Create a Stack Class

In the last section, we talked about what a stack is and how we can use an array to represent a stack. In this section, we will be creating our own stack class. Although you can use arrays to create stacks, sometimes it is best to limit the amount of control we have with our stacks. Apart from the `push` and `pop` method, stacks have other useful methods. Let's add a `peek`, `isEmpty`, and `clear` method to our stack class.

Write a `push` method that pushes an element to the top of the stack, a `pop` method that removes and returns the element on the top of the stack, a `peek` method that looks at the top element in the stack, an `isEmpty` method that checks if the stack is empty, and a `clear` method that removes all elements from the stack. Normally stacks don't have this, but we've added a `print` helper method that console logs the collection.

	Your Stack class should have a push method.
	Your Stack class should have a pop method.
	Your Stack class should have a peek method.
	Your Stack class should have a isEmpty method.
	Your Stack class should have a clear method.
	The peek method should return the top element of the stack.
	The pop method should remove and return the top element of the stack.
	The isEmpty method should return true if the stack does not contain any element.
	The clear method should remove all elements from the stack.