JavaScript Language

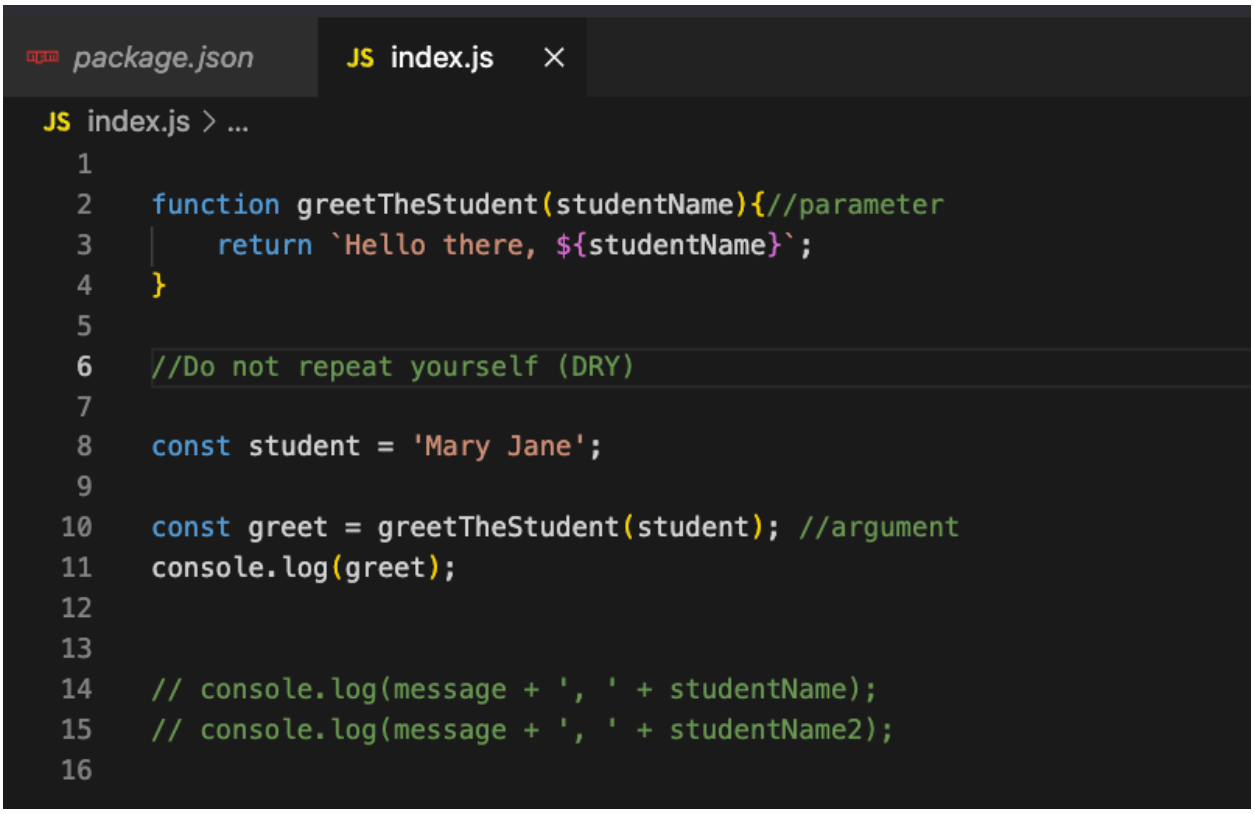
Edmundo Mendez, Jr.

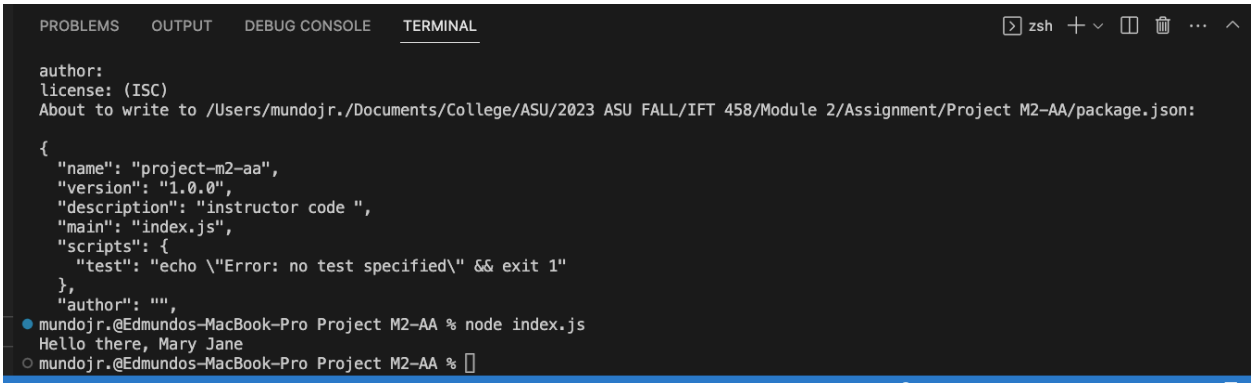
IFT 458

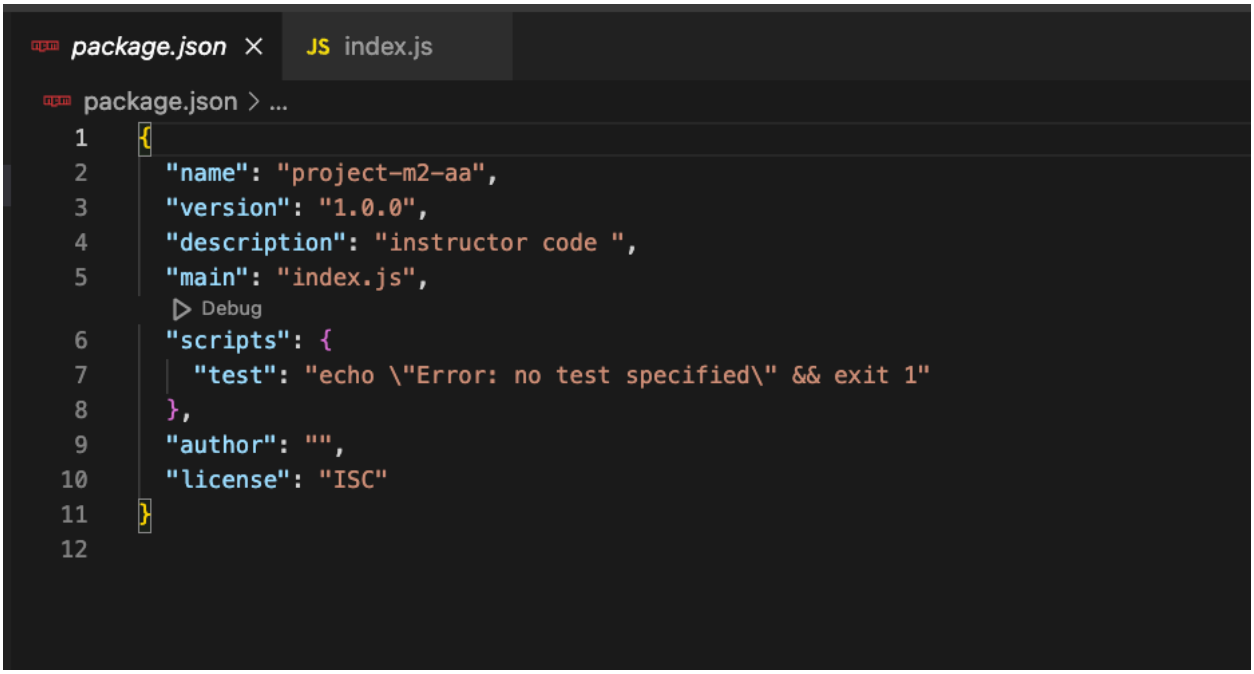
09/03/2023

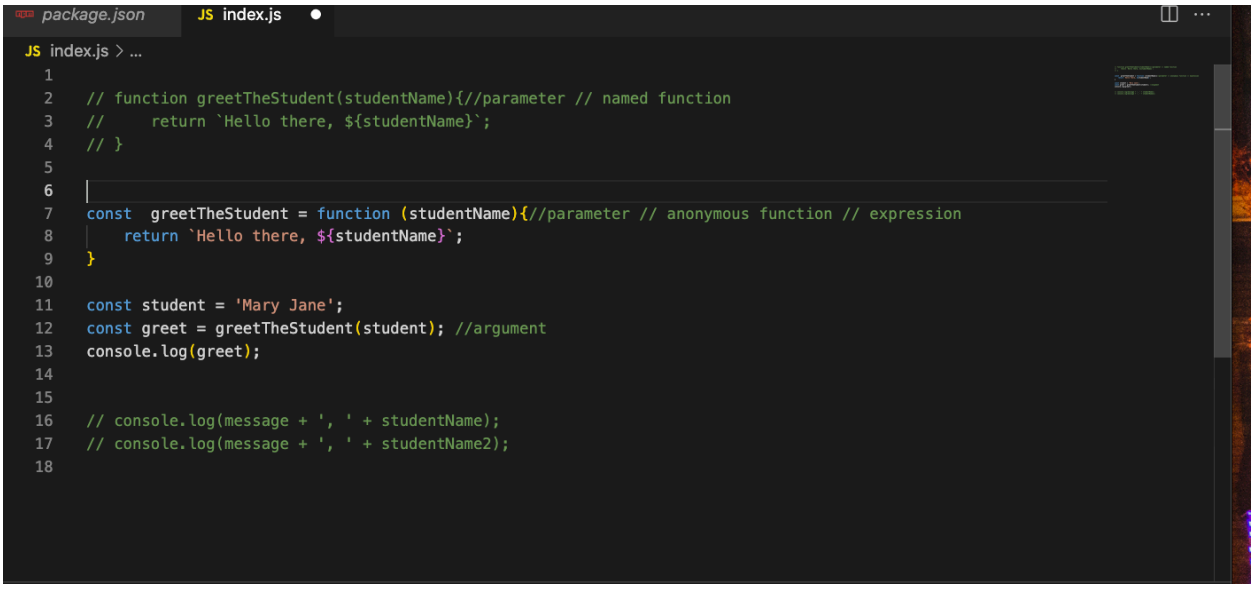
Introduction to JavaScript Video

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



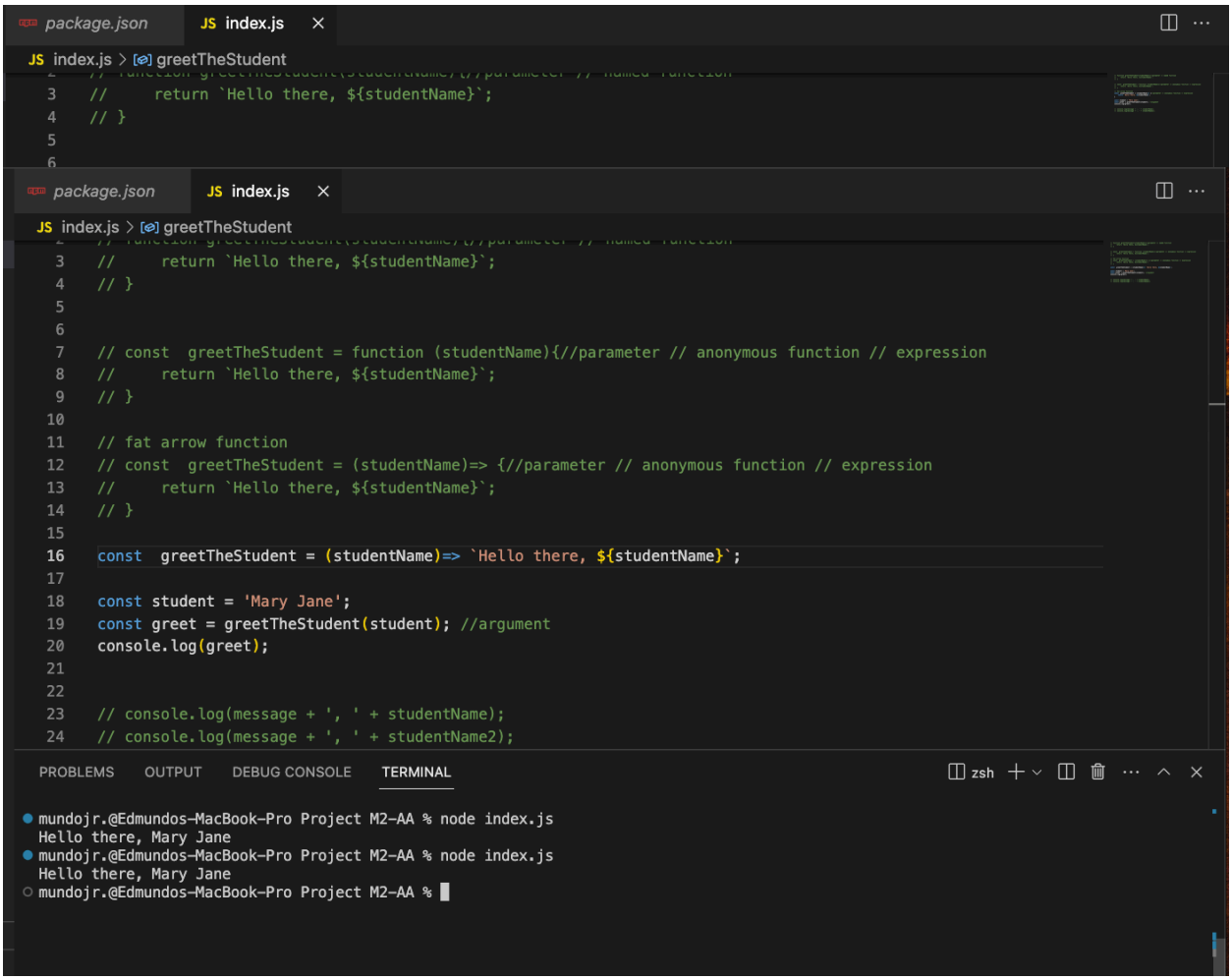

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
zsh
author:
license: (ISC)
About to write to /Users/mundojr./Documents/College/ASU/2023 ASU FALL/IFT 458/Module 2/Assignment/Project M2-AA/package.json:
{
"name": "project-m2-aa",
"version": "1.0.0",
"description": "instructor code ",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1"
},
"author": "",
mundojr.@Edmundos-MacBook-Pro Project M2-AA % node index.js
Hello there, Mary Jane
mundojr.@Edmundos-MacBook-Pro Project M2-AA %


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


package.json
index.js
index.js > ...
// function greetTheStudent(studentName){//parameter // named function
// return `Hello there, ${studentName}`;
// }
const greetTheStudent = function (studentName){//parameter // anonymous function // expression
return `Hello there, ${studentName}`;
}
const student = 'Mary Jane';
const greet = greetTheStudent(student); //argument
console.log(greet);
// console.log(message + ', ' + studentName);
// console.log(message + ', ' + studentName2);

Refactoring

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



```
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);
```

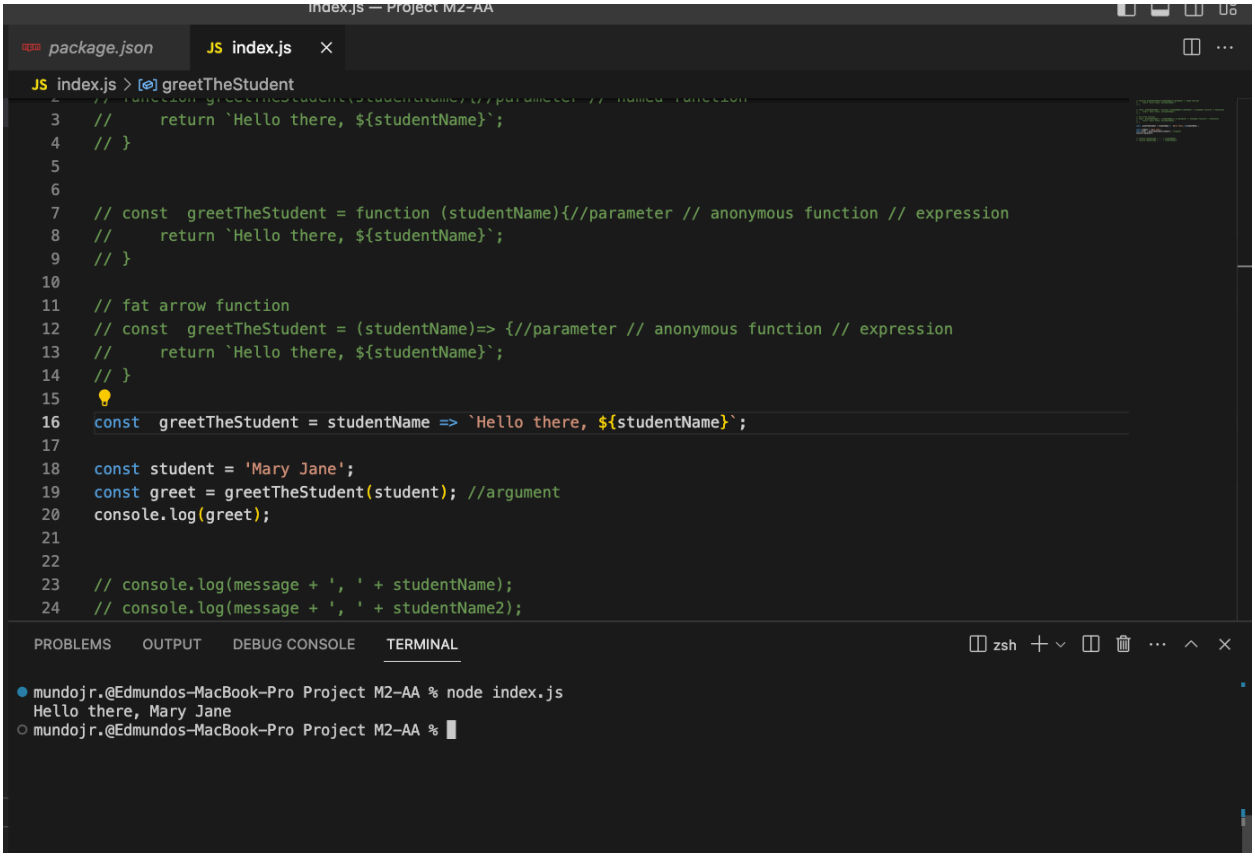
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh

```
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
```

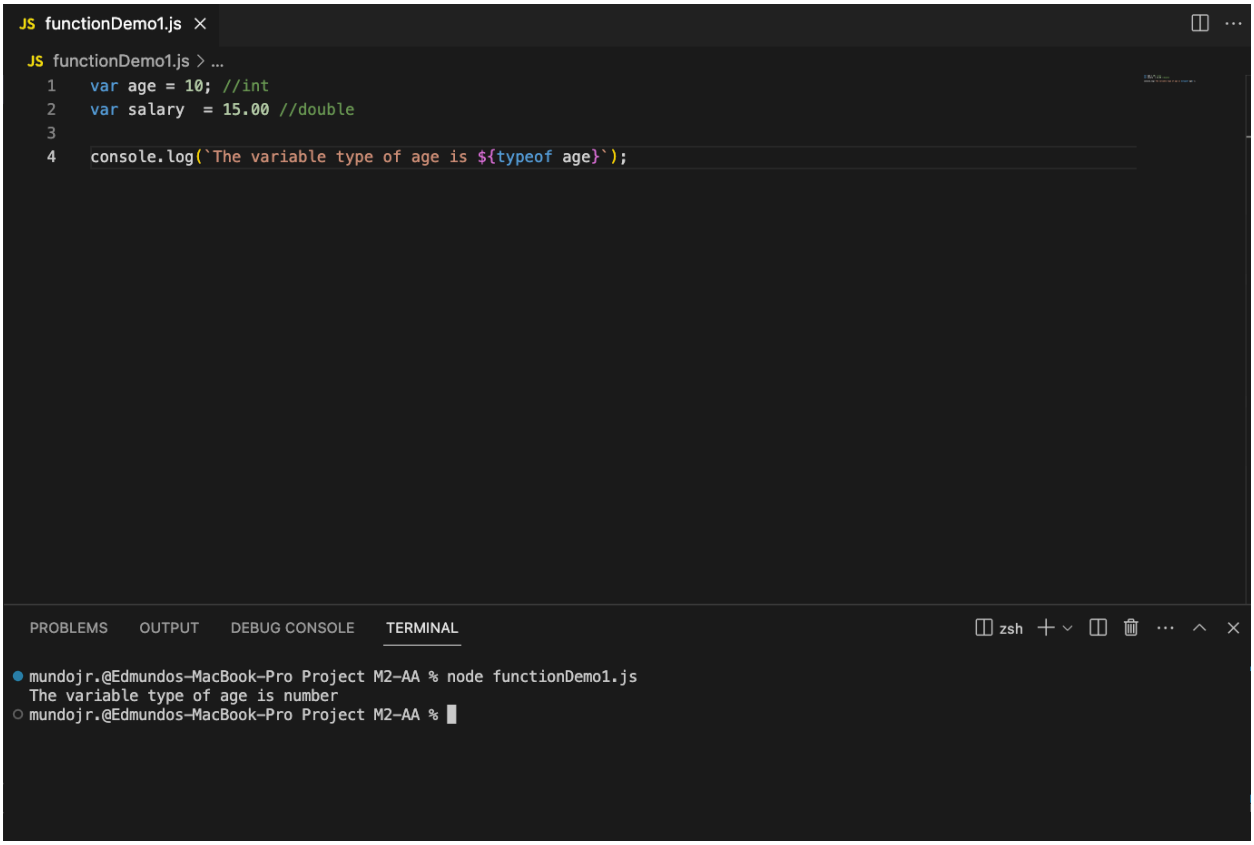
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh

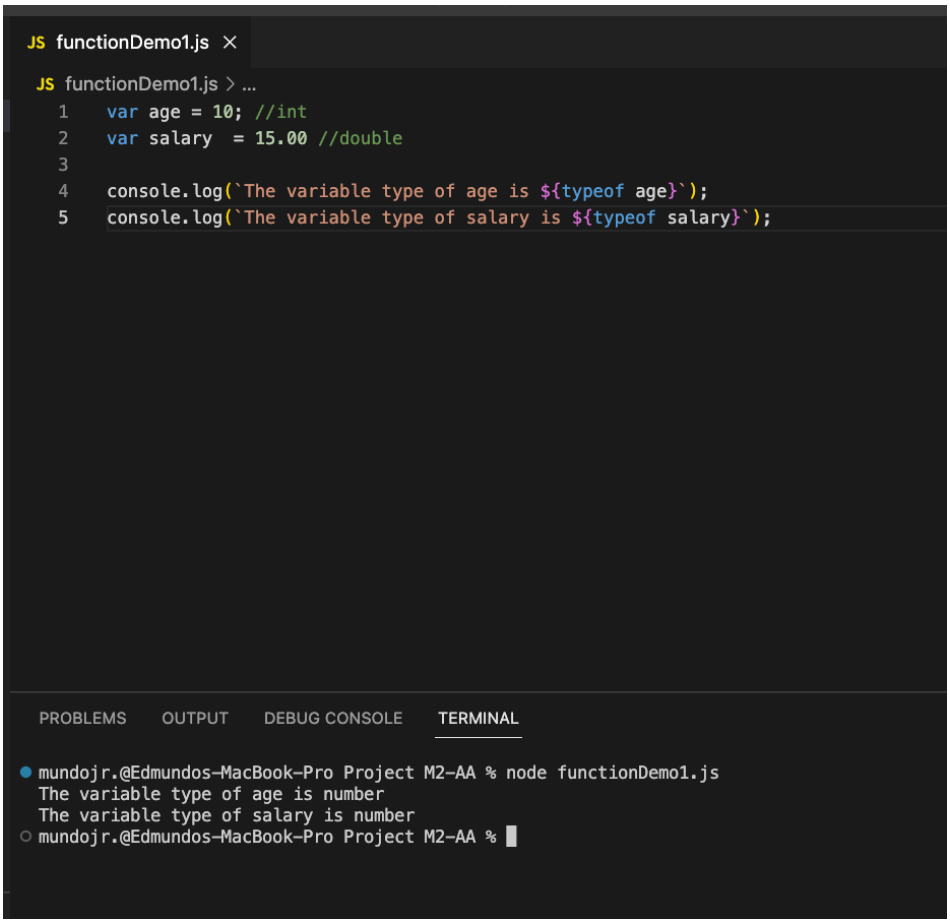
```
mundojr.@Edmundos-MacBook-Pro Project M2-AA % node index.js
Jane, R. Mary
mundojr.@Edmundos-MacBook-Pro Project M2-AA %
```


index.js — Project M2-AA
package.json
index.js
index.js > greetTheStudent
// function greetTheStudent(studentName){//parameter // named function
//   return `Hello there, ${studentName}`;
// }
// const greetTheStudent = function (studentName){//parameter // anonymous function // expression
//   return `Hello there, ${studentName}`;
// }
// fat arrow function
// const greetTheStudent = (studentName)=> {//parameter // anonymous function // expression
//   return `Hello there, ${studentName}`;
// }
const greetTheStudent = studentName => `Hello there, ${studentName}`;
const student = 'Mary Jane';
const greet = greetTheStudent(student); //argument
console.log(greet);
// console.log(message + ', ' + studentName);
// console.log(message + ', ' + studentName2);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
zsh
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.




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}`);
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
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

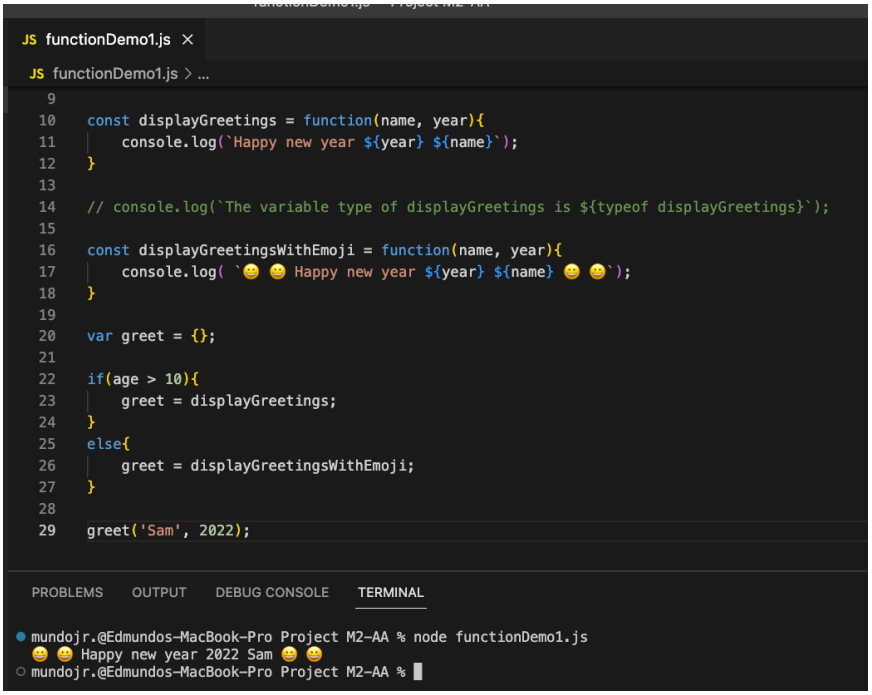
```
JS functionDemo1.js > ...
1   var age = 10; //int
2   var salary  = 15.00 //double
3
4   console.log(`The variable type of age is ${typeof age}`);
5   console.log(`The variable type of salary is ${typeof salary}`);
6
7   var name = 'John Smith';
8   console.log(`The variable type of name is ${typeof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

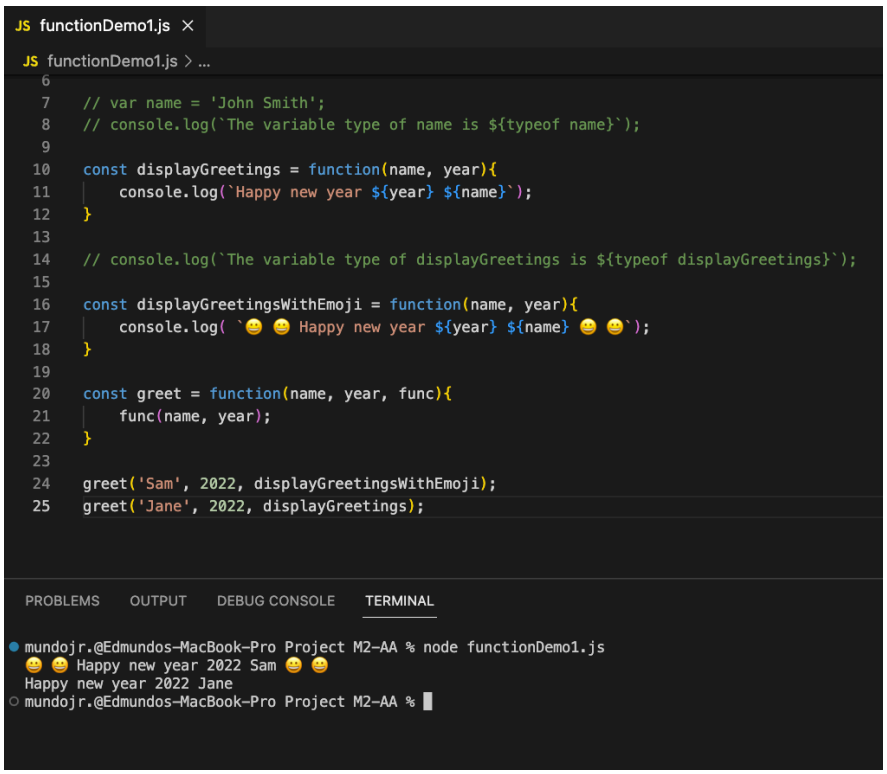
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

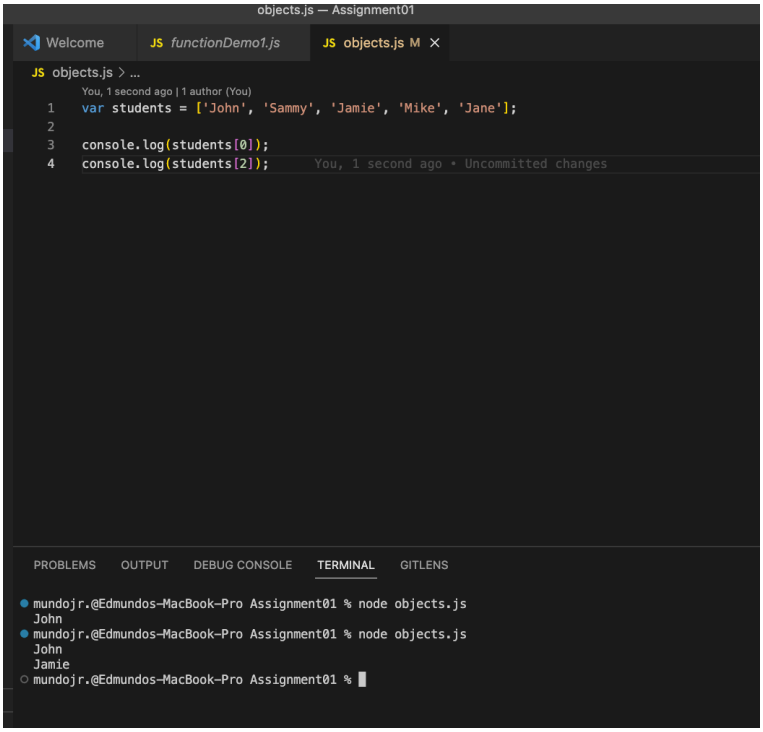
```
mundojr.@Edmundos-MacBook-Pro Project M2-AA % node functionDemo1.js
The variable type of age is number
The variable type of salary is number
The variable type of name is string
The variable type of displayGreetings is function
mundojr.@Edmundos-MacBook-Pro Project M2-AA %
```

Objects

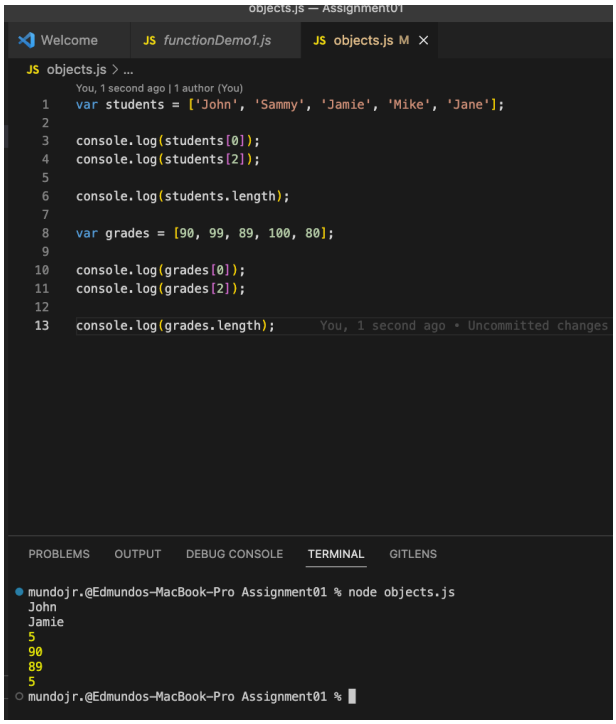
In this next section, we reviewed objects.

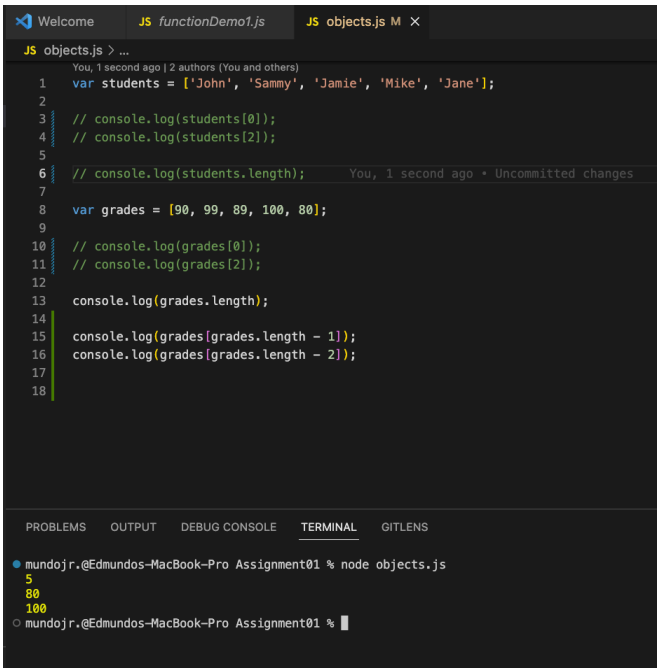





objects.js — Assignment01
Welcome
functionDemo1.js
objects.js M
objects.js > ...
You, 1 second ago | 1 author (You)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 console.log(students[0]);
4 console.log(students[2]);
You, 1 second ago • Uncommitted changes
PROBLEMS
OUTPUT
DEBUG CONSOLE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

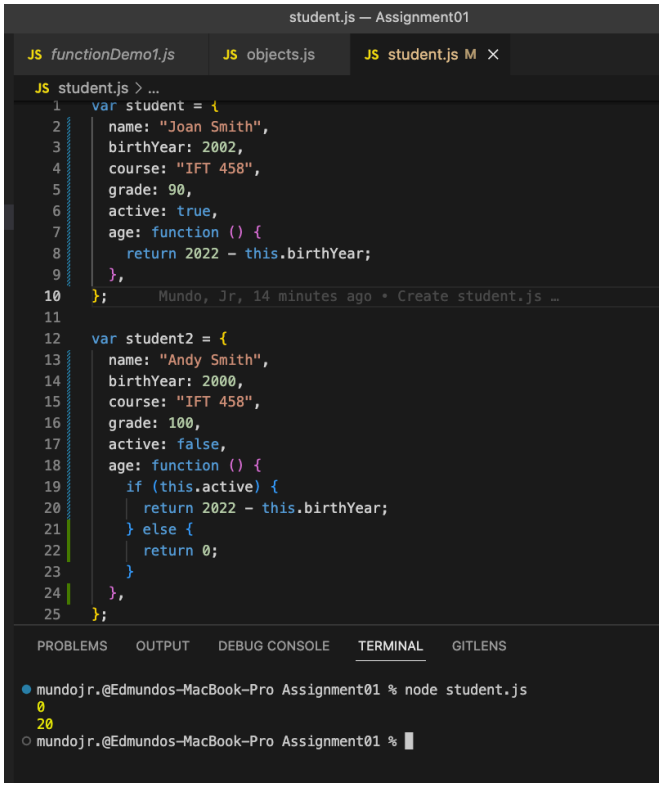

Welcome
functionDemo1.js
objects.js M
objects.js > ...
You, 1 second ago | 1 author (You)
1 var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
2
3 console.log(students[0]);
4 console.log(students[2]);
5
6 console.log(students.length);
You, now • Uncommitted change
PROBLEMS
OUTPUT
DEBUG CONSOLE
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
functionDemo1.js
objects.js M
objects.js > ...
You, 1 second ago | 1 author (You)
var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
console.log(students[0]);
console.log(students[2]);
console.log(students.length);
var grades = [90, 99, 89, 100, 80];
console.log(grades[0]);
console.log(grades[2]);
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
functionDemo1.js
objects.js M
objects.js > ...
You, 1 second ago | 2 authors (You and others)
var students = ['John', 'Sammy', 'Jamie', 'Mike', 'Jane'];
// console.log(students[0]);
// console.log(students[2]);
// console.log(students.length);
You, 1 second ago • Uncommitted changes
var grades = [90, 99, 89, 100, 80];
// console.log(grades[0]);
// console.log(grades[2]);
console.log(grades.length);
console.log(grades[grades.length - 1]);
console.log(grades[grades.length - 2]);
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
mundojr.@Edmundos-MacBook-Pro Assignment01 % node objects.js
5
80
100
mundojr.@Edmundos-MacBook-Pro Assignment01 %

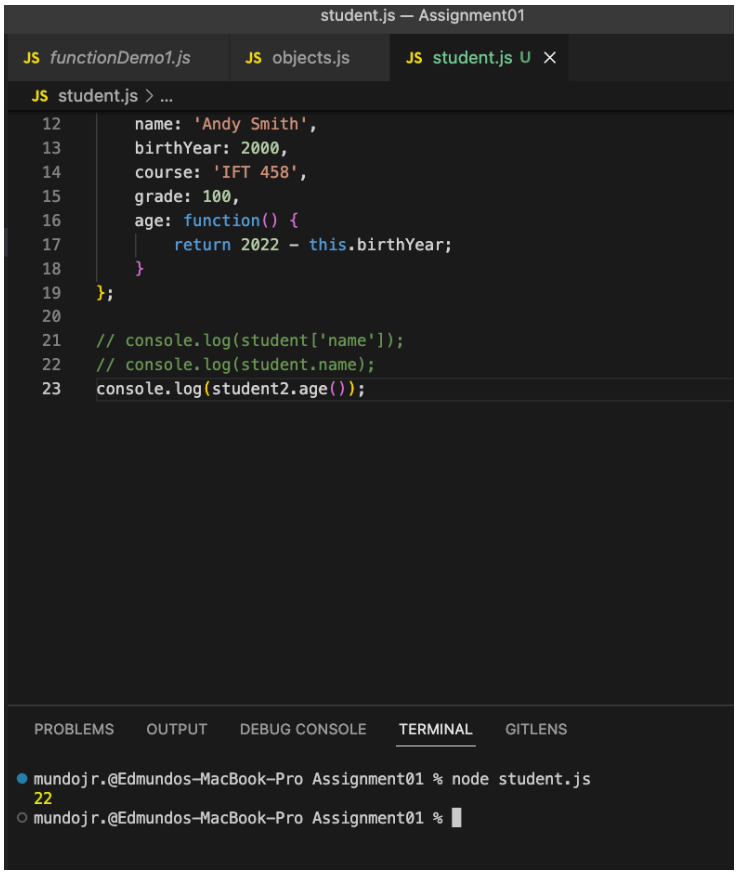
Arrays

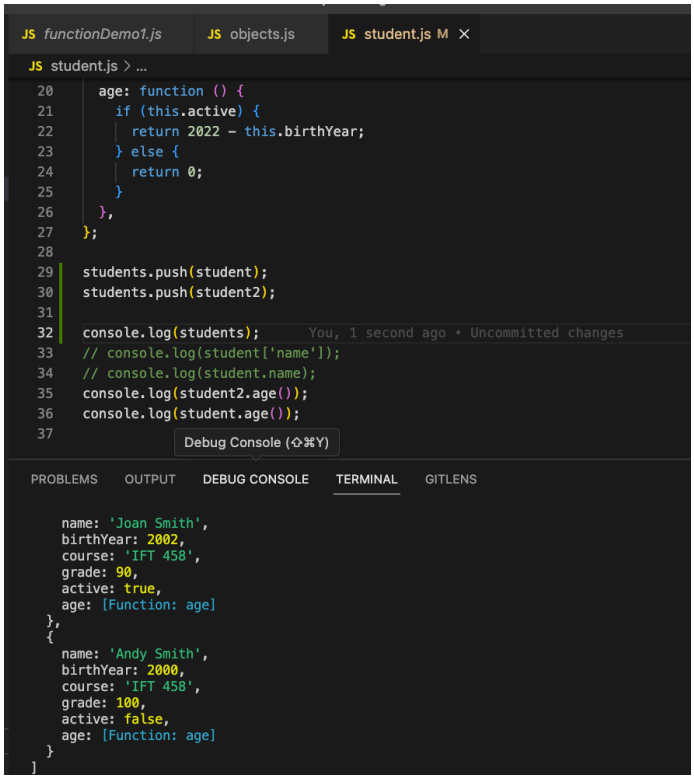
In this next section, we reviewed arrays.

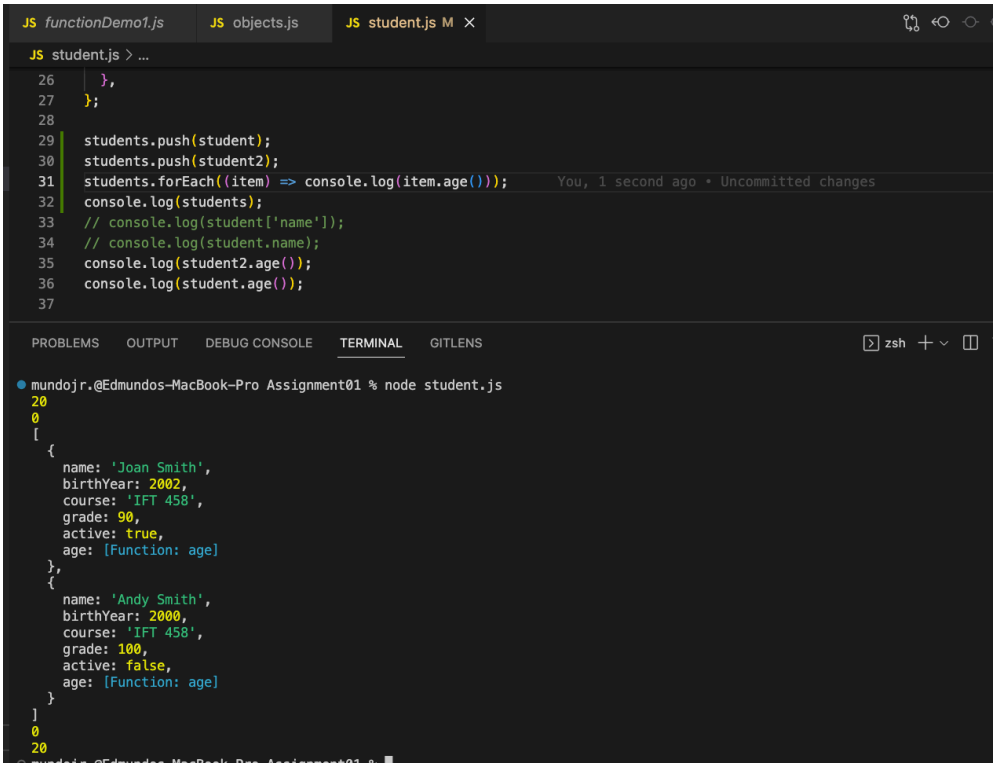

student.js — Assignment01
functionDemo1.js
objects.js
student.js M
student.js > ...
var student = {
name: "Joan Smith",
birthYear: 2002,
course: "IFT 458",
grade: 90,
active: true,
age: function () {
return 2022 - this.birthYear;
},
};
Mundo, Jr, 14 minutes ago • Create student.js …
var student2 = {
name: "Andy Smith",
birthYear: 2000,
course: "IFT 458",
grade: 100,
active: false,
age: function () {
if (this.active) {
return 2022 - this.birthYear;
} else {
return 0;
}
},
};
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
GITLENS
mundojr.@Edmundos-MacBook-Pro Assignment01 % node student.js
0
20
mundojr.@Edmundos-MacBook-Pro Assignment01 %

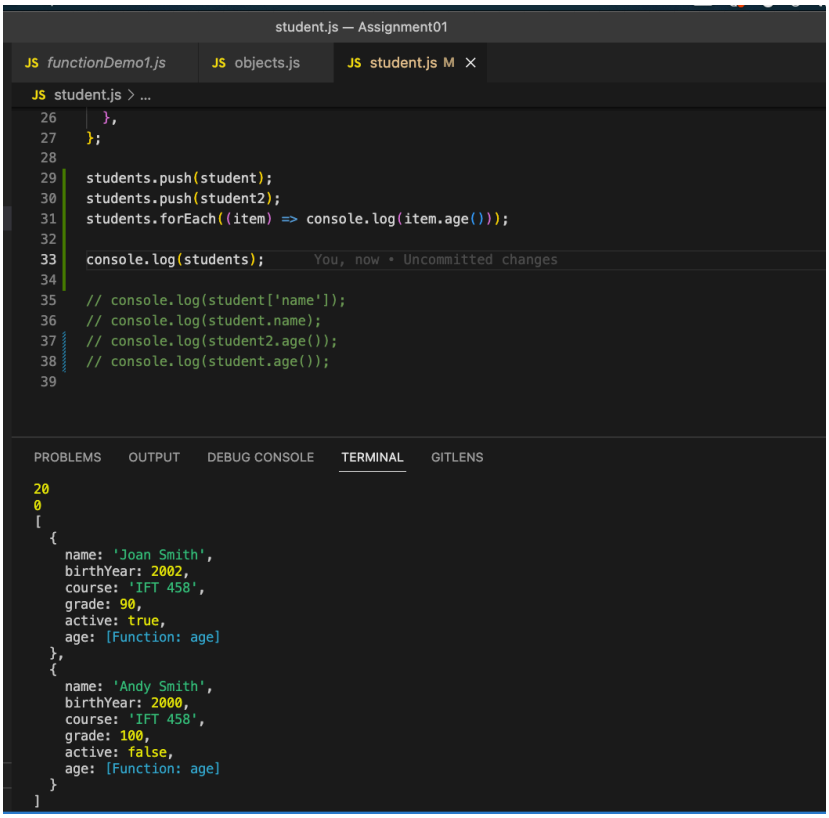
Object Arrays

In this last section, we reviewed object arrays






functionDemo1.js
objects.js
student.js M
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
zsh
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
functionDemo1.js
objects.js
student.js M
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]
}
]