

Functions As Objects

Functions and methods in boo are first class objects whose type exposes some very useful methods:

- `Invoke(<function args>)` as `<function return type>`: invokes the function synchronously (pretty much the same as using `()` in front of the function name)
- `BeginInvoke(<function args, > callback as AsyncCallback, asyncState)` as `IAAsyncResult`: invokes the function asynchronously and returns a handle that can be used to check the status of the execution as well as recovering the return value
- `BeginInvoke(<function args, > callback as AsyncCallback)` as `IAAsyncResult`: the same as above without the `asyncState` parameter
- `EndInvoke(AsyncResult as IAsyncResult)` as `<function return type>`: finishes an asynchronous invocation returning the final value

Invoke in action:

```
print("Hello!")  
print.Invoke("Hello!!")  
print.Invoke.Invoke("Hello!!!")
```

See [Asynchronous Design Pattern](#) for examples of `BeginInvoke/EndInvoke` in action.