```
function adder(base) {
    return function(no) {
        return base + no;
    }
}

var fiveAdder = adder(5);

fiveAdder(2);
```

Closure

base = 5

```
> tenAdder.prototype
{constructor: ƒ}
  constructor: ƒ (no)
    arguments: null
    caller: null
    length: 1
    name: ""
  prototype: {constructor: ƒ}
    [[FunctionLocation]]: VM336:3
  [[Prototype]]: ƒ ()
  [[Scopes]]: Scopes[2]
    0: Closure (adder) {base: 10}
    1: Global {0: Window, window: Window, self: Window, document: document, name: '', location: Location, …}
```

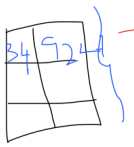
```
> fiveAdder.prototype
{constructor: ƒ}
  constructor: ƒ (no)
    arguments: null
    caller: null
    length: 1
    name: ""
  prototype: {constructor: ƒ}
    [[FunctionLocation]]: VM336:3
  [[Prototype]]: ƒ ()
  [[Scopes]]: Scopes[2]
    0: Closure (adder) {base: 5}
    1: Global {0: Window, window: Window, self: Window, document: document, name: '', location: Location, …}
```

```
function memoize(fn) {
    var cache = {};
    return function(args) {
        if(!cache[args]) {
            cache[args] = fn(args);
        }
        return cache[args];
    }
}
```

```
function fibanocci(no) {
    return (no == 0 || no == 1) ? 1 : fibanocci(no - 1) + fibanocci(no - 2);
}
```

```
var memFib = memoize(fibanocci);
```

```
▼ 0: Closure (memoize)
  ▶ cache: {34: 9227465}
  ▶ fn: ƒ fibanocci(no)
```



```
console.time("first");
console.log(memFib(34));
console.timeEnd("first");
```

```
console.time("second");
console.log(memFib(34));
console.timeEnd("second");
```