
Tern
http://ternjs.net

```
1  // Demo variable
2
3  var audience = "GOTO Berlin";
4
5  var greeting = "hello " + a
6
7  console.log(greeting);
8
```

Static scope

```
1  var aGlobal = 10;
2
3  function foo(arg1, arg2) {
4    a
5  }
6
```

Dynamic types

```
1  var player = {name: "Heinrich", score: 9999};
2
3  function foo(x) {
4    x.
5  }
6
7  foo(player);
8
```

Observation

```
var x = "foo"; // x is a string
var y = x;     // y is also a string
```

Observation

```
var x = "foo"; // x is a string OR bool
if (quux())
   x = false;

var y = x;     // y is also a string OR bool
```

Abstract Interpretation

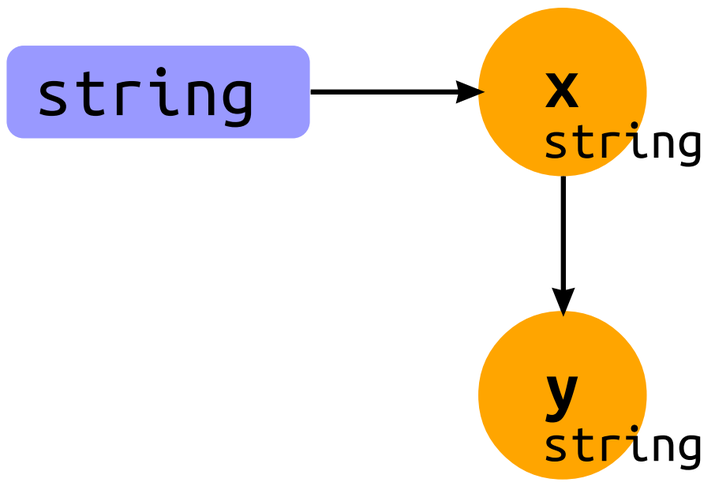
Syntax tree

```
{
  "type": "Program",
  "body": [
    {↔},
    {↔},
    {↔}
  ]
}
```

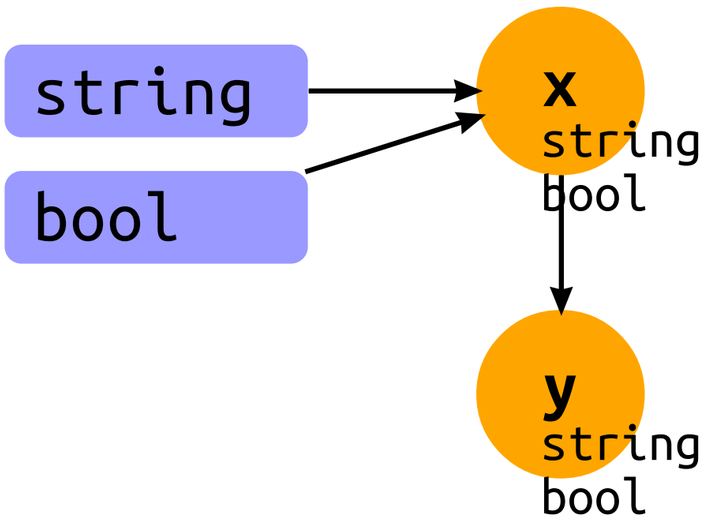
The user is typing

```
var array = [];
if (code.isBeing("edited")) {
  must.know.type.of(array.in
  orderTo = complete("a", "property");

```

```
var x = "foo";

var y = x;
```

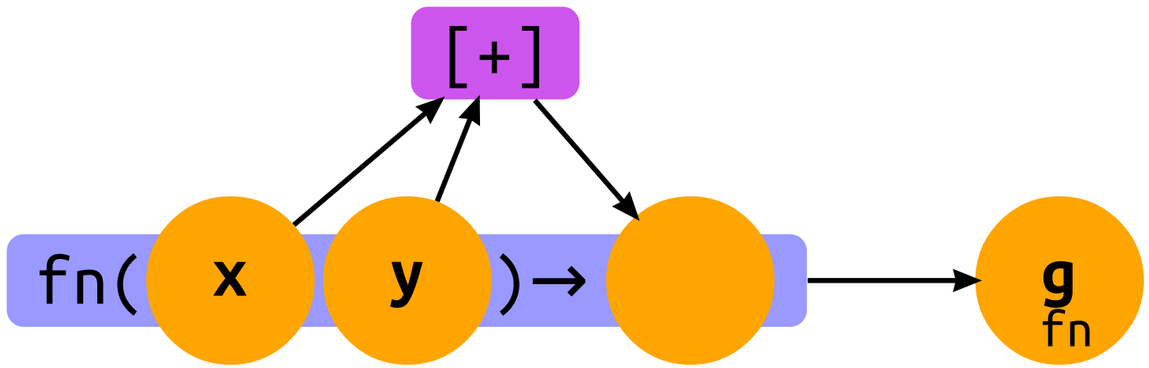


```
var x = "foo";
if (quux()) x = false;
var y = x;
```

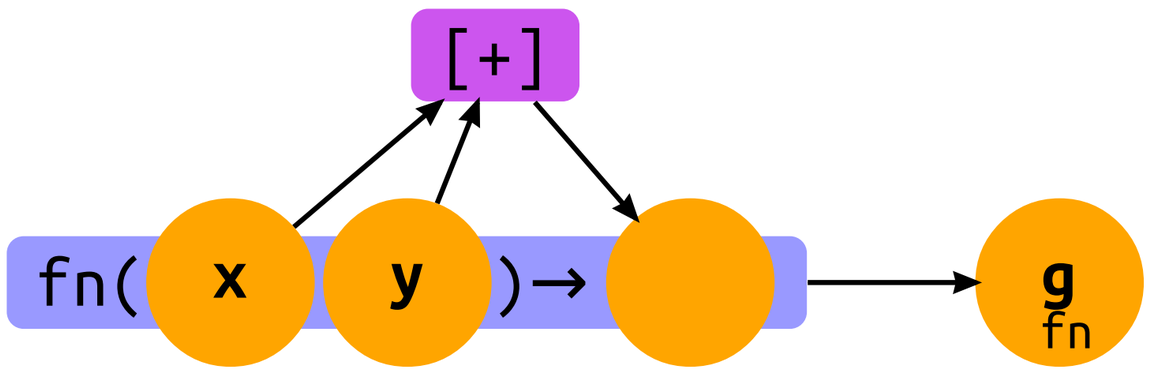


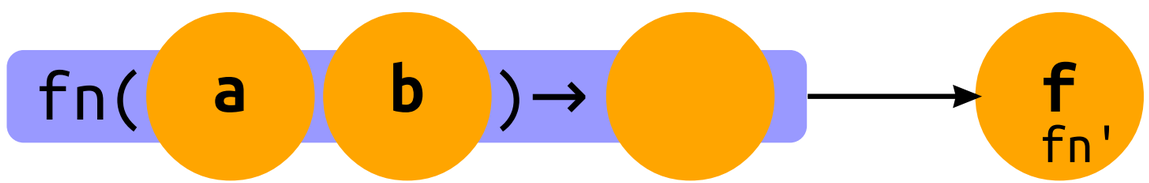
```
function f(a, b) { return g(b, a); }
function g(x, y) { return x + y; }
var r = f("foo", "bar");
```

```
function g(x, y) { return x + y; }
```

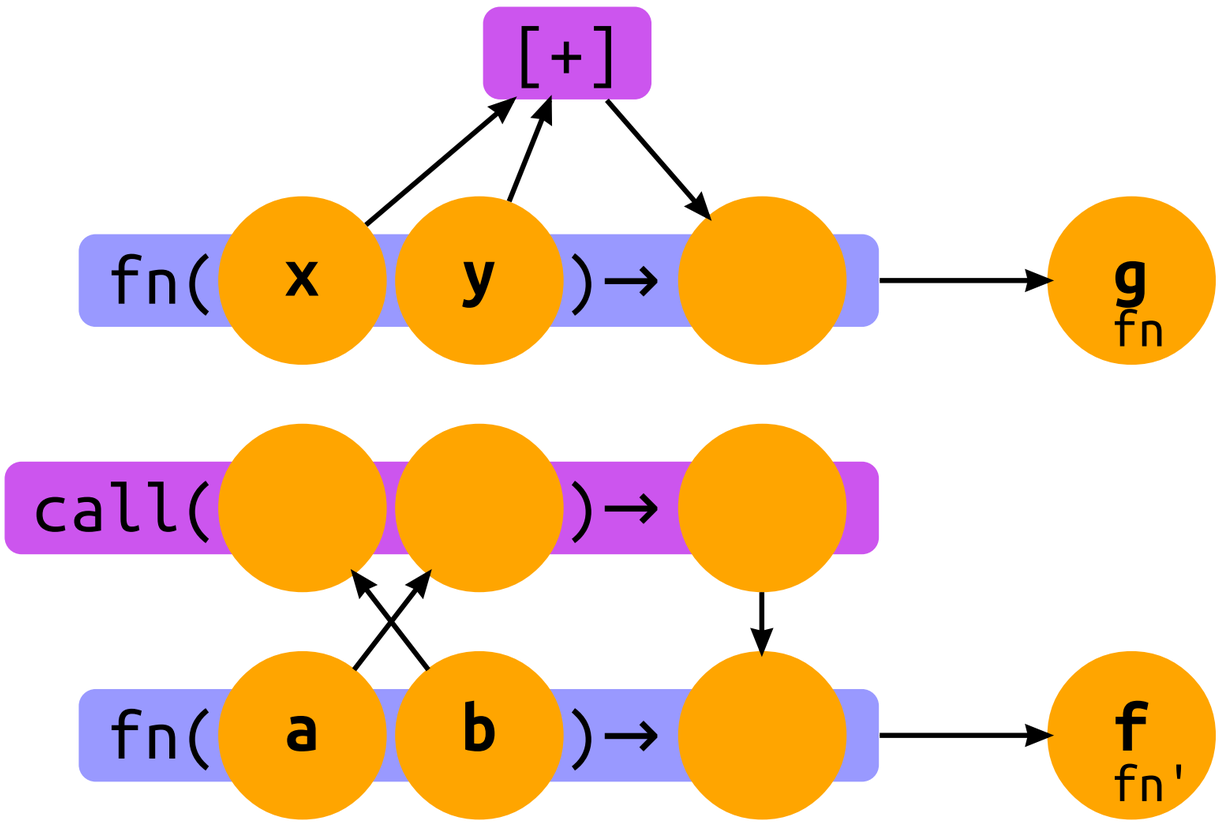


```
function f(a, b) { return g(b, a); }
function g(x, y) { return x + y; }
```

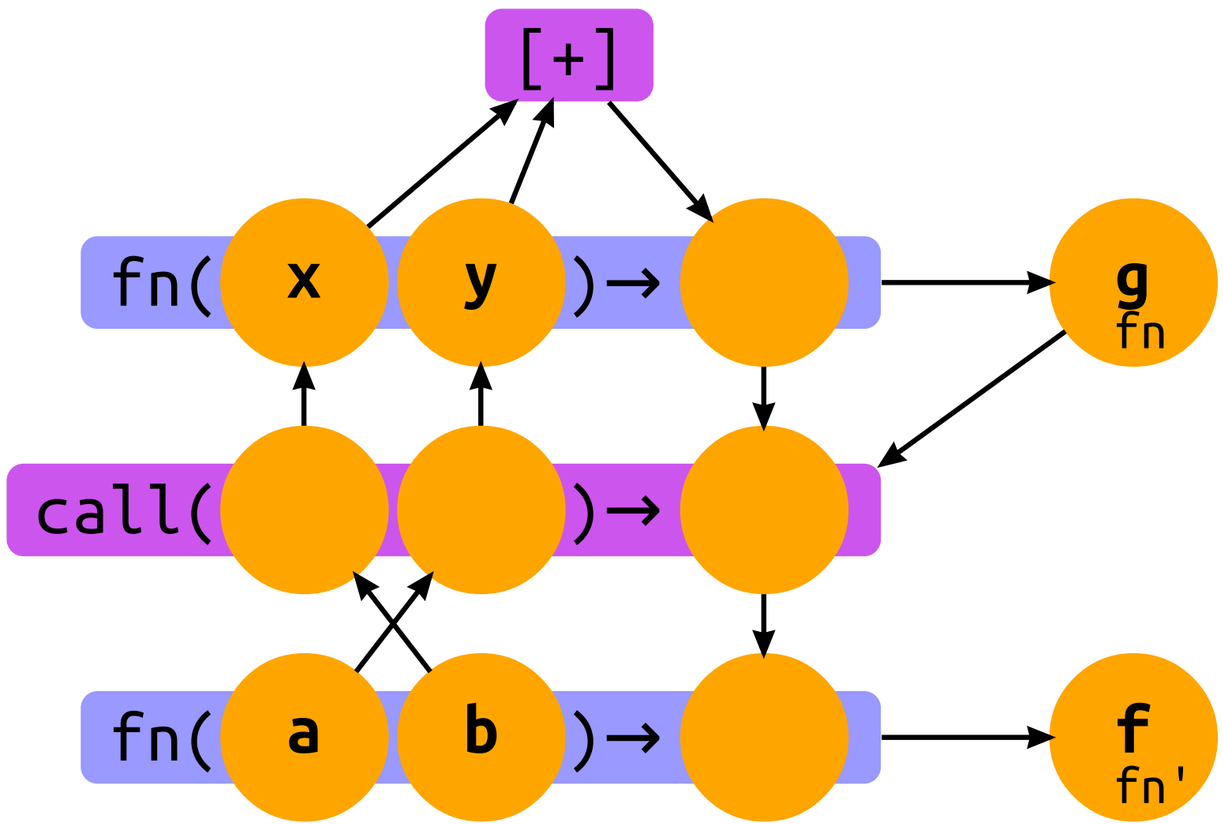


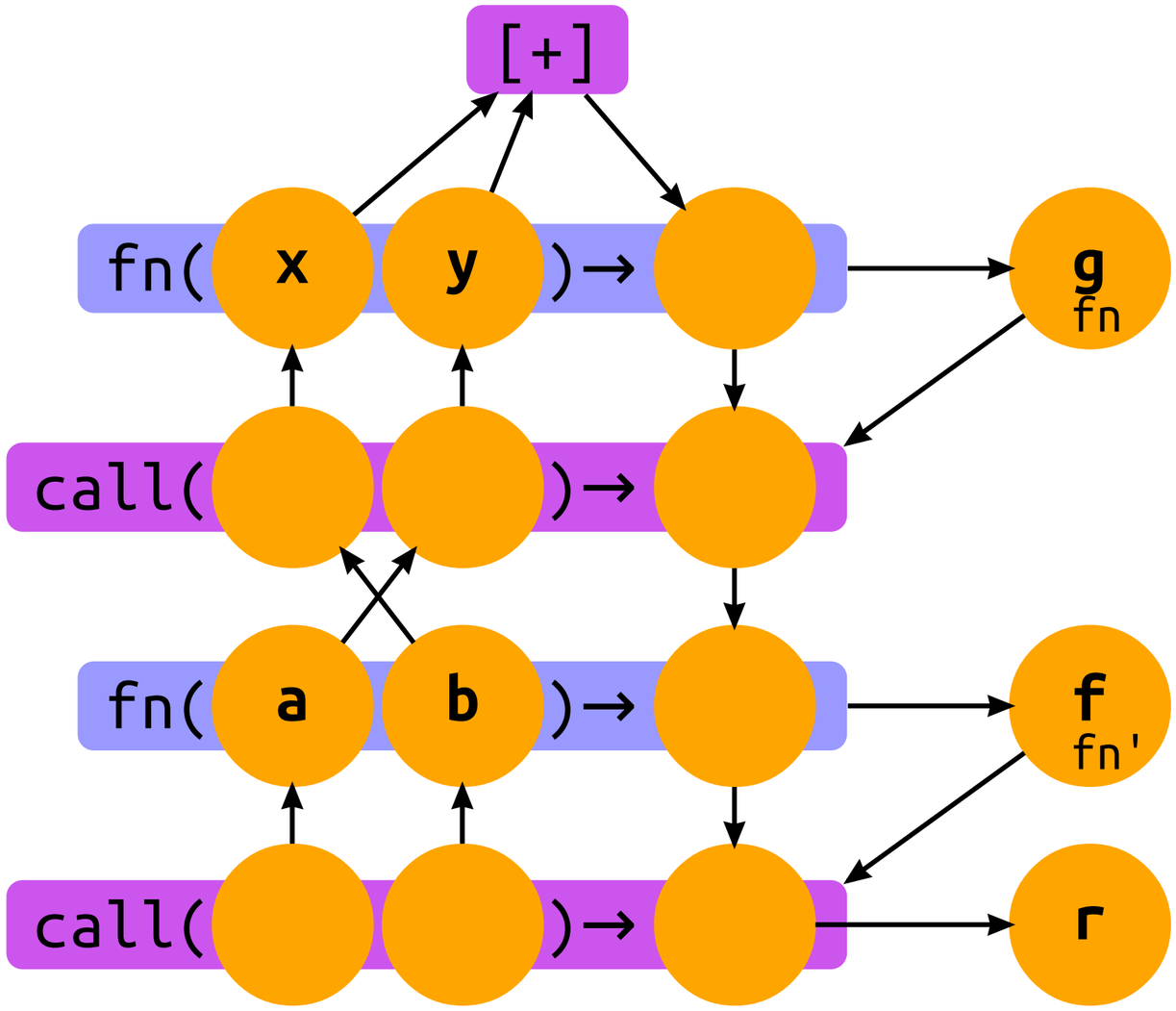


```
function f(a, b) { return g(b, a); }
function g(x, y) { return x + y; }
```

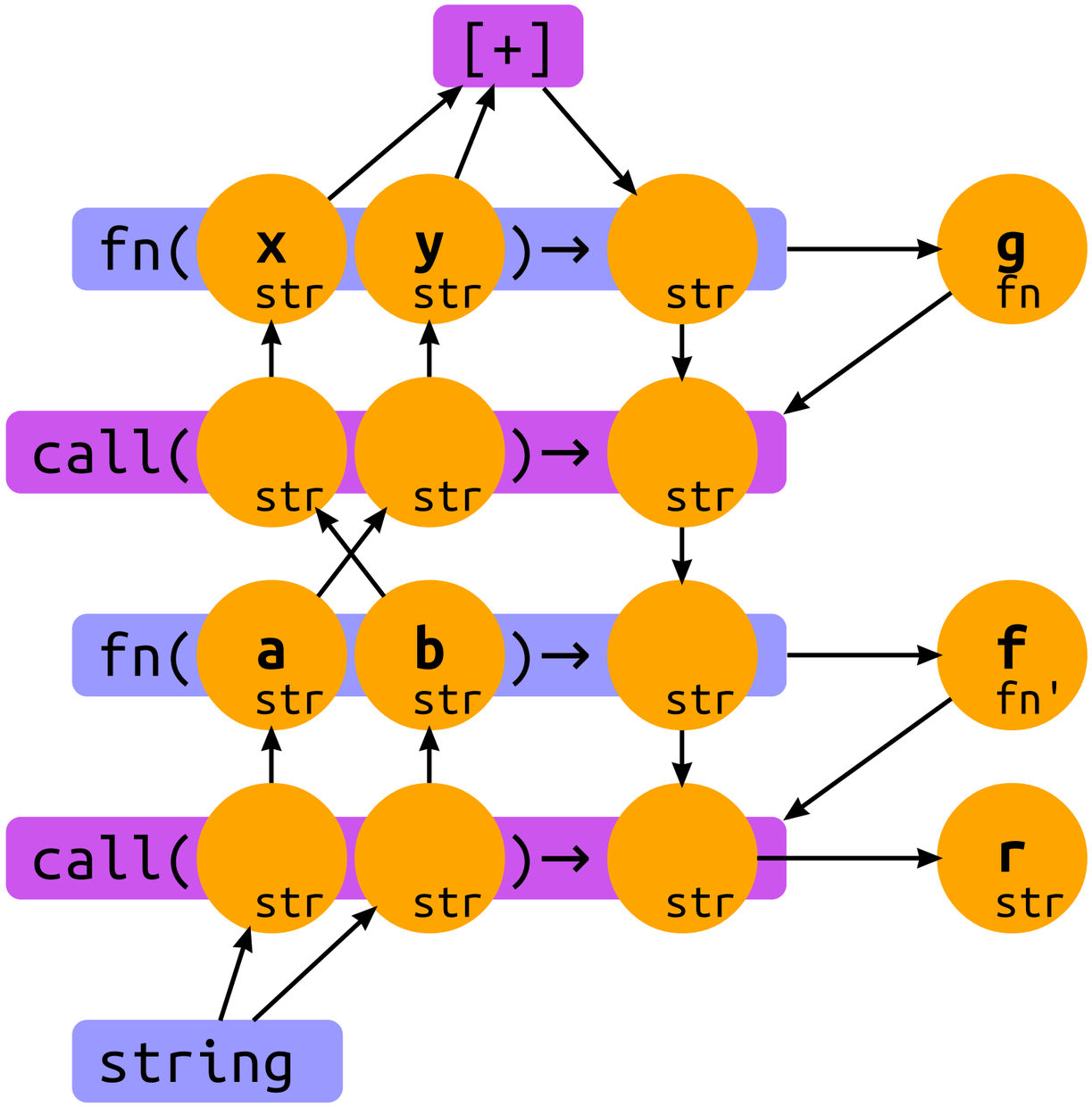


```
function f(a, b) { return g(b, a); }
function g(x, y) { return x + y; }
```

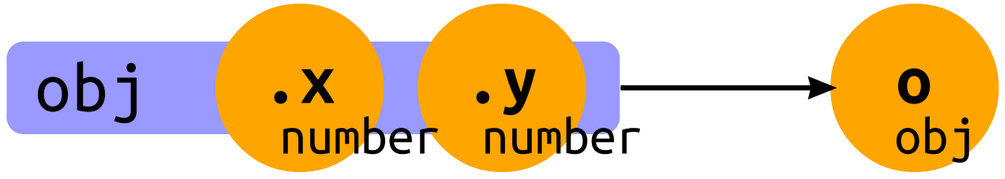




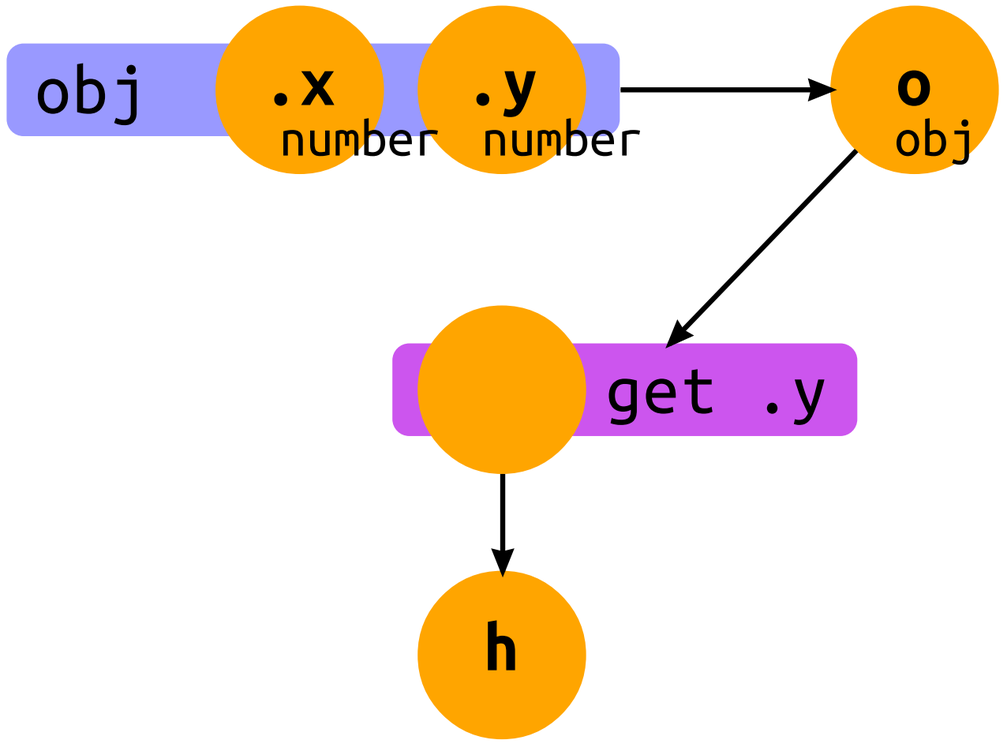

```
function f(a, b) { return g(b, a); }
function g(x, y) { return x + y; }
var r = f("foo", "bar");
```


[+]
fn(x str y str)→ str
g fn
call(str str)→ str
fn(a str b str)→ str
f fn'
call(str str)→ str
r str
string

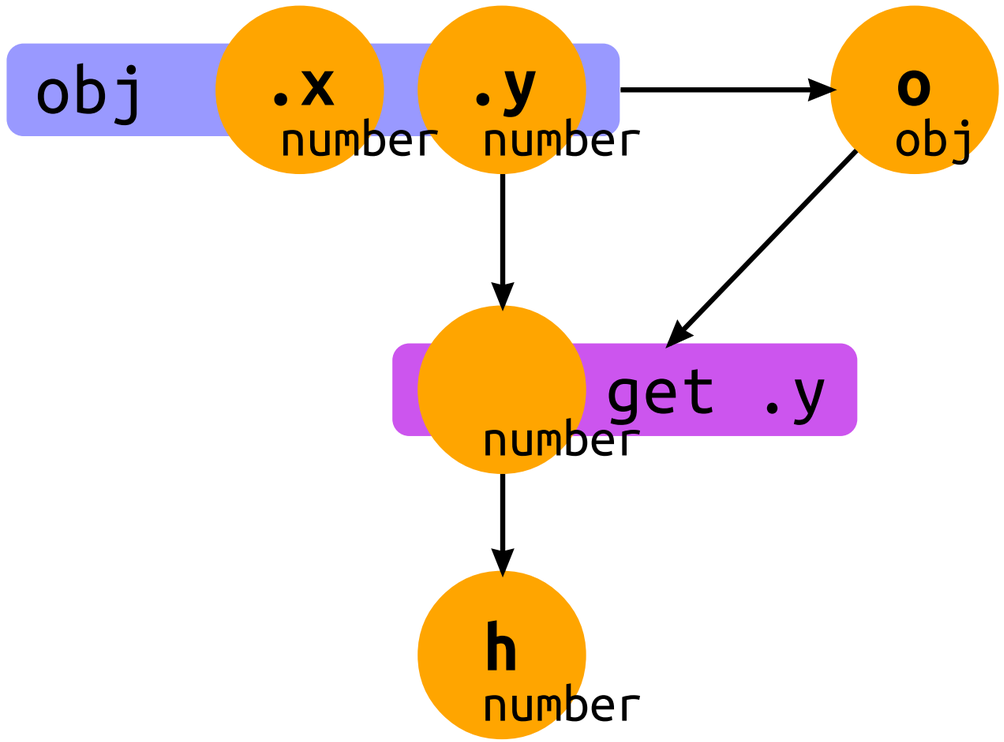
```
var o = {x: 10, y: 0};
```



```
var o = {x: 10, y: 0};
var h = o.y;
```



```
var o = {x: 10, y: 0};
var h = o.y;
```



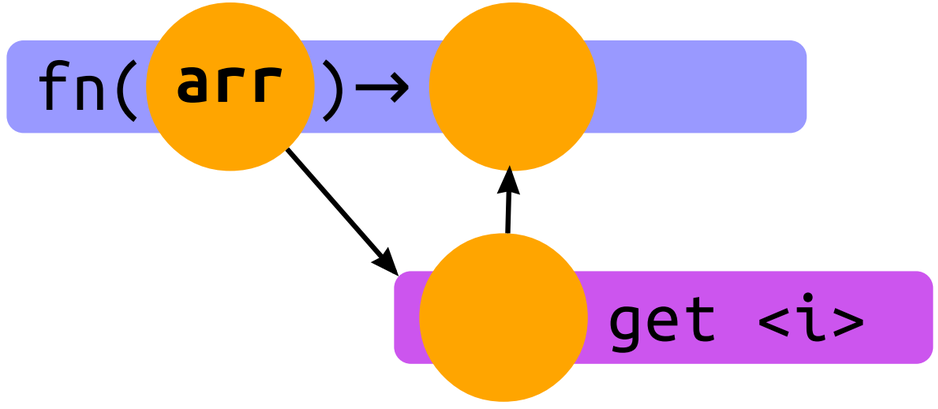
Parametric polymorphism

```
function lastOf(array) {
  return array[array.length - 1];
}
```

```
var num = lastOf([1, 2, 3]);
var str = lastOf(["a", "b", "c"]);
```

Parametric polymorphism

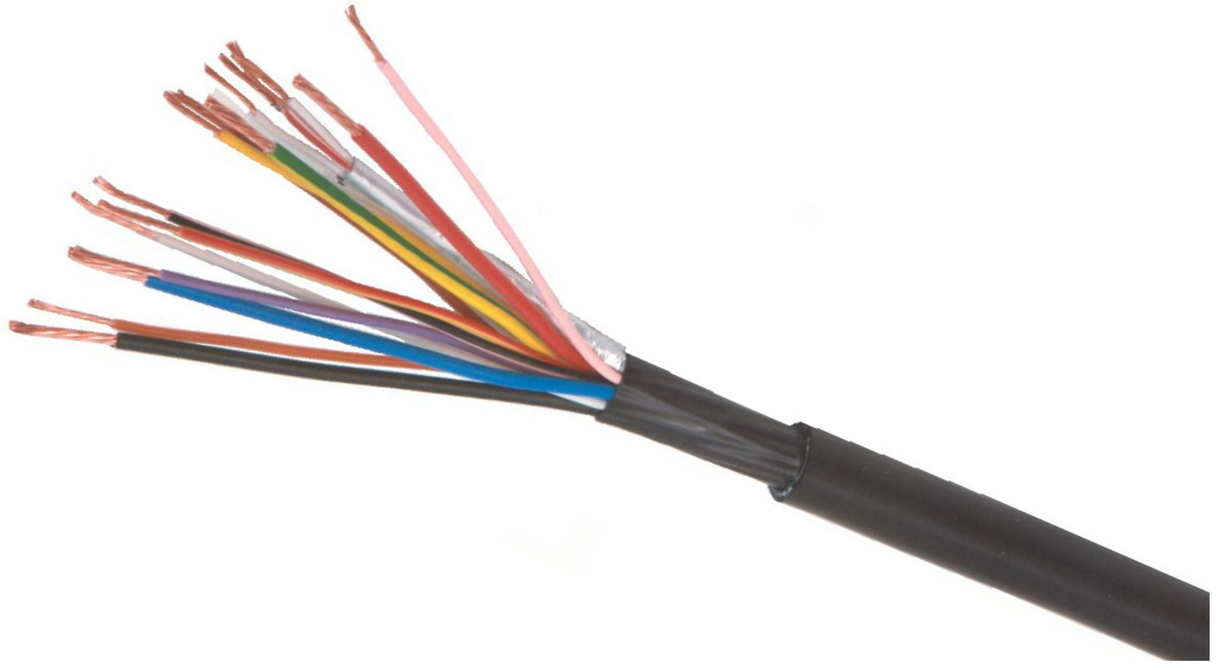
```
function lastOf(array) {
  return array[array.length - 1];
}
```



```
var __extends = function(child, parent) {
  for (var key in parent)
    child[key] = parent[key];
  function ctor() { this.constructor = child; }
  ctor.prototype = parent.prototype;
  child.prototype = new ctor();
};
```

Now what?

Missing Connections



Practical Use

```
// File 1

var myVar = {x: 1, y: 2};

(function(exports) {
  exports.capitalize = function(word)
    return word.charAt(0).toUpperCase(
      word.slice(1);
  };

  exports.garble = function(word) {
    function garbleCh(ch) {
      return String.fromCharCode(
        ch.charCodeAt(0) + 1);
    }
    return word.replace(/./g, garbleCl
  };
}(this.myMod = {}));

```

```
// File 2

my

```

For Emacs, Vim, & ST

- install editor plug-in
- `add .tern-project` file
- `go`

Related Work

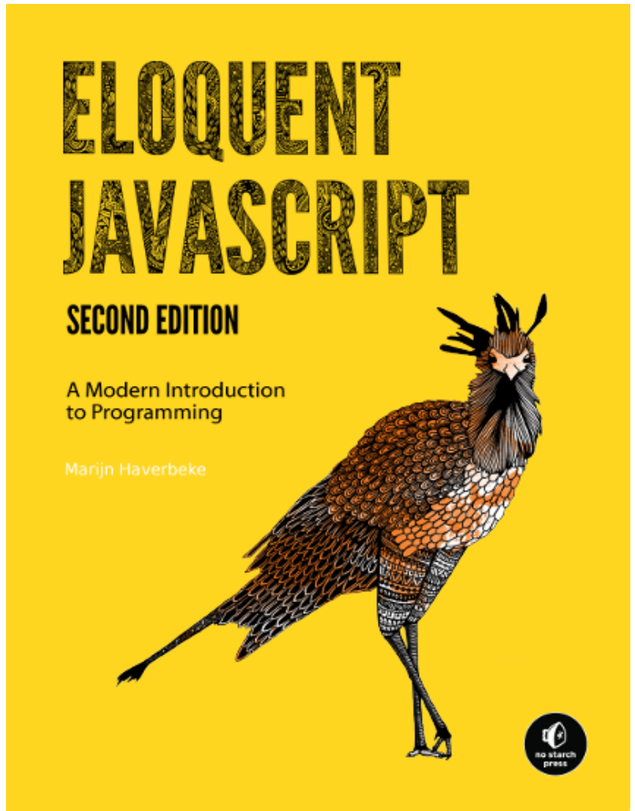
Fin.

ternjs.net

github.com/marijnh/tern

marijnhaverbeke.nl

twitter.com/marijnjh



eloquentjavascript.net