

JavaScript Language

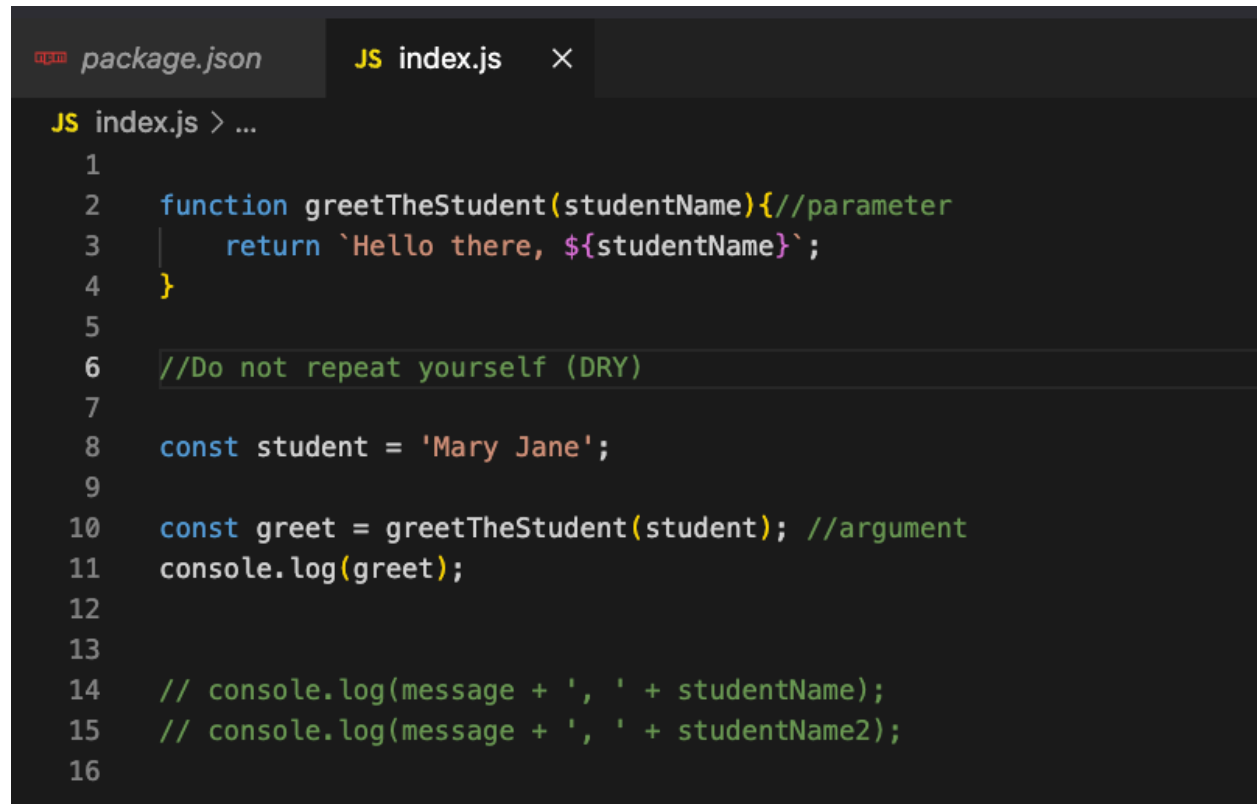
Edmundo Mendez, Jr.

IFT 458

09/03/2023

Introduction to JavaScript Video

In this section of the assignment, we reviewed basic JavaScript functions. Screenshots of work below:



The screenshot shows a code editor with two tabs: `package.json` and `JS index.js`. The `JS index.js` tab is active, showing the following code:

```
1
2  function greetTheStudent(studentName){//parameter
3      return `Hello there, ${studentName}`;
4  }
5
6  //Do not repeat yourself (DRY)
7
8  const student = 'Mary Jane';
9
10 const greet = greetTheStudent(student); //argument
11 console.log(greet);
12
13
14 // console.log(message + ', ' + studentName);
15 // console.log(message + ', ' + studentName2);
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

zsh + - [] ... ^

author:
license: (ISC)
About to write to /Users/mundojr./Documents/College/ASU/2023 ASU FALL/IFT 458/Module 2/Assignment/Project M2-AA/package.json:

{
 "name": "project-m2-aa",
 "version": "1.0.0",
 "description": "instructor code",
 "main": "index.js",
 "scripts": {
 "test": "echo \"Error: no test specified\" && exit 1"
 },
 "author": "",
 "license": "ISC"
}

● mundojr.@Edmundos-MacBook-Pro Project M2-AA % node index.js
Hello there, Mary Jane
○ mundojr.@Edmundos-MacBook-Pro Project M2-AA %

package.json × JS index.js

package.json > ...

1 {
2 "name": "project-m2-aa",
3 "version": "1.0.0",
4 "description": "instructor code",
5 "main": "index.js",
6 "scripts": {
7 "test": "echo \"Error: no test specified\" && exit 1"
8 },
9 "author": "",
10 "license": "ISC"
11 }
12

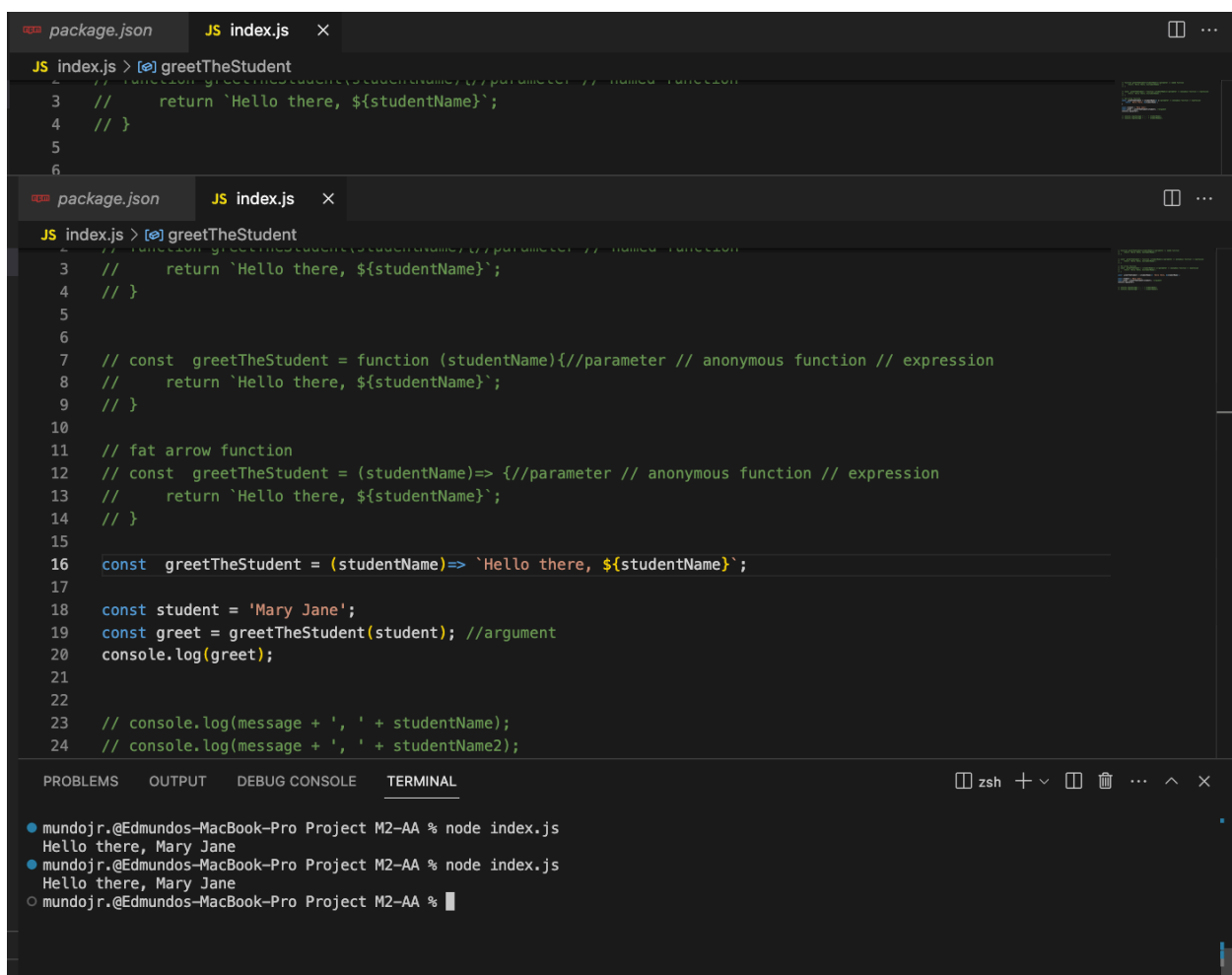
package.json JS index.js

JS index.js > ...

1
2 // function greetTheStudent(studentName){//parameter // named function
3 // return `Hello there, \${studentName}`;
4 // }
5
6
7 const greetTheStudent = function (studentName){//parameter // anonymous function // expression
8 return `Hello there, \${studentName}`;
9 }
10
11 const student = 'Mary Jane';
12 const greet = greetTheStudent(student); //argument
13 console.log(greet);
14
15
16 // console.log(message + ', ' + studentName);
17 // console.log(message + ', ' + studentName2);
18

Refactoring

In this next video, we learn about refactoring in JavaScript using the fat arrow function.



The screenshot shows a VS Code editor with two tabs: `package.json` and `JS index.js`. The `JS index.js` tab is active, showing the following code:

```
1 // function greetTheStudent(studentName) { // parameter // named function
2 //
3 //     return `Hello there, ${studentName}`;
4 // }
5
6
7 // const greetTheStudent = function (studentName) { // parameter // anonymous function // expression
8 //     return `Hello there, ${studentName}`;
9 // }
10
11 // fat arrow function
12 // const greetTheStudent = (studentName) => { // parameter // anonymous function // expression
13 //     return `Hello there, ${studentName}`;
14 // }
15
16 const greetTheStudent = (studentName) => `Hello there, ${studentName}`;
17
18 const student = 'Mary Jane';
19 const greet = greetTheStudent(student); // argument
20 console.log(greet);
21
22
23 // console.log(message + ', ' + studentName);
24 // console.log(message + ', ' + studentName2);
```

The bottom of the image shows the `TERMINAL` panel with the following output:

```
● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node index.js
Hello there, Mary Jane
● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node index.js
Hello there, Mary Jane
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %
```

```

15
16 const greetTheStudent = studentName => `Hello there, ${studentName}`;
17 // const fullName = function(firstName, middleName, lastName) {
18 //     return `${lastName}, ${middleName}. ${firstName}`;
19 // }
20
21 // const fullName = (firstName, middleName, lastName) => {
22 //     return `${lastName}, ${middleName}. ${firstName}`;
23 // }
24
25 const fullName = (firstName, middleName, lastName) => `${lastName}, ${middleName}. ${firstName}`;
26
27 // const student = 'Mary Jane';
28 // const greet = greetTheStudent(student); //argument
29 // console.log(greet);
30
31 const titleName = fullName('Mary', 'R', 'Jane');
32 console.log(titleName);
33
34 // console.log(message + ', ' + studentName);

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

zsh + v [] [] ... ^ x

```

● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node index.js
Jane, R. Mary
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %

```

```

14 // }
15
16 const greetTheStudent = studentName => `Hello there, ${studentName}`;
17
18
19 const fullName = function(firstName, middleName, lastName) {
20 |     return `${lastName}, ${middleName}. ${firstName}`;
21 | }
22
23 // const student = 'Mary Jane';
24 // const greet = greetTheStudent(student); //argument
25 // console.log(greet);
26
27 const titleName = fullName('Mary', 'R', 'Jane');
28 console.log(titleName);
29

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

zsh + v [] [] ... ^ x

```

● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node index.js
Jane, R. Mary
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %

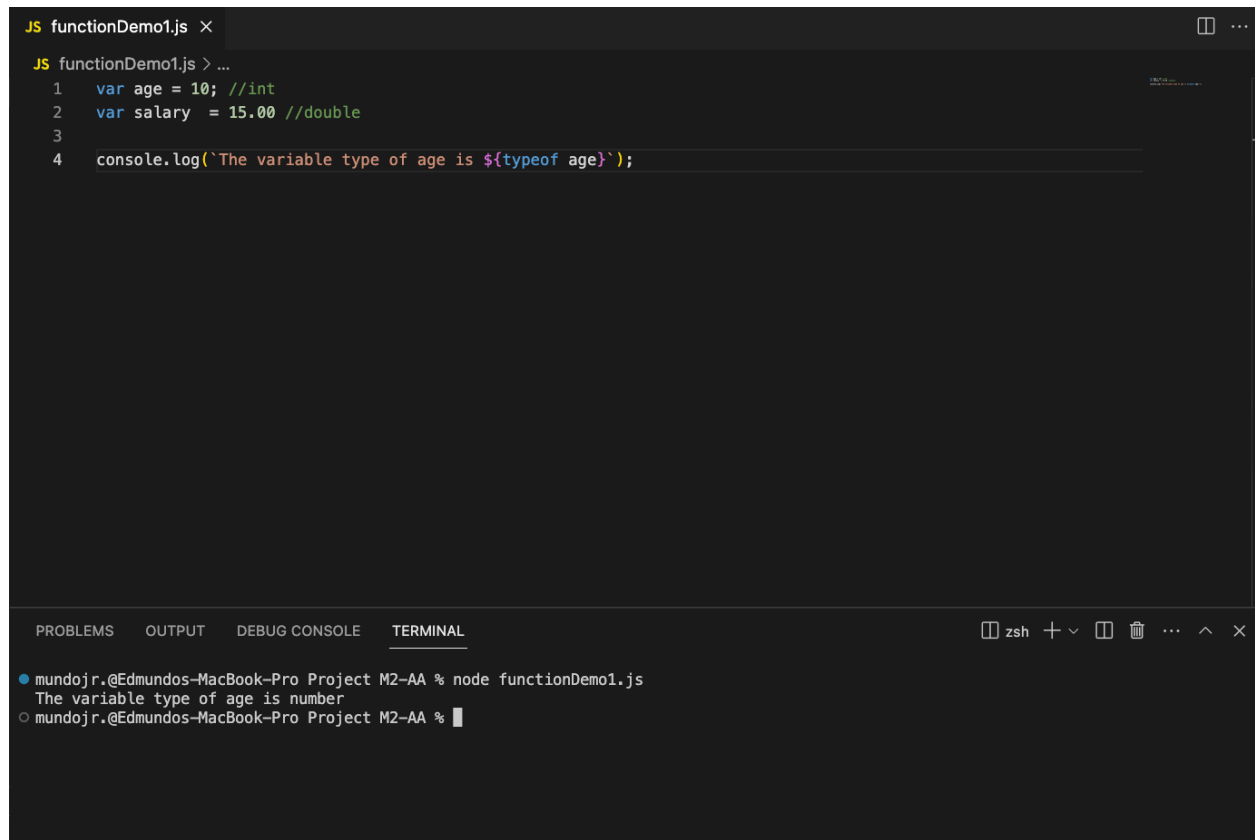
```

```
index.js — Project M2-AA
package.json JS index.js x
JS index.js > [e] greetTheStudent
2 // function greetTheStudent(studentName){//parameter // named function
3 //   return `Hello there, ${studentName}`;
4 // }
5
6
7 // const greetTheStudent = function (studentName){//parameter // anonymous function // expression
8 //   return `Hello there, ${studentName}`;
9 // }
10
11 // fat arrow function
12 // const greetTheStudent = (studentName)=> { //parameter // anonymous function // expression
13 //   return `Hello there, ${studentName}`;
14 // }
15
16 const greetTheStudent = studentName => `Hello there, ${studentName}`;
17
18 const student = 'Mary Jane';
19 const greet = greetTheStudent(student); //argument
20 console.log(greet);
21
22
23 // console.log(message + ', ' + studentName);
24 // console.log(message + ', ' + studentName2);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
• mundojr.@Edmundos-MacBook-Pro Project M2-AA % node index.js
Hello there, Mary Jane
○ mundojr.@Edmundos-MacBook-Pro Project M2-AA %
```

Variables

In this next section, we reviewed variables.



The screenshot shows a code editor with a dark theme. The top panel displays a JavaScript file named `functionDemo1.js` with the following code:

```
1  var age = 10; //int
2  var salary = 15.00 //double
3
4  console.log(`The variable type of age is ${typeof age}`);
```

The bottom panel shows the terminal output for the command `node functionDemo1.js`:

```
● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
  The variable type of age is number
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %
```

```
JS functionDemo1.js X

JS functionDemo1.js > ...
1  var age = 10; //int
2  var salary = 15.00 //double
3
4  console.log(`The variable type of age is ${typeof age}`);
5  console.log(`The variable type of salary is ${typeof salary}`);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
  The variable type of age is number
  The variable type of salary is number
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %
```

```
JS functionDemo1.js X

JS functionDemo1.js > ...
1  var age = 10; //int
2  var salary = 15.00 //double
3
4  console.log(`The variable type of age is ${typeof age}`);
5  console.log(`The variable type of salary is ${typeof salary}`);
6
7  var console: Console ;
8  console.log(`The variable type of name is ${typeof name}`);

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

● mundojr. @Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
  The variable type of age is number
  The variable type of salary is number
  The variable type of name is string
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %
```


JS functionDemo1.js ×

JS functionDemo1.js > ...

```
1  var age = 10; //int
2  var salary = 15.00 //double
3
4  console.log(`The variable type of age is ${typeof age}`);
5  console.log(`The variable type of salary is ${typeof salary}`);
6
7  var name = 'John Smith';
8  console.log(`The variable type of name is ${typeof name}`);
9
10 const displayGreetings = function(name, year){
11     console.log(`Happy new year ${year} ${name}`);
12 }
13
14 console.log(`The variable type of displayGreetings is ${typeof displayGreetings}`);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

- mundojr.@Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
The variable type of age is number
The variable type of salary is number
The variable type of name is string
The variable type of displayGreetings is function
- mundojr.@Edmundos-MacBook-Pro Project M2-AA %

Objects

In this next section, we reviewed objects.

```
JS functionDemo1.js X
JS functionDemo1.js > ...
9
10 const displayGreetings = function(name, year){
11 |   console.log(`Happy new year ${year} ${name}`);
12 | }
13
14 // console.log(`The variable type of displayGreetings is ${typeof displayGreetings}`);
15
16 const displayGreetingsWithEmoji = function(name, year){
17 |   console.log(` 🎉 🎉 Happy new year ${year} ${name} 🎉 🎉 `);
18 | }
19
20 var greet = {};
21
22 if(age > 10){
23 |   greet = displayGreetings;
24 | }
25 else{
26 |   greet = displayGreetingsWithEmoji;
27 | }
28
29 greet('Sam', 2022);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

• mundojr. @Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
🎉 🎉 Happy new year 2022 Sam 🎉 🎉
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %

```
JS functionDemo1.js X
JS functionDemo1.js > ...
6
7 // var name = 'John Smith';
8 // console.log(`The variable type of name is ${typeof name}`);
9
10 const displayGreetings = function(name, year){
11 |   console.log(`Happy new year ${year} ${name}`);
12 | }
13
14 // console.log(`The variable type of displayGreetings is ${typeof displayGreetings}`);
15
16 const displayGreetingsWithEmoji = function(name, year){
17 |   console.log(` 🎉 🎉 Happy new year ${year} ${name} 🎉 🎉 `);
18 | }
19
20 const greet = function(name, year, func){
21 |   func(name, year);
22 | }
23
24 greet('Sam', 2022, displayGreetingsWithEmoji);
25 greet('Jane', 2022, displayGreetings);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

• mundojr. @Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
🎉 🎉 Happy new year 2022 Sam 🎉 🎉
Happy new year 2022 Jane
○ mundojr. @Edmundos-MacBook-Pro Project M2-AA %

```
objects.js — Assignment01
Welcome JS functionDemo1.js JS objects.js M X
JS objects.js > ...
You, 1 second ago | 1 author (You)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 console.log(students[0]);
4 console.log(students[2]); You, 1 second ago • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• mundojr.@Edmundos-MacBook-Pro Assignment01 % node objects.js
John
• mundojr.@Edmundos-MacBook-Pro Assignment01 % node objects.js
John
Jamie
○ mundojr.@Edmundos-MacBook-Pro Assignment01 %
```

```
Welcome JS functionDemo1.js JS objects.js M X
JS objects.js > ...
You, 1 second ago | 1 author (You)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 console.log(students[0]);
4 console.log(students[2]);
5
6 console.log(students.length); You, now • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
John
• mundojr.@Edmundos-MacBook-Pro Assignment01 % node objects.js
John
Jamie
• mundojr.@Edmundos-MacBook-Pro Assignment01 % node objects.js
John
Jamie
5
○ mundojr.@Edmundos-MacBook-Pro Assignment01 %
```

```
objects.js — Assignment01
Welcome JS functionDemo1.js JS objects.js M X
JS objects.js > ...
You, 1 second ago | 1 author (You)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 console.log(students[0]);
4 console.log(students[2]);
5
6 console.log(students.length);
7
8 var grades = [90, 99, 89, 100, 80];
9
10 console.log(grades[0]);
11 console.log(grades[2]);
12
13 console.log(grades.length);
You, 1 second ago • Uncommitted changes

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
● mundojr. @Edmundos-MacBook-Pro Assignment01 % node objects.js
John
Jamie
5
90
89
5
○ mundojr. @Edmundos-MacBook-Pro Assignment01 %
```

```
Welcome JS functionDemo1.js JS objects.js M X
JS objects.js > ...
You, 1 second ago | 2 authors (You and others)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 // console.log(students[0]);
4 // console.log(students[2]);
5
6 // console.log(students.length);
You, 1 second ago • Uncommitted changes
7
8 var grades = [90, 99, 89, 100, 80];
9
10 // console.log(grades[0]);
11 // console.log(grades[2]);
12
13 console.log(grades.length);
14
15 console.log(grades[grades.length - 1]);
16 console.log(grades[grades.length - 2]);
17
18

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
● mundojr. @Edmundos-MacBook-Pro Assignment01 % node objects.js
5
80
100
○ mundojr. @Edmundos-MacBook-Pro Assignment01 %
```

Arrays

In this next section, we reviewed arrays.

```
student.js — Assignment01
JS functionDemo1.js JS objects.js JS student.js U X

JS student.js > ...
1  var student = {
2    name: 'Joan Smith',
3    birthYear: 2002,
4    course: 'IFT 458',
5    grade: 90
6  };
7
8  console.log(student['name']);
9  console.log(student.name);

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
Joan Smith
• mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
Joan Smith
Joan Smith
○ mundojr. @Edmundos-MacBook-Pro Assignment01 %
```

```
JS functionDemo1.js JS objects.js JS student.js X

JS student.js > ...
1  var student = {
2    name: 'Joan Smith',
3    birthYear: 2002,
4    course: 'IFT 458',
5    grade: 90,
6    age: function() {
7      return 2022 - this.birthYear;
8    }
9  };
10
11 var student2 = {
12   name: 'Andy Smith',
13   birthYear: 2000,
14   course: 'IFT 458',
15   grade: 100,
16   age: function() {
17     return 2022 - this.birthYear;
18   }
19 };
20
21 // console.log(student['name']);
22 // console.log(student.name);
23 console.log(student2.age());
24 console.log(student.age());

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
• mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
22
20
○ mundojr. @Edmundos-MacBook-Pro Assignment01 %
```

student.js — Assignment01

JS functionDemo1.jsJS objects.jsJS student.js M X

JS student.js > ...

```
1  var student = {
2    name: "Joan Smith",
3    birthYear: 2002,
4    course: "IFT 458",
5    grade: 90,
6    active: true,
7    age: function () {
8      return 2022 - this.birthYear;
9    },
10 };
11
12 var student2 = {
13   name: "Andy Smith",
14   birthYear: 2000,
15   course: "IFT 458",
16   grade: 100,
17   active: false,
18   age: function () {
19     if (this.active) {
20       return 2022 - this.birthYear;
21     } else {
22       return 0;
23     }
24   },
25 };

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

- mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
- 0
- 20
- mundojr. @Edmundos-MacBook-Pro Assignment01 %

Object Arrays

In this last section, we reviewed object arrays

```
student.js — Assignment01
JS functionDemo1.js JS objects.js JS student.js U X
JS student.js > ...
12     name: 'Andy Smith',
13     birthYear: 2000,
14     course: 'IFT 458',
15     grade: 100,
16     age: function() {
17         return 2022 - this.birthYear;
18     }
19 };
20
21 // console.log(student['name']);
22 // console.log(student.name);
23 console.log(student2.age());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
● mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
22
○ mundojr. @Edmundos-MacBook-Pro Assignment01 %
```

```
JS functionDemo1.js JS objects.js JS student.js M X
JS student.js > ...
20     age: function () {
21         if (this.active) {
22             return 2022 - this.birthYear;
23         } else {
24             return 0;
25         }
26     },
27 };
28
29 students.push(student);
30 students.push(student2);
31
32 console.log(students);
33 // console.log(student['name']);
34 // console.log(student.name);
35 console.log(student2.age());
36 console.log(student.age());
37
```

Debug Console (⌘⇧Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

```
name: 'Joan Smith',
birthYear: 2002,
course: 'IFT 458',
grade: 90,
active: true,
age: [Function: age]
},
{
  name: 'Andy Smith',
  birthYear: 2000,
  course: 'IFT 458',
  grade: 100,
  active: false,
  age: [Function: age]
}
]
```

```
JS functionDemo1.js JS objects.js JS student.js M X
JS student.js > ...
26 | },
27 | };
28 |
29 | students.push(student);
30 | students.push(student2);
31 | students.forEach((item) => console.log(item.age())); You, 1 second ago • Uncommitted changes
32 | console.log(students);
33 | // console.log(student['name']);
34 | // console.log(student.name);
35 | console.log(student2.age());
36 | console.log(student.age());
37 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
mundojr. @Edmundos-MacBook-Pro Assignment01 % node student.js
20
0
[
  {
    name: 'Joan Smith',
    birthYear: 2002,
    course: 'IFT 458',
    grade: 90,
    active: true,
    age: [Function: age]
  },
  {
    name: 'Andy Smith',
    birthYear: 2000,
    course: 'IFT 458',
    grade: 100,
    active: false,
    age: [Function: age]
  }
]
0
20
```

```
student.js — Assignment01
JS functionDemo1.js JS objects.js JS student.js M X
JS student.js > ...
26 | },
27 | };
28 |
29 | students.push(student);
30 | students.push(student2);
31 | students.forEach((item) => console.log(item.age()));
32 |
33 | console.log(students); You, now • Uncommitted changes
34 |
35 | // console.log(student['name']);
36 | // console.log(student.name);
37 | // console.log(student2.age());
38 | // console.log(student.age());
39 |

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS
20
0
[
  {
    name: 'Joan Smith',
    birthYear: 2002,
    course: 'IFT 458',
    grade: 90,
    active: true,
    age: [Function: age]
  },
  {
    name: 'Andy Smith',
    birthYear: 2000,
    course: 'IFT 458',
    grade: 100,
    active: false,
    age: [Function: age]
  }
]
}
```