JavaScript Destructuring

Learning Objectives

- Understand how to use JavaScript destructuring

Material

Destructuring Objects

Destructuring is a useful shorthand for dealing with objects in JavaScript. We can quickly pluck properties from objects into variables like so:

```
let user = {
    username: "hello123",
    email: "hello123@multiverse.io",
    daysSinceActive: 7,
    verified: false,
};
const { username, verified } = user;
console.log(username); // "hello123"
console.log(verified); // false
```

We can also use this in function parameters:

```
let person = {
    firstname: "Mike",
    surname: "Pound",
    job: "Academic",
};

function fullName({ firstname, surname }) {
    return firstname + " " + surname;
}
console.log(fullName(person));
// Mike Pound
```

We can also reverse this process: taking variables and placing them as properties on an object:

```
let username = "hello123";
let age = 76;

let user = { username, age };
console.log(user);
// { username: 'hello123', age: 76 }
```

Destructuring Arrays

Arrays can also be destructured:

```
let [gold, silver] = ["Bolt", "Thompson", "Dix", "Martina"];
console.log(gold); // Bolt
console.log(silver); // Thompson
```

Core Assignments

Destructuring Quiz

For each of the following code snippets, predict the output. Some of the examples might throw errors. After guessing each one, try running the code to see if your answer was correct.

```
let teacher = {
    name: "Walter",
    subject: "Chemistry",
};
console.log(name);
```

```
let teacher = {
    name: "Walter",
    subject: "Chemistry",
};
const { name } = teacher;
console.log(name);
```

```
const animals = ["Red Panda", "Otter", "Raccoon"];
const [pet] = animals;
console.log(pet);
```

```
function shouldDelete({ admin, verified }) {
    return !admin && !verified;
}

let user = {
    id: 8732429,
    admin: true,
    verified: false,
};

console.log(shouldDelete(user));
```

```
const [a, b, c] = [3, 1, 4, 1, 5];
console.log(c);
```

```
let a = 6;
let b = 22;

console.log({ a, b });
```