Ignition / ArrayHelpers

Namespace **Playniax.Ignition**

Script can be found in **Assets/Playniax/Framework/Ignition/Scripts/ArrayHelpers.cs**

Class ArrayHelpers

Description

Collection of array functions.

Public Methods	Description
`static T[] Add(T[] array, T value)`	Returns a new array with added value.
`static T[] Insert(T[] array, T value)`	Returns a new array with inserted value.
`static T[] Merge(T[] array1, T[] array2)`	Returns a new array with both arrays merged.
`static T[] Skim(T[] array)`	Returns a new array with the first value removed.
`static T[] Shuffle(T[] array)`	Shuffle array.

Example

Example can be found in **Assets/Playniax/Framework/Ignition/Examples/01 - Framework/ArrayHelpers.unity**

Example script can be found in **Assets/Playniax/Framework/Ignition/Examples/01 - Framework/Scripts (MonoBehaviour)/ArrayHelpers_Example.cs**

```
using UnityEngine;
using Playniax.Ignition;

public class ArrayHelpers_Example : MonoBehaviour
{
    // Custom type.
    public class MyType
    {
        public string name = "Unknown";
    }
    void Start()
    {
        // Create types.
        var myType1 = new MyType();
        var myType2 = new MyType();
        var myType3 = new MyType();

        // Fill types with data.
        myType1.name = "Tony";
        myType2.name = "Tanya";
        myType3.name = "David";

        // Declare the array.
        MyType[] list = null;

        // Add the types to the array.
        list = ArrayHelpers.Add(list, myType1);
        list = ArrayHelpers.Add(list, myType2);
        list = ArrayHelpers.Insert(list, myType3);

        // Shuffle the array.
        list = ArrayHelpers.Shuffle(list);

        // Show results.
        for (int i = 0; i < list.Length; i++)
        {
            Debug.Log(list[i].name);
        }
    }
}
```