THESE THINGS "EXIST"

1 PRIVATE FIELDS

Private properties for Classes are natively supported in JavaScript.

Just prefix the property (fields and methods) with a # and they become inaccessible outside the Class

```
class MyClass {
  #privateField = 120;

  #privateMethod() {
    // ...
  }
}
```

2 TO SORTED

The `Array#sort()` method sorts an array in place (modifies the arrary)

A new method `Array#toSorted()` returns a new array instead in the sorted order.

```
const origArray = [7, 6, 8, 5];
const newArray =
  origArray.toSorted(comparator);

origArray; // [7, 6, 8, 5]
newArray; // [5, 6, 7, 8]
```

3 OPTIONAL CHAINING

When accessing a property using the dot operator, if the operand is null or undefined, an error is thrown.

This can be avoided by using the optional chaining **?.** operator.

```
const user = { name: 'Maddy' };

user.address.street;
// Throws error as address is undef.

user.address?.street;
// Return undefined
```

4 ARRAY AT

This is similar to the square bracket notation to access array items, with the benefit that you can use negative numbers to access items from the end.

```
const arr = [0, 1, 2, 3, 4, 5];

arr.at(1); // 1
arr.at(2); // 2
arr.at(-1); // 5
arr.at(-2); // 4
```

5 DYNAMIC PROPS

Dynamic property names can be set by using the square bracket notation when declaring an Object.

```
const propName = 'foo';

const myObj = {
  bar: 123,
  [propName]: 456,
};

myObj.foo; // 456
```

FOLLOW