

# fluid.accumulate

## fluid.accumulate(list, fn, arg)

Scans through a list of objects, "accumulating" a value over them (may be a straightforward "sum" or some other chained computation). The results will be added to the initial value passed in by "arg" and returned

```
fluid.accumulate(list, fn, arg);
```

**File name:** Fluid.js

## Parameters

<b>list</b>	(Array) The list of objects to be accumulated over.
<b>fn</b>	(Function) An "accumulation function" accepting the signature (object, total, index) where object is the list member, total is the "running total" object (which is the return value from the previous function), and index is the index number.
<b>arg</b>	(Object) The initial value for the "running total" object.

## Return Value

<b>Object</b>	The modified arg object.
---------------	--------------------------

## See Also

---

## Notes

To a Google developer, this would be "reduce", from the "map/reduce" pairing.

## Example

```
var func = function (column, list) {  
    return list.concat(column.elements);  
};  
var modules = fluid.accumulate(layout.columns, func, [])
```

In this example, the function `func` will add the `elements` property of each entry of `list.columns` to an initially empty array and return the filled array.