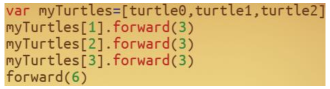
Python		


`myTurtles=(turtle0,turtle1,turtle2)`
`myTurtles(0).forward(3)`
`myTurtles(1).forward(3)`
`myTurtles(2).forward(3)`
`forward(6)`
	
	`myTurtles=[turtle0,turtle1,turtle2]`
	`myTurtles[0]forward(3)`
	`myTurtles[1]forward(3)`
	`myTurtles[2]forward(3)`
	`forward(6)`
		
		`myTurtles=[turtle0,turtle1,turtle2]`
		`myTurtles[0].forward(3)`
		`myTurtles[1].forward(3)`
		`myTurtles[2].forward(3)`
		`forward(6)`

✗	✗	✓
Why is it wrong?	**Why is it wrong?**	**Correct form:**
In order to recall the items in the arrays, we must write their numbers inside square brackets. “[]” (Alt Gr 8-9)	When we want to assign our characters special tasks we must put a point between them while writing the tasks with their names.	Our command is correct, because our numbers to reach array items are within square brackets and point is used after character names when task is written.

JAVASCRIPT

Usage

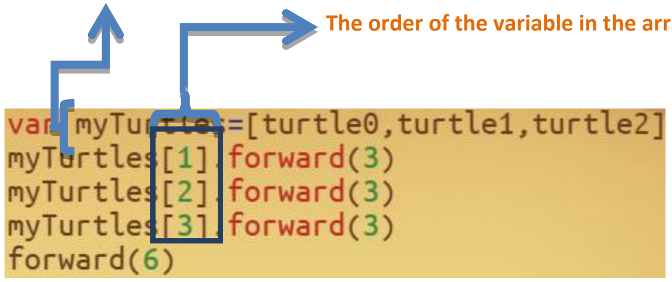


```
var myTurtles=[turtle0,turtle1,turtle2]
myTurtles[1].forward(3)
myTurtles[2].forward(3)
myTurtles[3].forward(3)
forward(6)
```

Explanation

Array definition

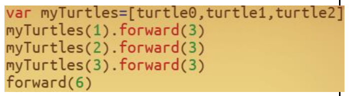
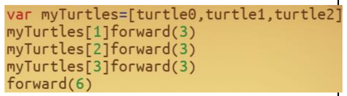
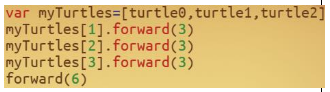
The order of the variable in the array



```
var myTurtles=[turtle0,turtle1,turtle2]
myTurtles[1].forward(3)
myTurtles[2].forward(3)
myTurtles[3].forward(3)
forward(6)
```

Erraneous writing forms

JavaScript		


var myTurtles=[turtle0,turtle1,turtle2]
myTurtles(1).forward(3)
myTurtles(2).forward(3)
myTurtles(3).forward(3)
forward(6)
	
	var myTurtles=[turtle0,turtle1,turtle2]
	myTurtles[1]forward(3)
	myTurtles[2]forward(3)
	myTurtles[3]forward(3)
	forward(6)
		
		var myTurtles=[turtle0,turtle1,turtle2]
		myTurtles[1].forward(3)
		myTurtles[2].forward(3)
		myTurtles[3].forward(3)
		forward(6)

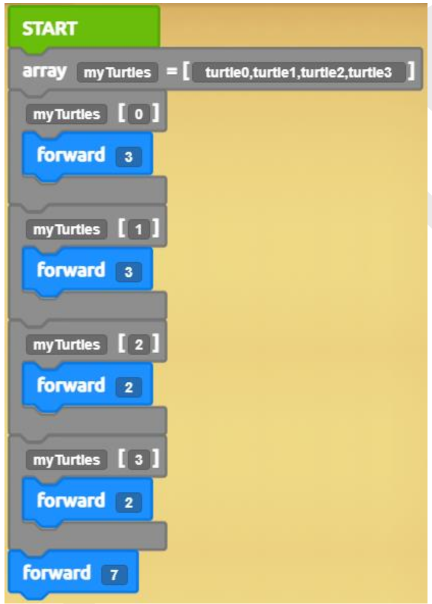
✗	✗	✓

Sample Solution: Scene 33

Algorithm

- Start
- define the array with the name "myTurtles"
- assign the variables to "myTurtles" array [turtle0, turtle1, turtle2, turtle3]
- First [0] turtle move 3 squares forward
- Second[1] turtle move 3 squares forward
- Third [2] turtle move 3 squares forward
- Forth [3] turtle move 3 squares forward
- Main character move 7 squares forward

Block



```
START
array myTurtles =[ turtle0,turtle1,turtle2,turtle3 ]
myTurtles [ 0 ]
forward 3
myTurtles [ 1 ]
forward 3
myTurtles [ 2 ]
forward 2
myTurtles [ 3 ]
forward 2
forward 7
```

Python

```
myTurtles = [turtle0,turtle1,turtle2,turtle3]
myTurtles[0].forward(3)
myTurtles[1].forward(3)
myTurtles[2].forward(2)
myTurtles[3].forward(2)
forward(7)
```

Javascript

```
var myTurtles = [turtle0,turtle1,turtle2,turtle3]
myTurtles[0].forward(3)
myTurtles[1].forward(3)
myTurtles[2].forward(2)
myTurtles[3].forward(2)
forward(7)
```