The JavaScript Language (Part 2)

Robert M. Dondero, Ph.D.
Princeton University

Objectives

- We will cover:
  - A subset of JavaScript…
  - That is appropriate for COS 333…
  - Through example programs

Agenda

- **Modules**
- Objects

Modules

- See **euclid.js** and **euclidclient2.js**

```
$ node euclidclient2.js
Enter the first integer:
8
Enter the second integer:
12
gcd: 4
lcm: 24
$
```

euclid.js (Page 1 of 1)

```
  1: //-------------------------------------------------------------------------
  2: // euclid.js
  3: // Author: Bob Dondero
  4: //-------------------------------------------------------------------------
  5:
  6: 'use strict';
  7:
  8: //-------------------------------------------------------------------------
  9:
 10: function gcd(i, j) {
 11:
 12:    if ((i === 0) && (j === 0))
 13:       throw new Error('Computation is undefined');
 14:
 15:    i = Math.abs(i);
 16:    j = Math.abs(j);
 17:    while (j !== 0) {
 18:       let temp = i % j;
 19:       i = j;
 20:       j = temp;
 21:    }
 22:    return i;
 23: }
 24:
 25: //-------------------------------------------------------------------------
 26:
 27: function lcm(i, j) {
 28:
 29:    if ((i === 0) || (j === 0))
 30:       throw new Error('Computation is undefined');
 31:
 32:    i = Math.abs(i);
 33:    j = Math.abs(j);
 34:    return (i / gcd(i, j)) * j;
 35: }
 36:
 37: //-------------------------------------------------------------------------
 38:
 39: module.exports = { gcd, lcm };
```

euclidclient2.js (Page 1 of 1)

```
  1: //-------------------------------------------------------------------------
  2: // euclidclient2.js
  3: // Author: Bob Dondero
  4: //-------------------------------------------------------------------------
  5:
  6: 'use strict';
  7:
  8: const readlineSync = require('readline-sync');
  9: const euclid = require('./euclid.js');
 10:
 11: //-------------------------------------------------------------------------
 12:
 13: function readInt(prompt) {
 14:    let line = readlineSync.question(prompt);
 15:    if (line === '')
 16:       throw new Error('Missing integer');
 17:    if (isNaN(line))
 18:       throw new Error('Not a number');
 19:    let n = Number(line);
 20:    if (! Number.isInteger(n))
 21:       throw new Error('Not an integer');
 22:    return n;
 23: }
 24:
 25: //-------------------------------------------------------------------------
 26:
 27: function main() {
 28:    try {
 29:       let i = readInt('Enter the first integer:\n');
 30:       let j = readInt('Enter the second integer:\n');
 31:
 32:       let myGcd = euclid.gcd(i, j);
 33:       process.stdout.write('gcd: ' + String(myGcd) + '\n');
 34:
 35:       let myLcm = euclid.lcm(i, j);
 36:       process.stdout.write('lcm: ' + String(myLcm) + '\n');
 37:    }
 38:    catch (e) {
 39:       process.stderr.write(String(e) + '\n');
 40:    }
 41: }
 42:
 43: if (require.main === module)
 44:    main();
```

Modules

- Kinds of JavaScript modules
  - **Node.js** modules
    - Used by Node.js
    - Not used by browsers
  - **ES6** modules
    - Used in (recent) browsers
    - Not used by Node.js

Agenda

- Modules
- **Objects**

Objects

- Object definition

```
someobj = {
    property1: value1,
    property2: value2,
    …
}
```

Objects

- See **fraction1.js**, **fraction1client.js**

```
$ node fraction1client.js
Numerator 1: 1
Denominator 1: 2
Numerator 2: 3
Denominator 2: 4
f1: 1/2
f2: 3/4
f1 is not identical to f2
f1 is less than f2
-f1: -1/2
f1 + f2: 5/4
f1 - f2: -1/4
f1 * f2: 3/8
f1 / f2: 2/3
$
```

fraction1.js (Page 1 of 2)

```
 1: //-----------------------------------------------------------------------
 2: // fraction1.js
 3: // Author: Bob Dondero
 4: //-----------------------------------------------------------------------
 5:
 6: 'use strict';
 7:
 8: const euclid = require('./euclid.js');
 9:
10: function create(num=0, den=1) {
11:
12:    if (arguments.length > 2)
13:       throw new Error('Too many arguments');
14:
15:    if (den === 0)
16:       throw new Error('Denominator cannot be zero');
17:
18:    let f = {};
19:
20:    f._num = num;
21:    f._den = den;
22:
23:    if (f._den < 0) {
24:       f._num *= -1;
25:       f._den *= -1;
26:    }
27:    if (f._num === 0)
28:       f._den = 1;
29:    else {
30:       let gcden = euclid.gcd(f._num, f._den);
31:       f._num /= gcden;
32:       f._den /= gcden;
33:    }
34:
35:    return f;
36: }
37:
38: function toString(f1) {
39:    return String(f1._num) + '/' + String(f1._den);
40: }
41:
42: function compareTo(f1, f2) {
43:    if ((f1._num * f2._den) < (f2._num * f1._den))
44:       return -1;
45:    if ((f1._num * f2._den) > (f2._num * f1._den))
46:       return 1;
47:    return 0;
48: }
49:
50: function negate(f1) {
51:    return create(-f1._num, f1._den);
52: }
53:
54: function add(f1, f2) {
55:    let newNum = (f1._num * f2._den) + (f2._num * f1._den);
56:    let newDen = f1._den * f2._den;
57:    return create(newNum, newDen);
58: }
59:
60: function subtract(f1, f2) {
61:    let newNum = (f1._num * f2._den) - (f2._num * f1._den);
62:    let newDen = f1._den * f2._den;
63:    return create(newNum, newDen);
64: }
65:
```

fraction1.js (Page 2 of 2)

```
66: function multiply(f1, f2) {
67:    let newNum = f1._num * f2._num;
68:    let newDen = f1._den * f2._den;
69:    return create(newNum, newDen);
70: }
71:
72: function divide(f1, f2) {
73:    let newNum = f1._num * f2._den;
74:    let newDen = f1._den * f2._num;
75:    return create(newNum, newDen);
76: }
77:
78: module.exports = { create, toString, compareTo, negate, add,
79:    subtract, multiply, divide };
```

fraction1client.js (Page 1 of 2)

```
 1: //---------------------------------------------------------------------
 2: // fraction1client.js
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------
 5:
 6: 'use strict';
 7:
 8: const readlineSync = require('readline-sync');
 9: const fraction = require('./fraction1.js');
10:
11: //---------------------------------------------------------------------
12:
13: function readInt(prompt) {
14:    let line = readlineSync.question(prompt);
15:    if (line === '')
16:       throw new Error('Missing integer');
17:    if (isNaN(line))
18:       throw new Error('Not a number');
19:    let n = Number(line);
20:    if (! Number.isInteger(n))
21:       throw new Error('Not an integer');
22:    return n;
23: }
24:
25: //---------------------------------------------------------------------
26:
27: function main() {
28:    try {
29:       let n1 = readInt('Numerator 1: ');
30:       let d1 = readInt('Denominator 1: ');
31:       let n2 = readInt('Numerator 2: ');
32:       let d2 = readInt('Denominator 2: ');
33:
34:       let f1 = fraction.create(n1, d1);
35:       let f2 = fraction.create(n2, d2);
36:
37:       process.stdout.write('f1: ' + fraction.toString(f1) + '\n');
38:       process.stdout.write('f2: ' + fraction.toString(f2) + '\n');
39:
40:       if (f1 === f2)
41:          process.stdout.write('f1 is identical to f2\n');
42:       else
43:          process.stdout.write('f1 is not identical to f2\n');
44:
45:       let compare = fraction.compareTo(f1, f2);
46:       if (compare < 0)
47:          process.stdout.write('f1 is less than f2\n');
48:       if (compare > 0)
49:          process.stdout.write('f1 is greater than f2\n');
50:       if (compare === 0)
51:          process.stdout.write('f1 is equal to f2\n');
52:
53:       let f3;
54:
55:       f3 = fraction.negate(f1);
56:       process.stdout.write('-f1: ' + fraction.toString(f3) + '\n');
57:
58:       f3 = fraction.add(f1, f2);
59:       process.stdout.write('f1 + f2: ' + fraction.toString(f3) + '\n');
60:
61:       f3 = fraction.subtract(f1, f2);
62:       process.stdout.write('f1 - f2: ' + fraction.toString(f3) + '\n');
63:
64:       f3 = fraction.multiply(f1, f2);
65:       process.stdout.write('f1 * f2: ' + fraction.toString(f3) + '\n');
```

fraction1client.js (Page 2 of 2)

```
66:
67:       f3 = fraction.divide(f1, f2);
68:       process.stdout.write('f1 / f2: ' + fraction.toString(f3) + '\n');
69:    }
70:    catch (e) {
71:       process.stderr.write(String(e) + '\n');
72:    }
73: }
74:
75: if (require.main === module)
76:    main();
```

Objects

- **Problem**
  - Instead of calling functions:
    - `f3 = fraction.add(f1, f2);`
  - We want to send messages:
    - `f3 = f1.add(f2);`
- **Solution**
  - The value of an object property can be a function definition…

Objects

- See **fraction2.js**, **fraction2client.js**

```
$ node fraction2client.js
Numerator 1: 1
Denominator 1: 2
Numerator 2: 3
Denominator 2: 4
f1: 1/2
f2: 3/4
f1 is not identical to f2
f1 is less than f2
-f1: -1/2
f1 + f2: 5/4
f1 - f2: -1/4
f1 * f2: 3/8
f1 / f2: 2/3
$
```

fraction2.js (Page 1 of 2)

```
 1: //---------------------------------------------------------------------------
 2: // fraction2.js
 3: // Author: Bob Dondero
 4: //---------------------------------------------------------------------------
 5:
 6: 'use strict';
 7:
 8: const euclid = require('./euclid.js');
 9:
10: function createFraction(num=0, den=1)
11: {
12:    if (arguments.length > 2)
13:       throw new Error('Too many arguments');
14:
15:    if (den === 0)
16:       throw new Error('Denominator cannot be zero');
17:
18:    let f = {};
19:
20:    f._num = num;
21:    f._den = den;
22:
23:    if (f._den < 0) {
24:       f._num *= -1;
25:       f._den *= -1;
26:    }
27:    if (f._num === 0)
28:       f._den = 1;
29:    else {
30:       let gcden = euclid.gcd(f._num, f._den);
31:       f._num /= gcden;
32:       f._den /= gcden;
33:    }
34:
35:    f.toString = function() {
36:       return String(this._num) + '/' + String(this._den);
37:    };
38:
39:    f.compareTo = function(other) {
40:       if ((this._num * other._den) < (other._num * this._den))
41:          return -1;
42:       if ((this._num * other._den) > (other._num * this._den))
43:          return 1;
44:       return 0;
45:    };
46:
47:    f.negate = function() {
48:       return createFraction(-this._num, this._den);
49:    };
50:
51:    f.add = function(other) {
52:       let newNum = (this._num * other._den) + (other._num * this._den);
53:       let newDen = this._den * other._den;
54:       return createFraction(newNum, newDen);
55:    };
56:
57:    f.subtract = function(other) {
58:       let newNum = (this._num * other._den) - (other._num * this._den);
59:       let newDen = this._den * other._den;
60:       return createFraction(newNum, newDen);
61:    };
62:
63:    f.multiply = function(other) {
64:       let newNum = this._num * other._num;
65:       let newDen = this._den * other._den;
```

fraction2.js (Page 2 of 2)

```
66:       return createFraction(newNum, newDen);
67:    };
68:
69:    f.divide = function(other) {
70:       let newNum = this._num * other._den;
71:       let newDen = this._den * other._num;
72:       return createFraction(newNum, newDen);
73:    };
74:
75:    return f;
76: }
77:
78: module.exports = { createFraction };
```

fraction2client.js (Page 1 of 2)

```
1: //-----------------------------------------------------------------------
2: // fraction2client.js
3: // Author: Bob Dondero
4: //-----------------------------------------------------------------------
5:
6: 'use strict';
7:
8: const readlineSync = require('readline-sync');
9: const fraction = require('./fraction2.js');
10:
11: //-----------------------------------------------------------------------
12:
13: function readInt(prompt) {
14:    let line = readlineSync.question(prompt);
15:    if (line === '')
16:       throw new Error('Missing integer');
17:    if (isNaN(line))
18:       throw new Error('Not a number');
19:    let n = Number(line);
20:    if (! Number.isInteger(n))
21:       throw new Error('Not an integer');
22:    return n;
23: }
24:
25: //-----------------------------------------------------------------------
26:
27: function main() {
28:    try {
29:       let n1 = readInt('Numerator 1: ');
30:       let d1 = readInt('Denominator 1: ');
31:       let n2 = readInt('Numerator 2: ');
32:       let d2 = readInt('Denominator 2: ');
33:
34:       let f1 = fraction.createFraction(n1, d1);
35:       let f2 = fraction.createFraction(n2, d2);
36:
37:       process.stdout.write('f1: ' + f1.toString() + '\n');
38:       process.stdout.write('f2: ' + String(f2) + '\n');
39:
40:       if (f1 === f2)
41:          process.stdout.write('f1 is identical to f2\n');
42:       else
43:          process.stdout.write('f1 is not identical to f2\n');
44:
45:       let compare = f1.compareTo(f2);
46:       if (compare < 0)
47:          process.stdout.write('f1 is less than f2\n');
48:       if (compare > 0)
49:          process.stdout.write('f1 is greater than f2\n');
50:       if (compare === 0)
51:          process.stdout.write('f1 is equal to f2\n');
52:
53:       let f3;
54:
55:       f3 = f1.negate();
56:       process.stdout.write('-f1: ' + String(f3) + '\n');
57:
58:       f3 = f1.add(f2);
59:       process.stdout.write('f1 + f2: ' + String(f3) + '\n');
60:
61:       f3 = f1.subtract(f2);
62:       process.stdout.write('f1 - f2: ' + String(f3) + '\n');
63:
64:       f3 = f1.multiply(f2);
65:       process.stdout.write('f1 * f2: ' + String(f3) + '\n');
```

fraction2client.js (Page 2 of 2)

```
66:
67:       f3 = f1.divide(f2);
68:       process.stdout.write('f1 / f2: ' + String(f3) + '\n');
69:    }
70:    catch (e) {
71:       process.stderr.write(String(e) + '\n');
72:    }
73: }
74:
75: if (require.main === module)
76:    main();
```

Objects

- **Problem**:
  - Space inefficiency…

Objects

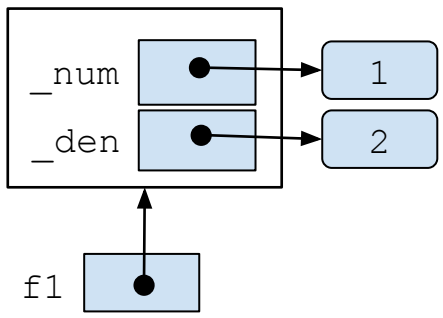
In Python

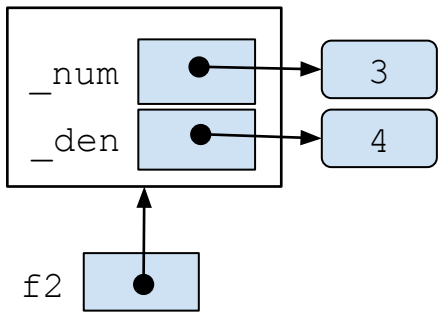
```
f1 = Fraction(1, 2)
f2 = Fraction(3, 4)
```

```
add(self, other):
    …
```

```
sub(self, other):
    …
```

…





Explicit `self` parameter allows `Fraction` objects to share same function defs

Objects

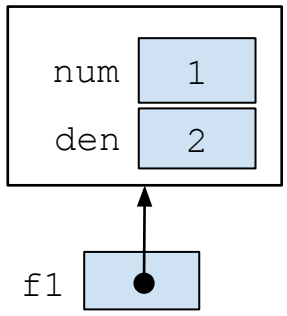
```
Fraction f1 = new Fraction(1, 2);
Fraction f2 = new Fraction(3, 4);
```

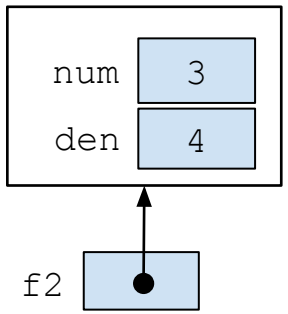
In Java

```
add(this, other)
{…}
```

```
sub(this, other)
{…}
```

…



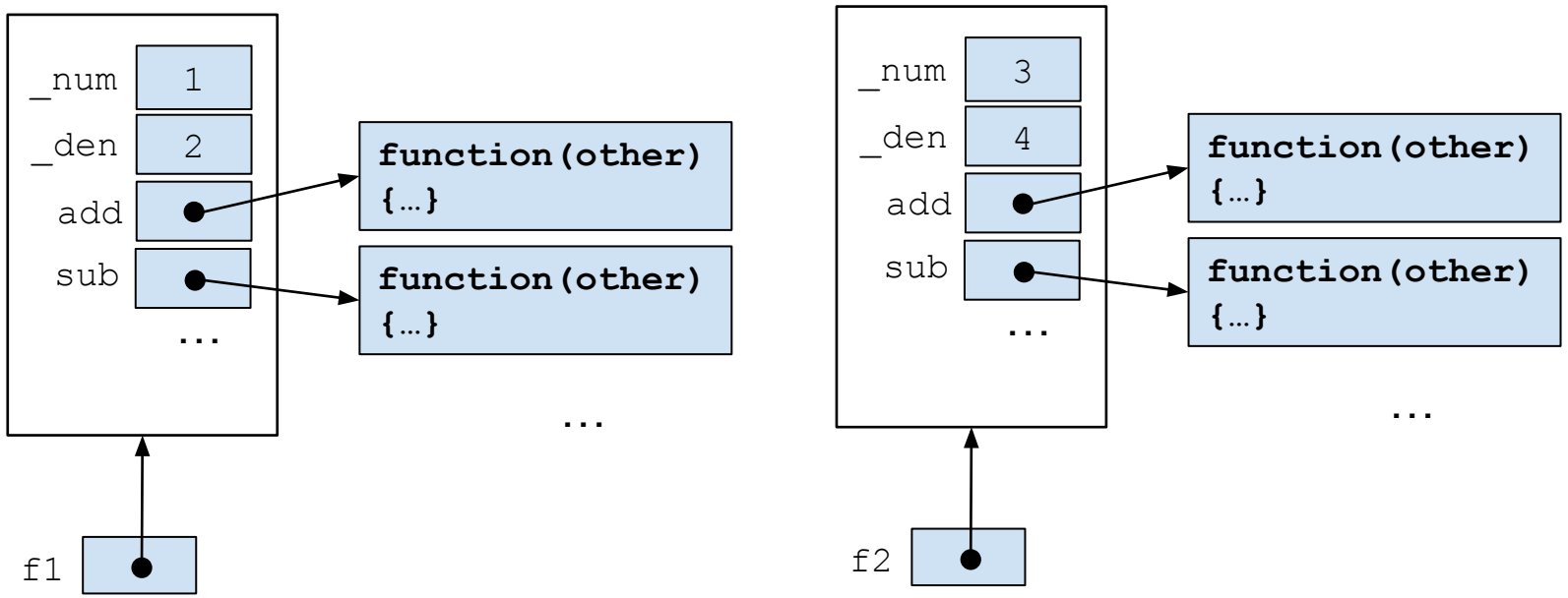


Implicit `this` parameter allows `Fraction` objects to share same method defs

Objects

In JavaScript (so far)

```
let f1 = createFraction(1, 2);
let f2 = createFraction(3, 4);
```



Summary

- We have covered:
  - Modules
  - Objects