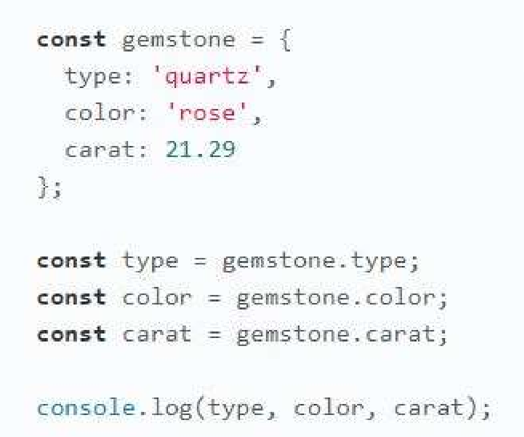
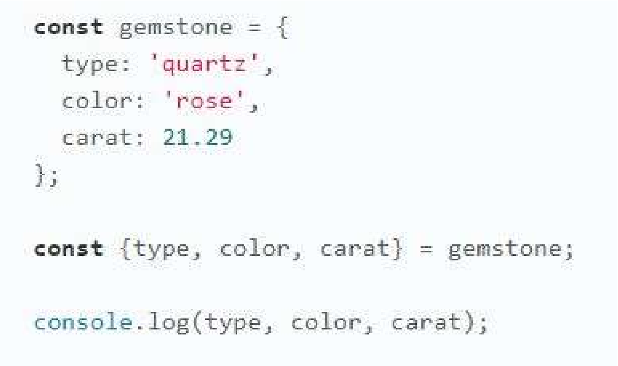
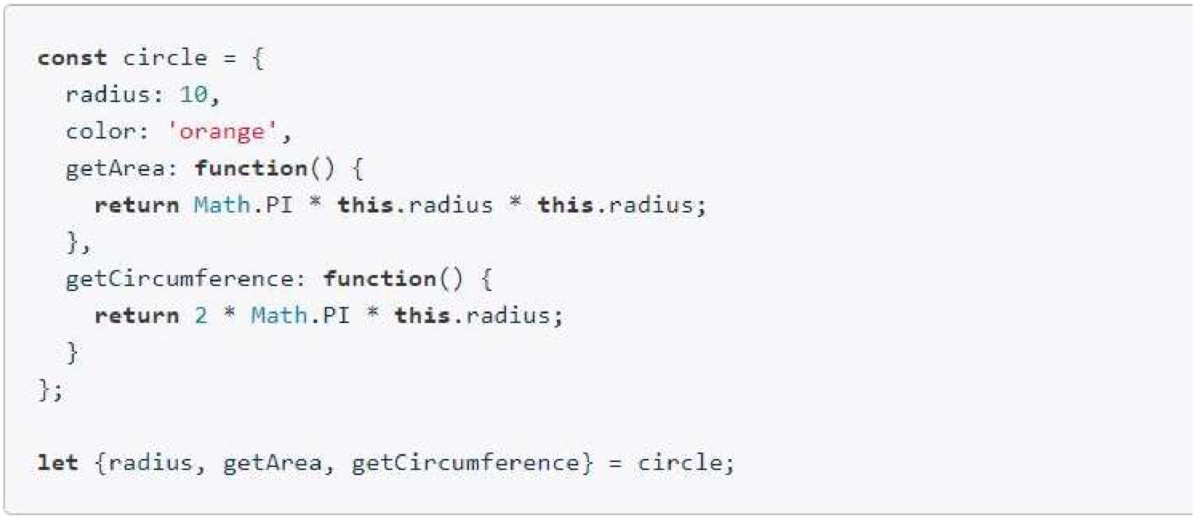


```
const gemstone = {
  type: 'quartz',
  color: 'rose',
  carat: 21.29
};

const type = gemstone.type;
const color = gemstone.color;
const carat = gemstone.carat;

console.log(type, color, carat);
```

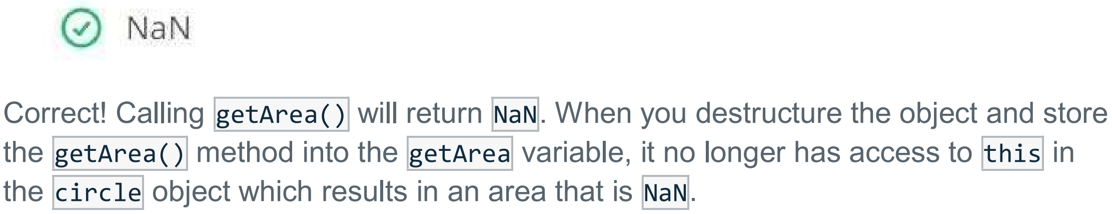


```
const gemstone = {
  type: 'quartz',
  color: 'rose',
  carat: 21.29
};

const {type, color, carat} = gemstone;

console.log(type, color, carat);
```

<For example>



```
const circle = {
  radius: 10,
  color: 'orange',
  getArea: function() {
    return Math.PI * this.radius * this.radius;
  },
  getCircumference: function() {
    return 2 * Math.PI * this.radius;
  }
};

let {radius, getArea, getCircumference} = circle;
```



✓ NaN

Correct! Calling `getArea()` will return `NaN`. When you destructure the object and store the `getArea()` method into the `getArea` variable, it no longer has access to `this` in the `circle` object which results in an area that is `NaN`.