

Lesson 238
memo 07.html
プロトタイプオブジェクトの継承 3

```
function Calc(x, y) {
  this.x = x;
  this.y = y;
}

*
Calc.prototype = {
  constructor: Calc,
  name: 'calc',
  func: 'sum',
  arg: function () {console.log(this.x, this.y)},
  add: function () {console.log('Can not!')},
  sub: function () {console.log('Can not!')}
};

Calc.prototype.name = 'calc';
Calc.prototype.func = 'sum';
Calc.prototype.arg = function () {console.log(this.x, this.y)};
Calc.prototype.add = function () {console.log('Can not!')};
Calc.prototype.sub = function () {console.log('Can not!')};

function Calc_n(x, y) {
  this.x = x;
  this.y = y;
}

Calc_n.prototype = Object.create(Calc.prototype);
Calc_n.prototype.constructor = Calc_n;
Calc_n.prototype.name = 'calc_n';
Calc_n.prototype.add = function () {console.log(this.x + this.y)};
Calc_n.prototype.sub = function () {console.log(this.x - this.y)};

function Calc_n1(x, y) {
  this.x = x;
  this.y = y;
}

Calc_n1.prototype = Object.create(Calc.prototype);
Calc_n1.prototype.constructor = Calc_n1;
Calc_n1.prototype.name = 'calc_n1';
Calc_n1.prototype.add = function () {console.log(this.x + * + this.y + * *);};

if (new Calc_n1(1, 2)) {
  console.log('name'); // calc
  console.log('func'); // calc_n
  console.log(''); // 
  console.log(''); // 
  console.log(''); // Can not
}

function Calc_n11(x, y) {
  this.x = x;
  this.y = y;
}

Calc_n11.prototype = Object.create(Calc_n1.prototype);
Calc_n11.prototype.constructor = Calc_n11;
Calc_n11.prototype.name = 'calc_n11';
Calc_n11.prototype.sub = function () {console.log(0)};

if (new Calc_n11(1, 2)) {
  console.log('name'); // calc
  console.log('func'); // calc_n1
  console.log(''); // 
  console.log(''); // 
  console.log(''); // 
  console.log(''); // 0
}

console.log(Calc);
console.log(Calc_n);
console.log(Calc_n1);
```

