Array with a Skipped Value

This lesson goes over how to skip a value on an array

An array doesn't require all their indices to be defined sequentially.

```
const arr = [0, 1, 2, 3, 4]; // Unskipped array
```

When an index is skipped, it is undefined by default. There are two ways to have an array that skips elements.

The **first** is when the array is defined. Elements defined between brackets can skip a single element by having two commas next to each other. In the following example, **line 1** defines a skipped array. It starts with `0`, then skips the expected `1, 2`.

```
const arrskip1 = [0, , , 3, 4]; // Skipped array

arrskip1.forEach(i => {
    console.log("{" + i + "}");
});
console.log("This is undefined:" + arrskip1[1]);
```

The **second** one is to access the index on a defined array. The next example defines an empty array in **line 1**. The next lines define the indices `0` and `2` but skips `1`.

```
const arrskip2 = [];

arrskip2[0] = 100;
arrskip2[2] = 200;
arrskip2.forEach(i => {
    console.log("{" + i + "}");
});
console.log("This is undefined:" + arrskip2[1]);
```

TypeScript, as in JavaScript, can skip values in an array.