

```
function addLogging(code) {  
  var ast = esprima.parse(code);  
  estraverse.traverse(ast, {  
    enter: function(node, parent) {  
      if (node.type === 'FunctionDeclaration'  
        || node.type === 'FunctionExpression') {  
        addBeforeCode(node);  
      }  
    }  
  });  
  return escodegen.generate(ast);  
}
```

```
function addBeforeCode(node) {  
  var name = node.id ? node.id.name : '<anonymous function>';  
  var beforeCode = "console.log('Entering " + name + "()');";  
  var beforeNodes = esprima.parse(beforeCode).body;  
  node.body.body = beforeNodes.concat(node.body.body);  
}
```

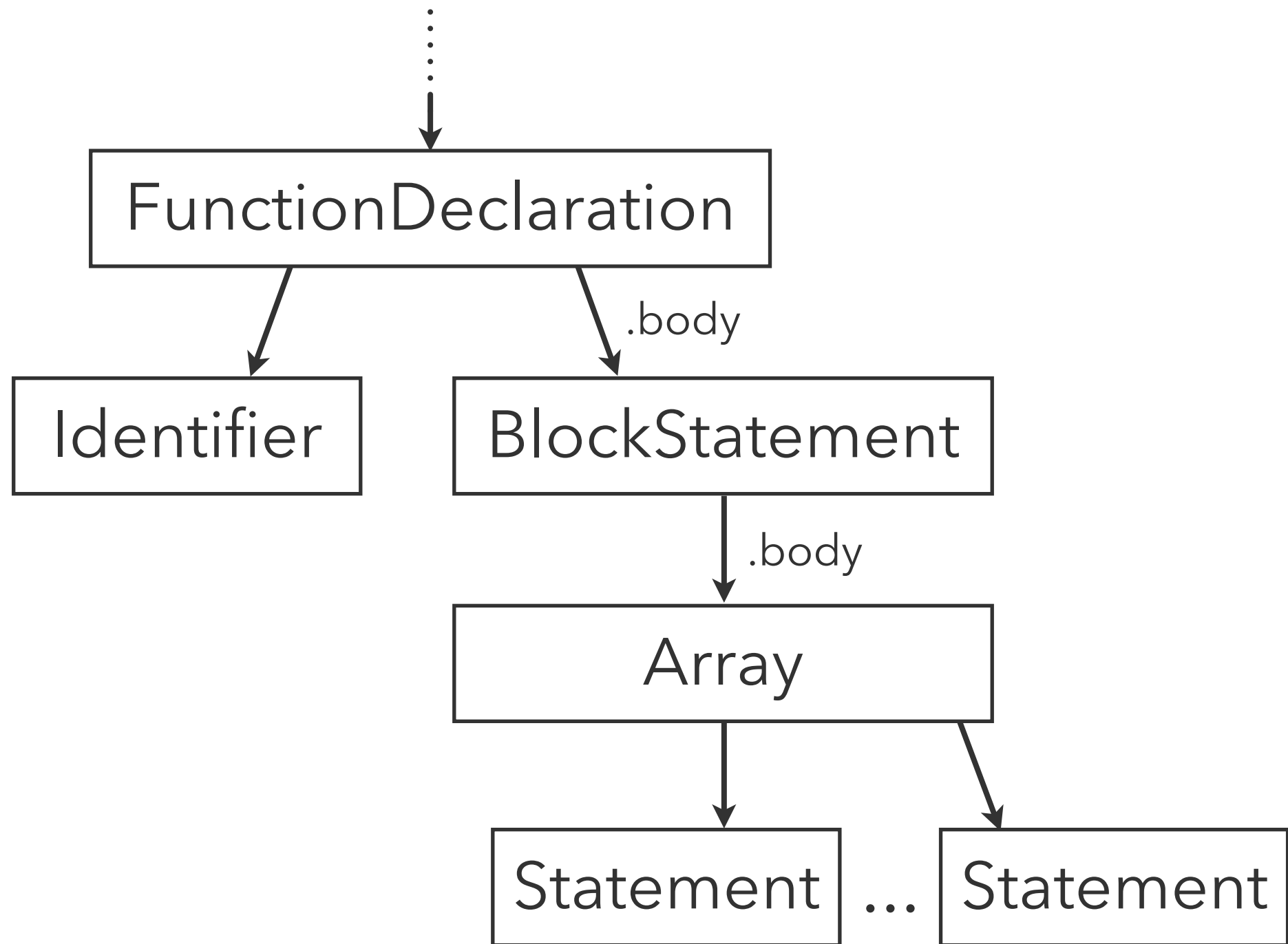


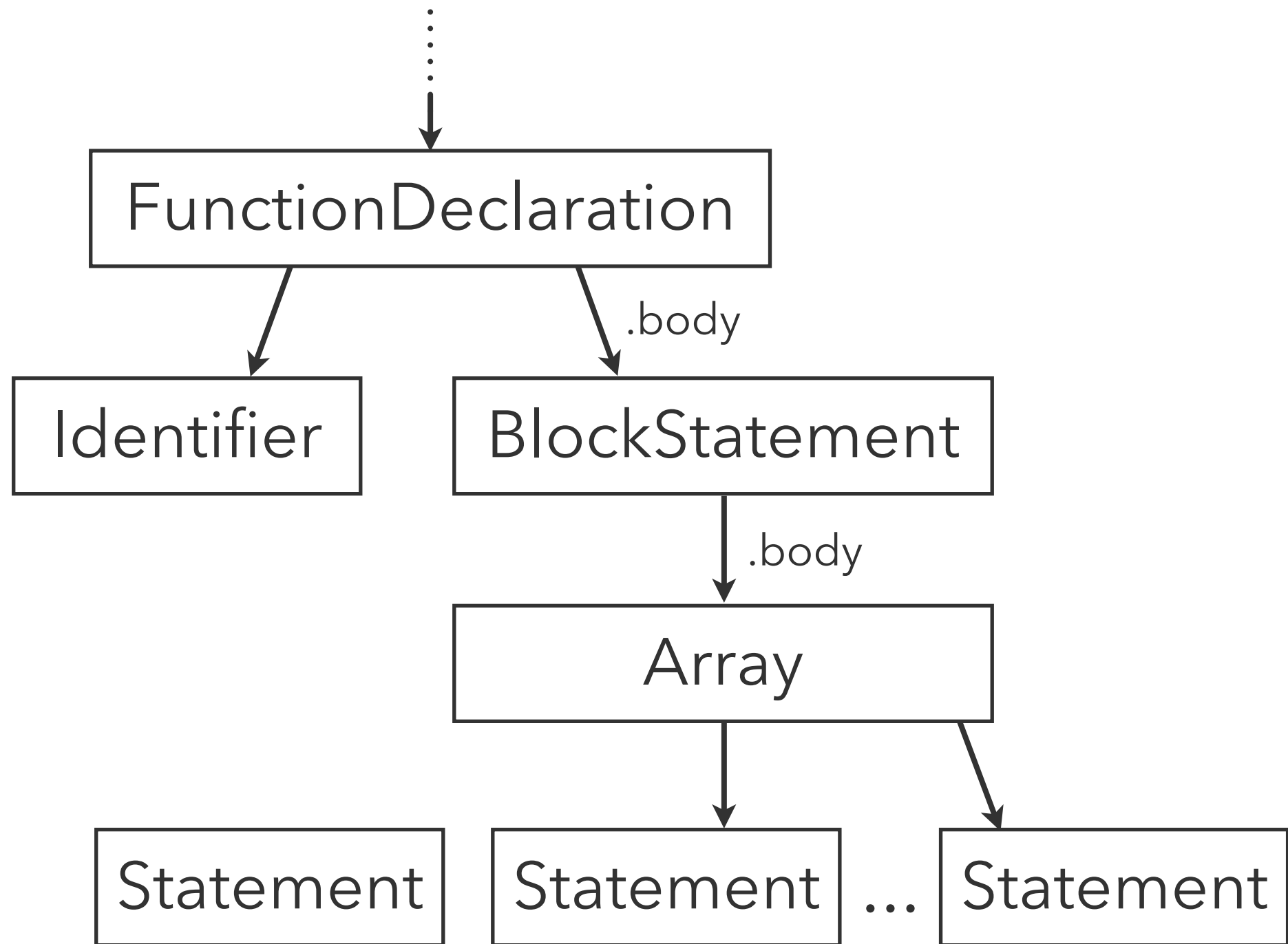
```
function addLogging(code) {  
  var ast = esprima.parse(code);  
  estraverse.traverse(ast, {  
    enter: function(node, parent) {  
      if (node.type === 'FunctionDeclaration'  
        || node.type === 'FunctionExpression') {  
        addBeforeCode(node);  
      }  
    }  
  });  
  return escodegen.generate(ast);  
}
```

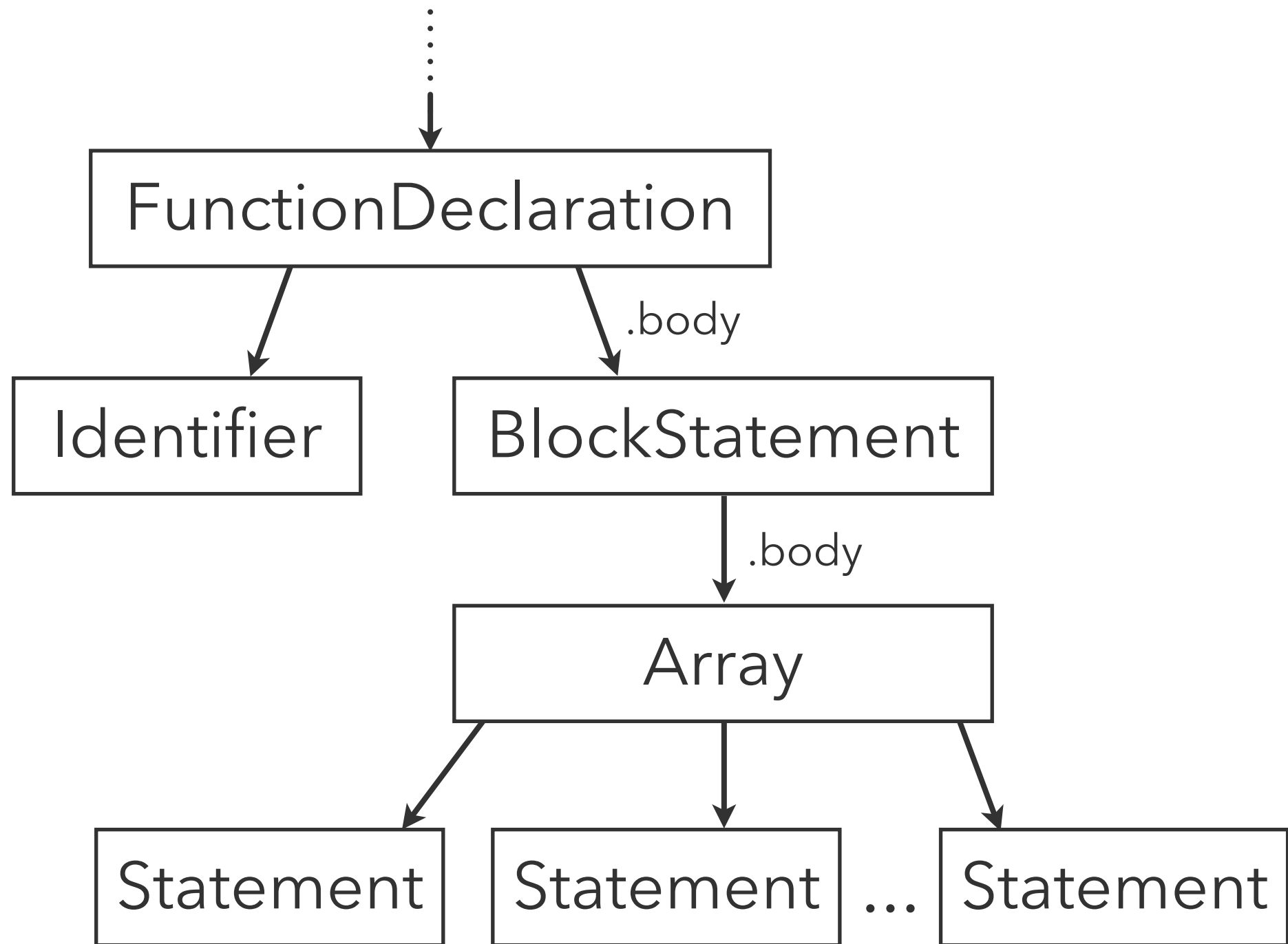
```
function addBeforeCode(node) {  
  var name = node.id ? node.id.name : '<anonymous function>';  
  var beforeCode = "console.log('Entering " + name + "()');";  
  var beforeNodes = esprima.parse(beforeCode).body;  
  node.body.body = beforeNodes.concat(node.body.body);  
}
```

```
function addLogging(code) {  
  var ast = esprima.parse(code);  
  estraverse.traverse(ast, {  
    enter: function(node, parent) {  
      if (node.type === 'FunctionDeclaration'  
        || node.type === 'FunctionExpression') {  
        addBeforeCode(node);  
      }  
    }  
  });  
  return escodegen.generate(ast);  
}
```

```
function addBeforeCode(node) {  
  var name = node.id ? node.id.name : '<anonymous function>';  
  var beforeCode = "console.log('Entering " + name + "()');";  
  var beforeNodes = esprima.parse(beforeCode).body;  
  node.body.body = beforeNodes.concat(node.body.body);  
}
```








```
function addLogging(code) {
  var ast = esprima.parse(code);
  estraverse.traverse(ast, {
    enter: function(node, parent) {
      if (node.type === 'FunctionDeclaration'
          || node.type === 'FunctionExpression') {
        addBeforeCode(node);
      }
    }
  });
  return escodegen.generate(ast);
}
```

```
function addBeforeCode(node) {
  var name = node.id ? node.id.name : '<anonymous function>';
  var beforeCode = "console.log('Entering " + name + "()');";
  var beforeNodes = esprima.parse(beforeCode).body;
  node.body.body = beforeNodes.concat(node.body.body);
}
```



```
addLogging("\n\nfunction foo(a, b) {\n  var x = 'blah';\n  var y = (function () {\n    return 3;\n  })();\n}\nfoo(1, 'wut', 3);\n");
```

