JavaScript

String Methods

Aravind G

String Methods

①

1. at()
2. charAt()
3. charCodeAt()
4. codePointAt()
5. concat()
6. endsWith()

②

7. includes()
8. indexOf()
9. isWellFormed()
10. lastIndexOf()
11. localeCompare()
12. match()

③

13. matchAll()
14. normalize()
15. padEnd()
16. padStart()
17. repeat()
18. replace()

④

19. replaceAll()
20. search()
21. slice()
22. split()
23. startsWith()
24. substring()

String Methods

5

25.toLocaleLowerCase()

26.toLocaleUpperCase()

27.toLowerCase()

28.toString()

29.toUpperCase()

30.toWellFormed()

6

31.trim()

32.trimEnd()

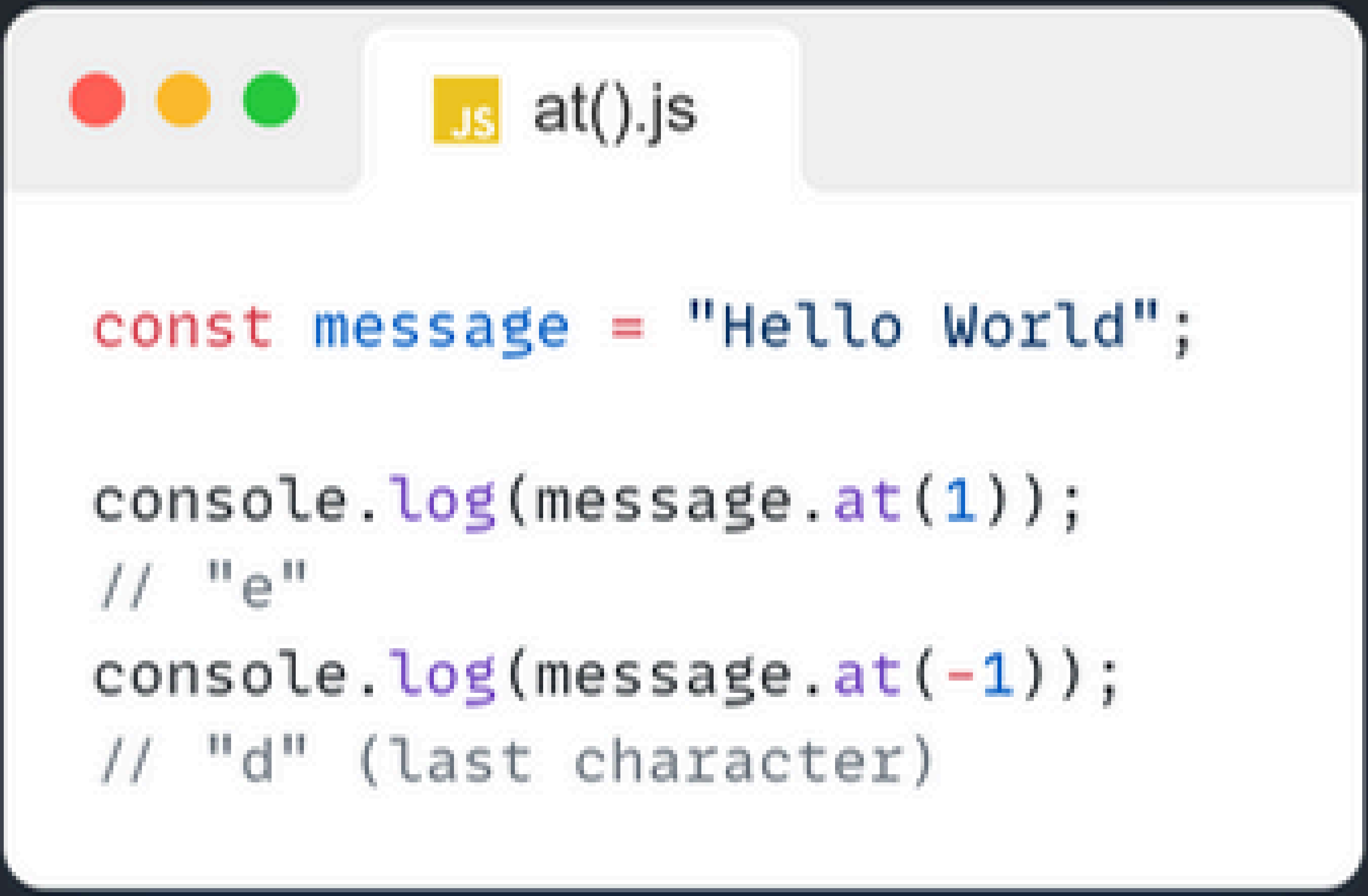
33.trimStart()

34.valueOf()

35.Symbol.iterator

1. at()

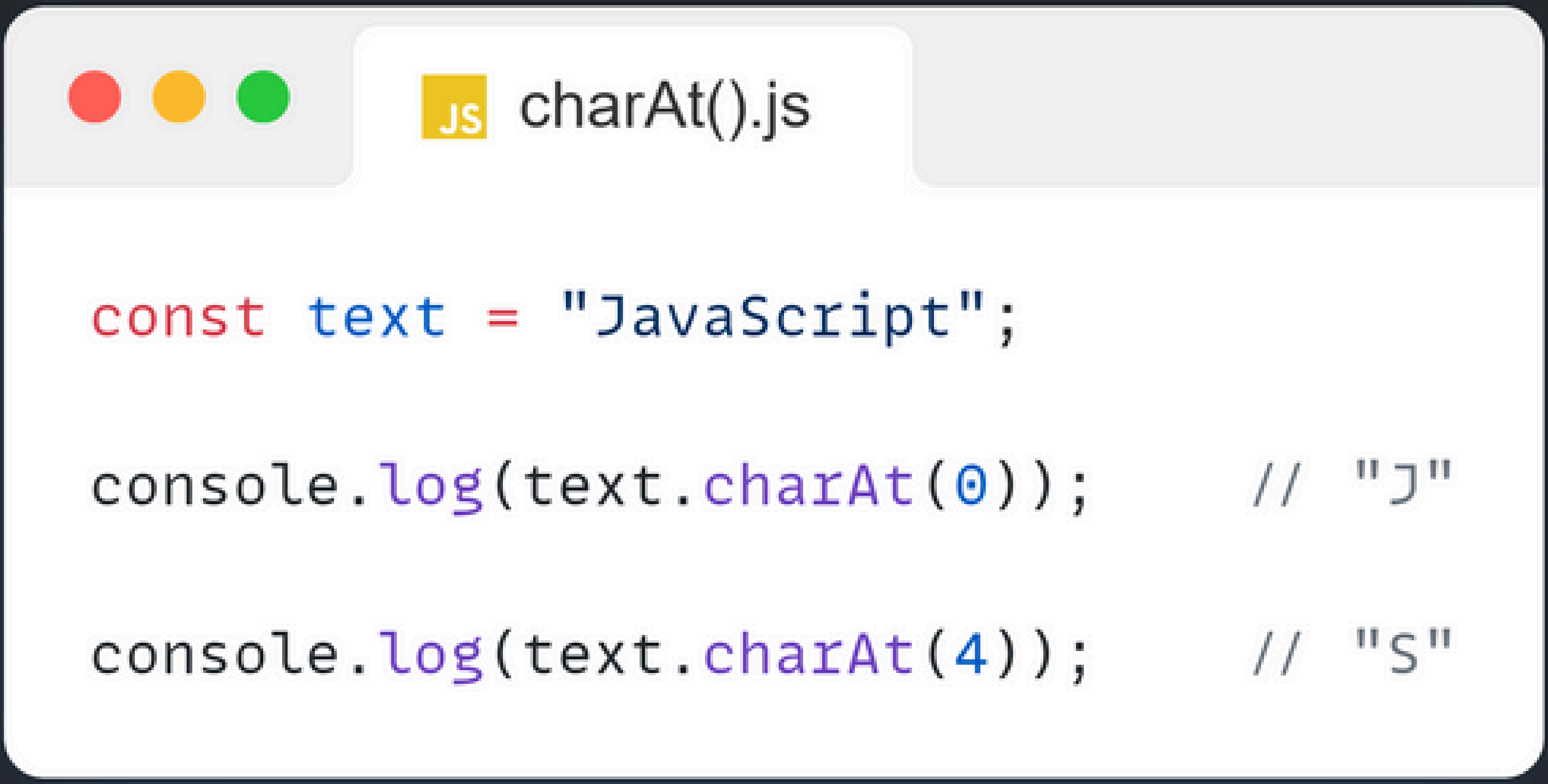
Returns the character at specified index, supports negative indices



```
const message = "Hello World";

console.log(message.at(1));
// "e"
console.log(message.at(-1));
// "d" (last character)
```

2.charAt()

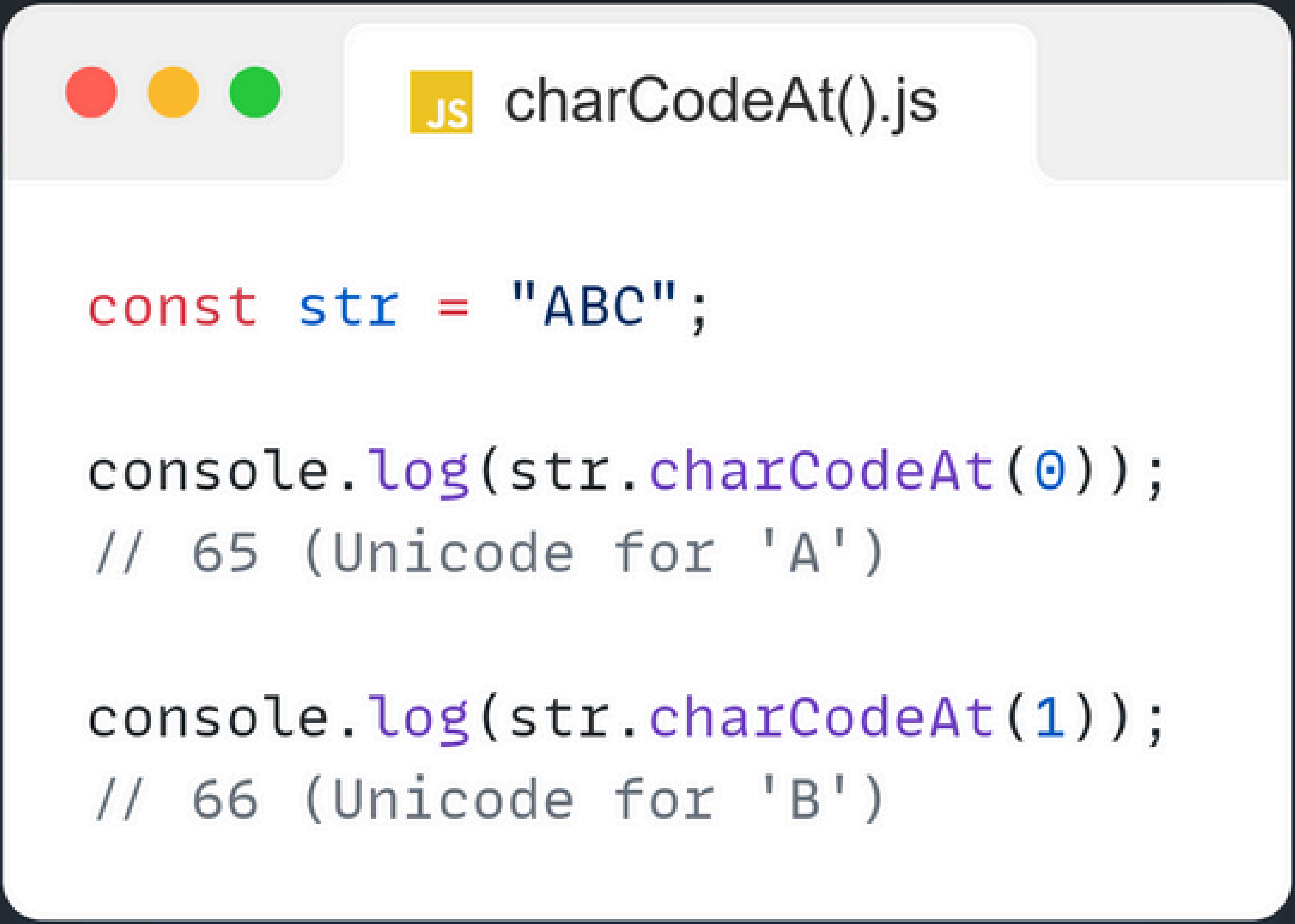
Returns the character at specified index , not negative indices



```
const text = "JavaScript";

console.log(text.charAt(0));     // "J"

console.log(text.charAt(4));     // "S"
```

3. charCodeAt()

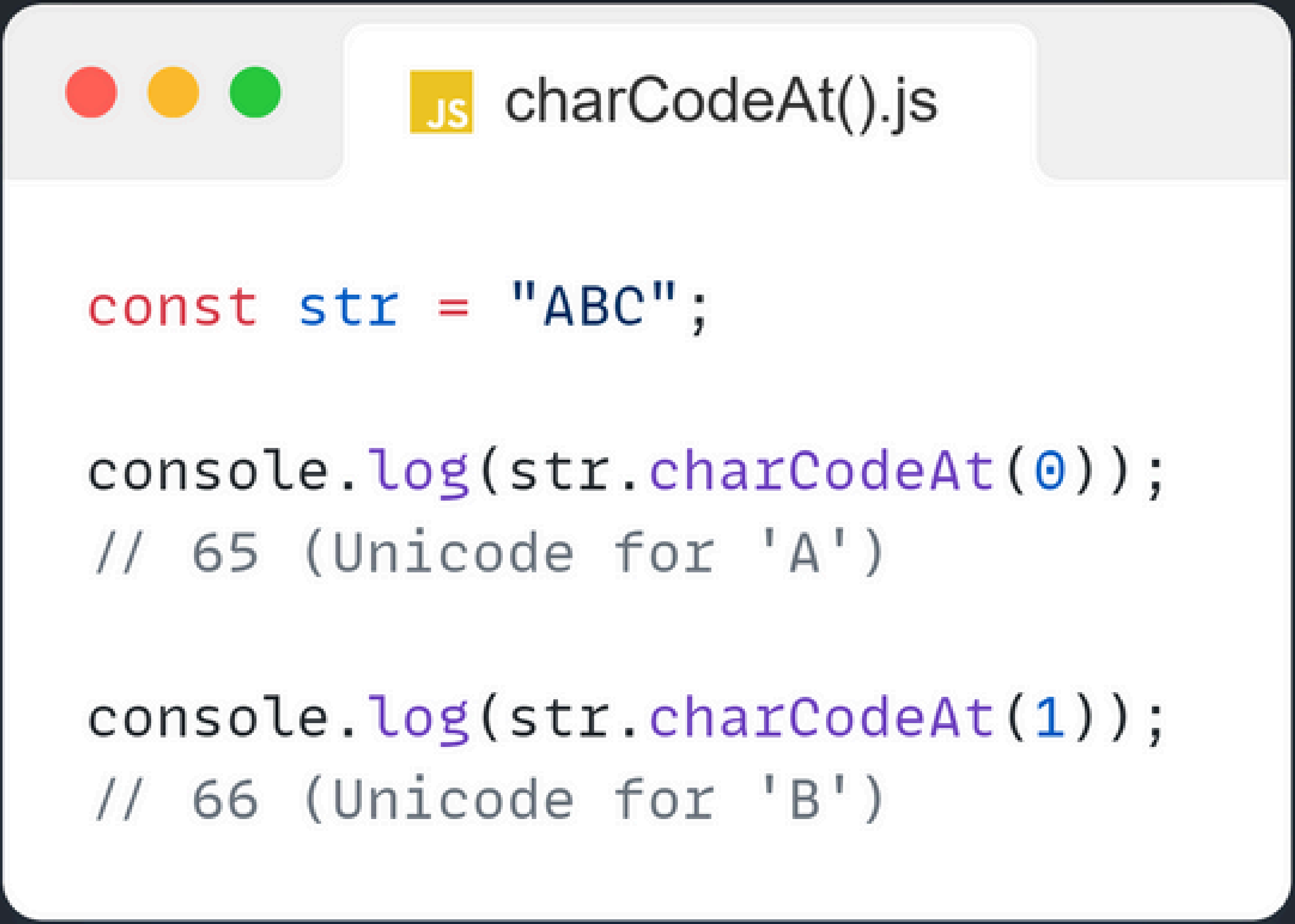
Returns the Unicode value of character at specified index



```
const str = "ABC";

console.log(str.charCodeAt(0));
// 65 (Unicode for 'A')

console.log(str.charCodeAt(1));
// 66 (Unicode for 'B')
```

4.codePointAt()

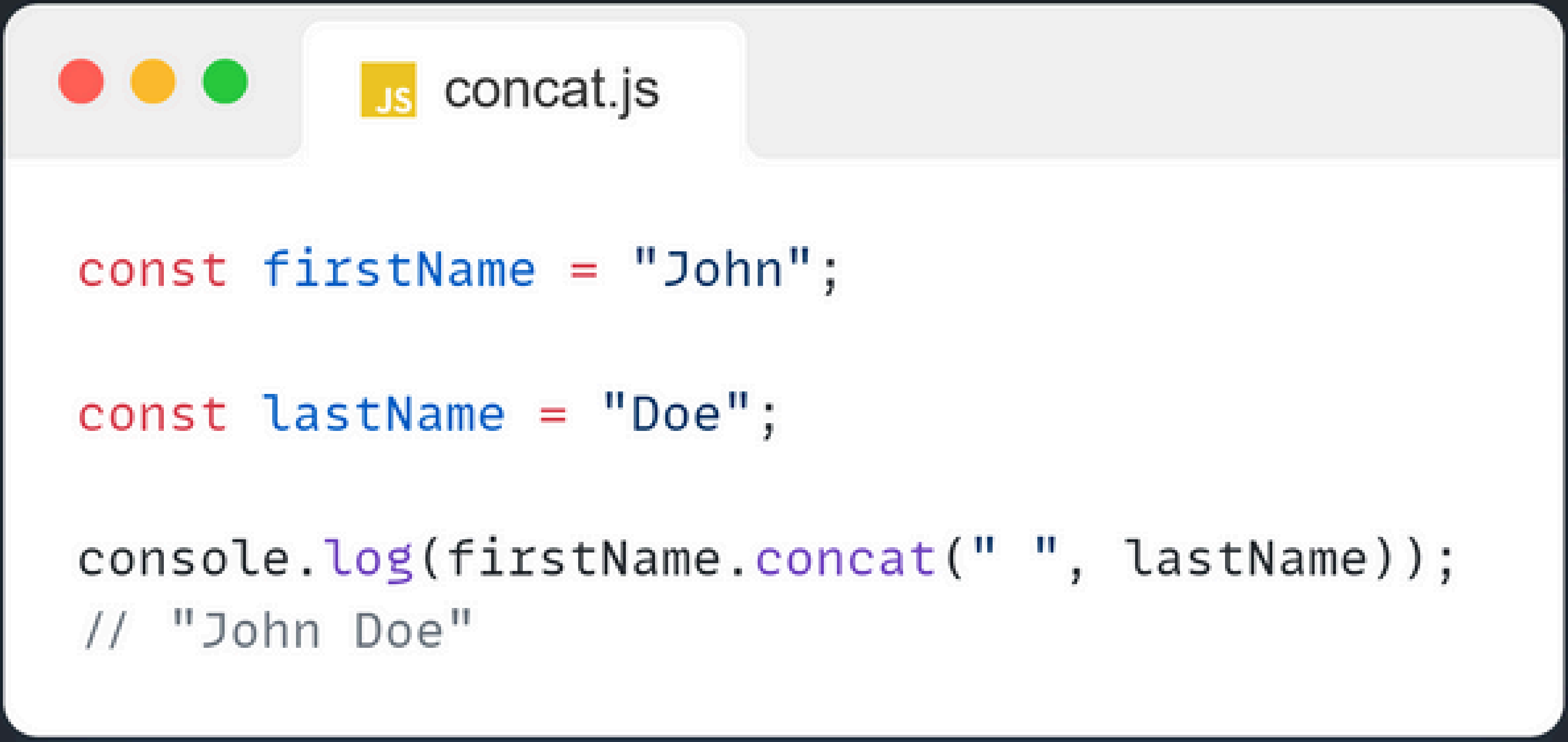
Returns the Unicode code point value



```
const str = "ABC";

console.log(str.charCodeAt(0));
// 65 (Unicode for 'A')

console.log(str.charCodeAt(1));
// 66 (Unicode for 'B')
```

5. concat()

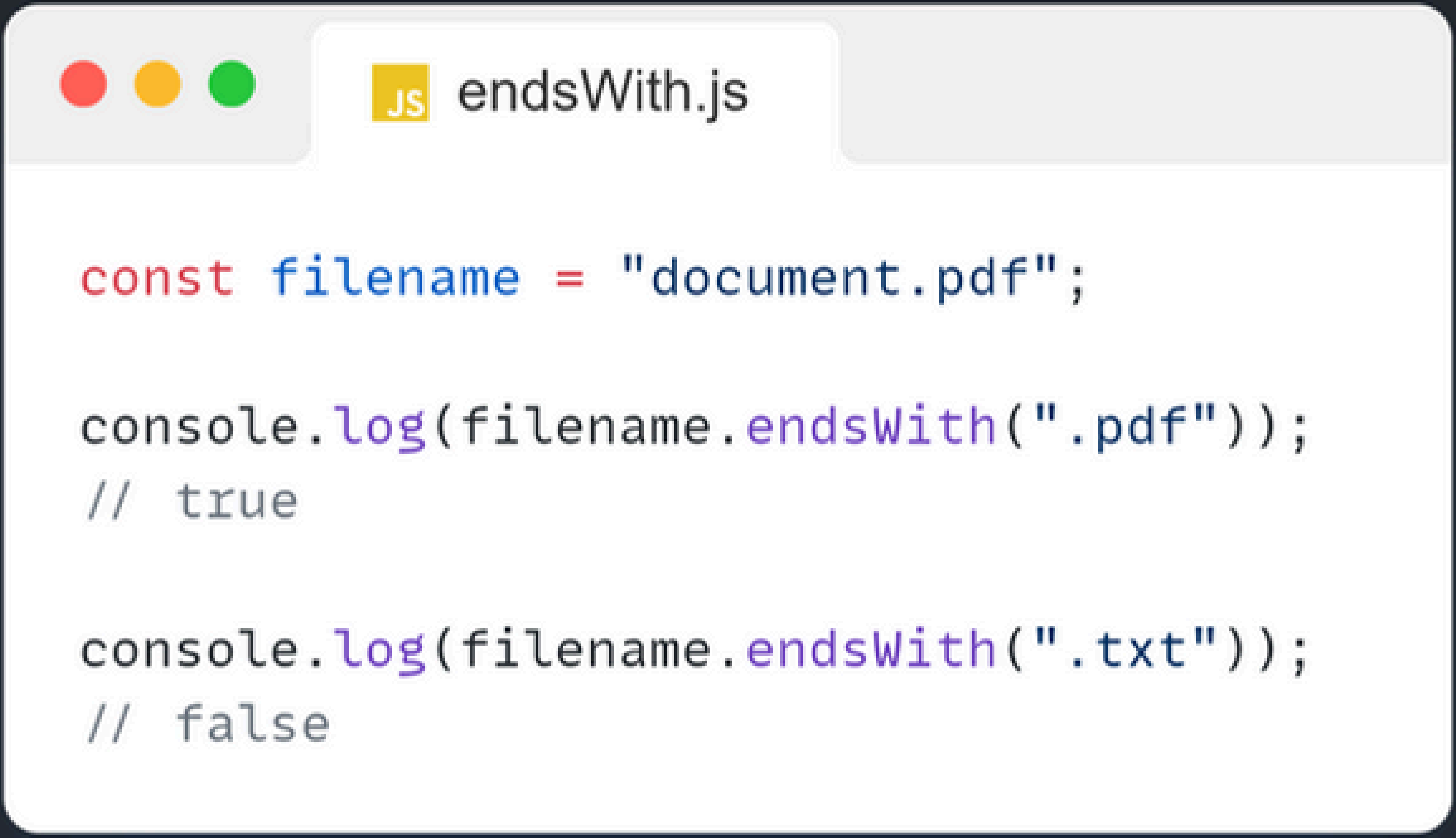
Joins two or more strings



```
const firstName = "John";

const lastName = "Doe";

console.log(firstName.concat(" ", lastName));
// "John Doe"
```

6. endsWith()

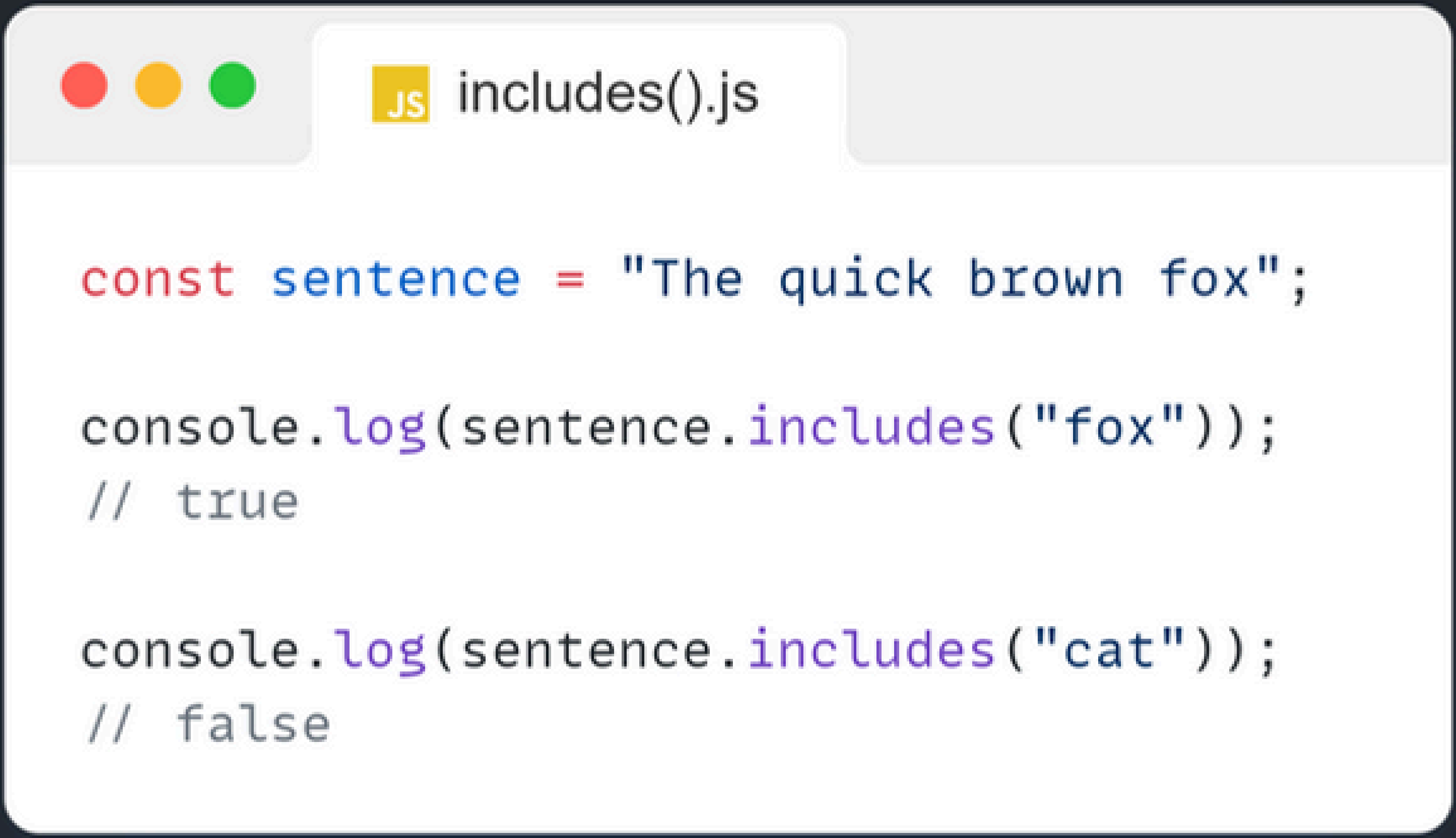
Checks if a string ends with specified characters



```
const filename = "document.pdf";

console.log(filename.endsWith(".pdf"));
// true

console.log(filename.endsWith(".txt"));
// false
```

7. includes()

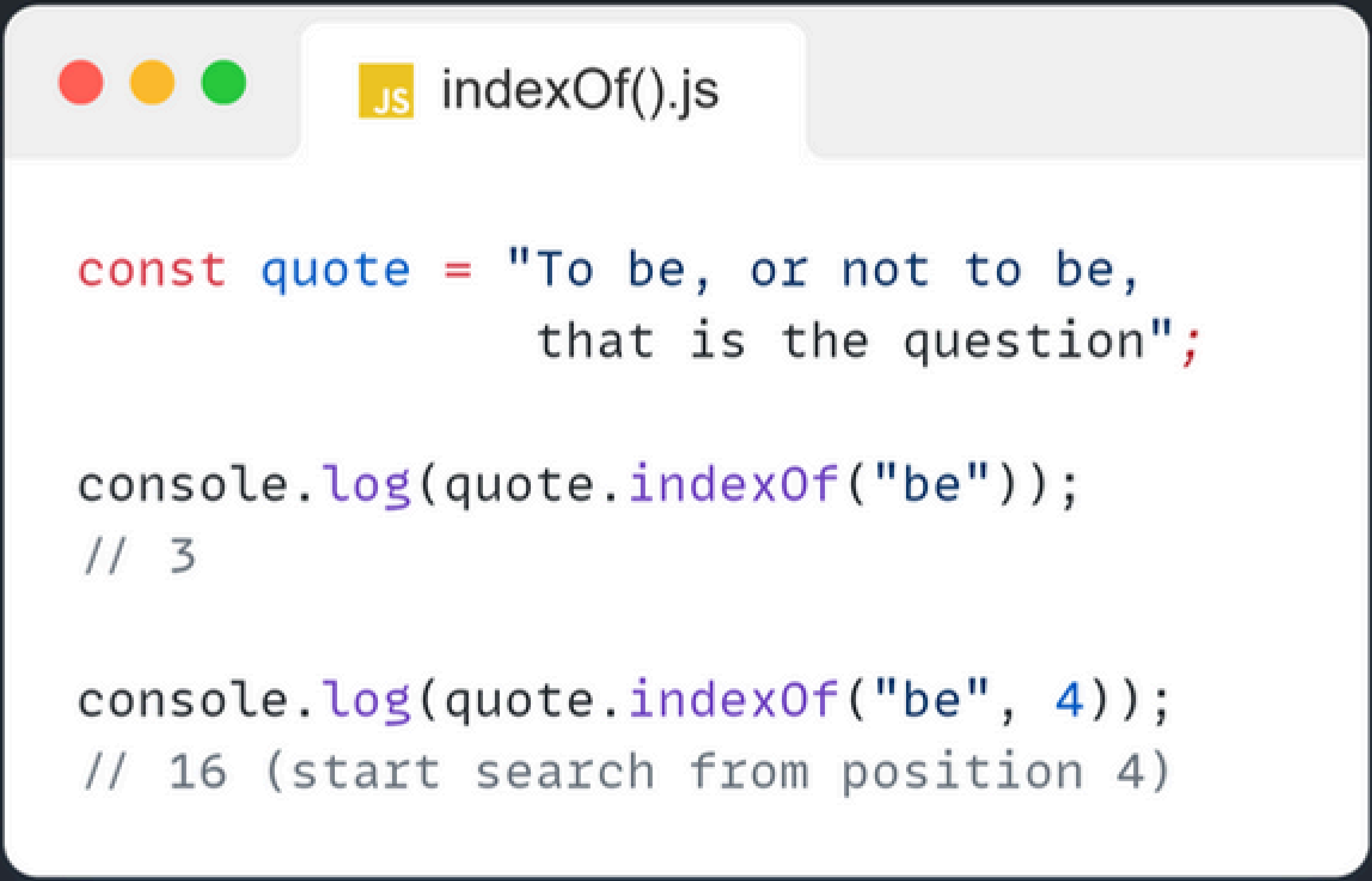
Checks if string contains a substring



```
const sentence = "The quick brown fox";

console.log(sentence.includes("fox"));
// true

console.log(sentence.includes("cat"));
// false
```

8. indexOf()

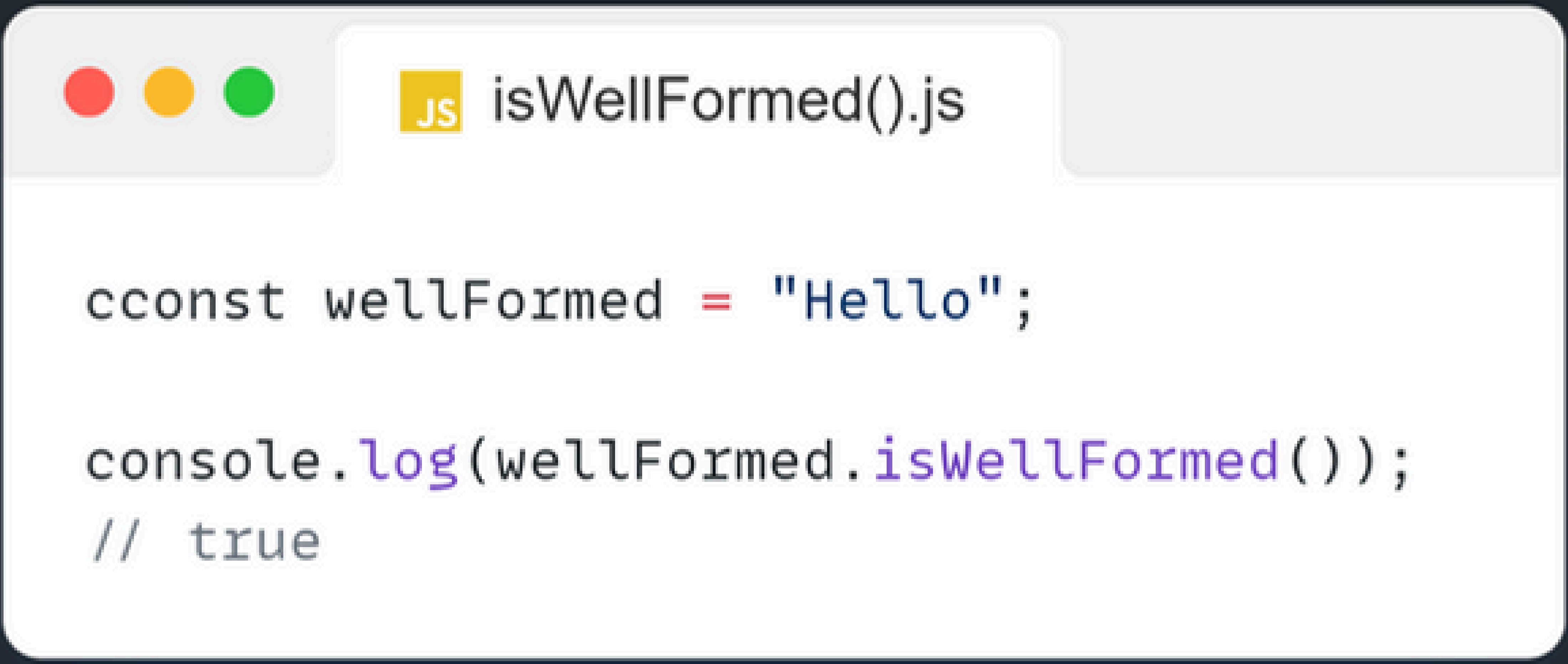
Returns the position of **first occurrence** of specified value



```
const quote = "To be, or not to be,
               that is the question";

console.log(quote.indexOf("be"));
// 3

console.log(quote.indexOf("be", 4));
// 16 (start search from position 4)
```

9. isWellFormed()

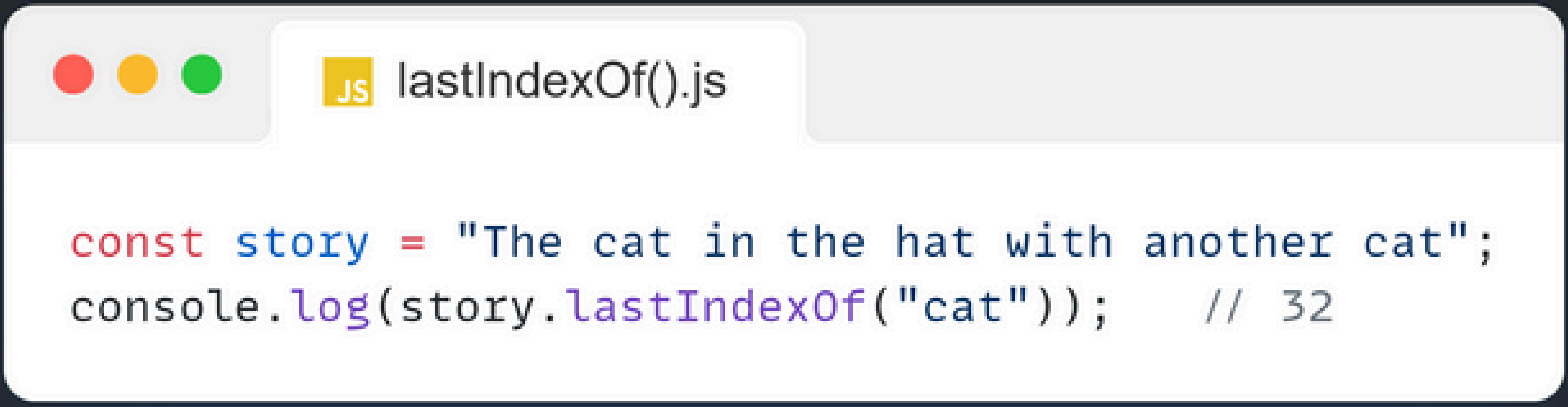
Checks if string contains valid UTF-16 code units



```
cconst wellFormed = "Hello";

console.log(wellFormed.isWellFormed());
// true
```

10. lastIndexOf()

Returns the last occurrence of specified value

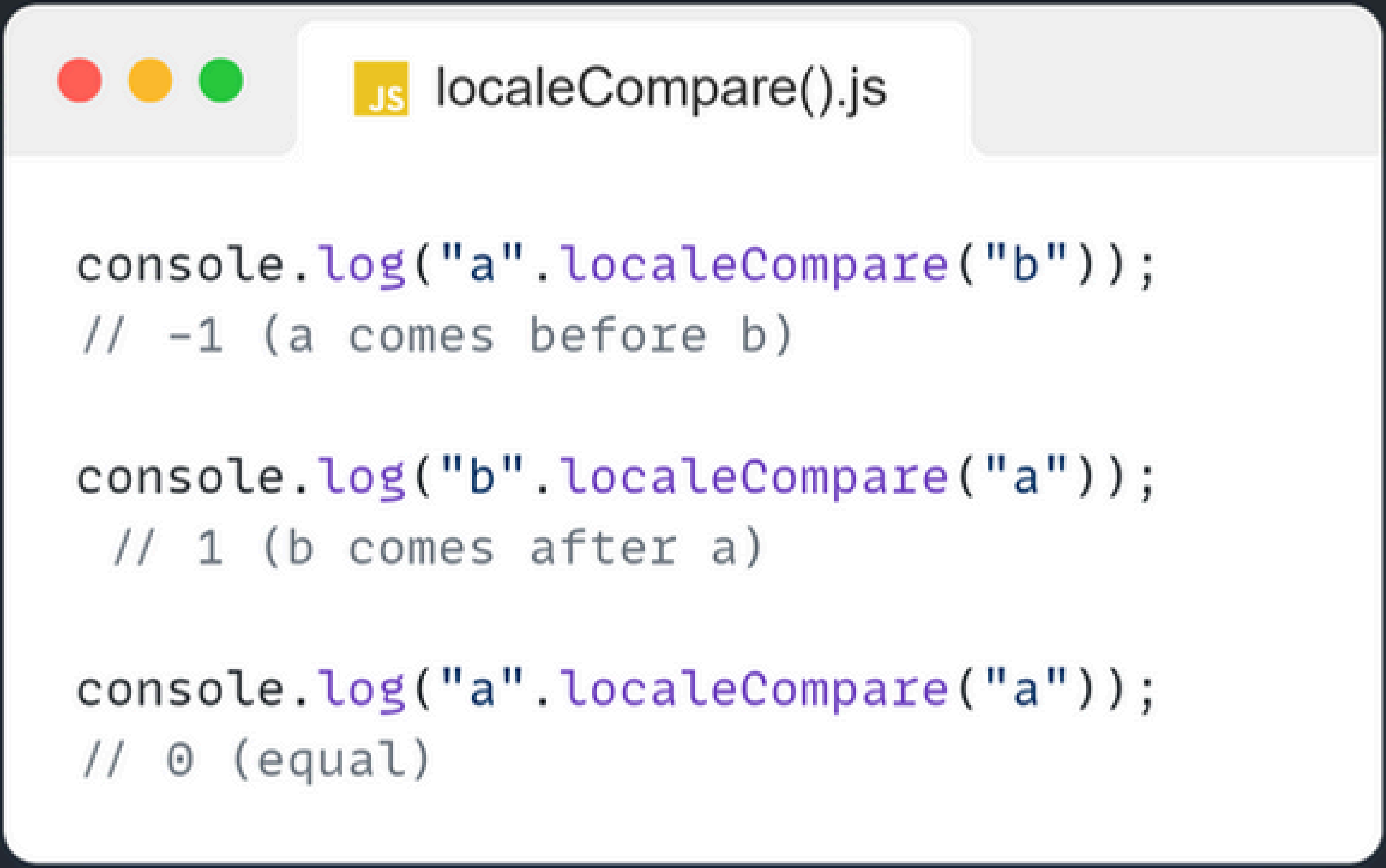


lastIndexOf().js

```
const story = "The cat in the hat with another cat";
console.log(story.lastIndexOf("cat"));    // 32
```

11. localeCompare()

Compares two strings in current locale

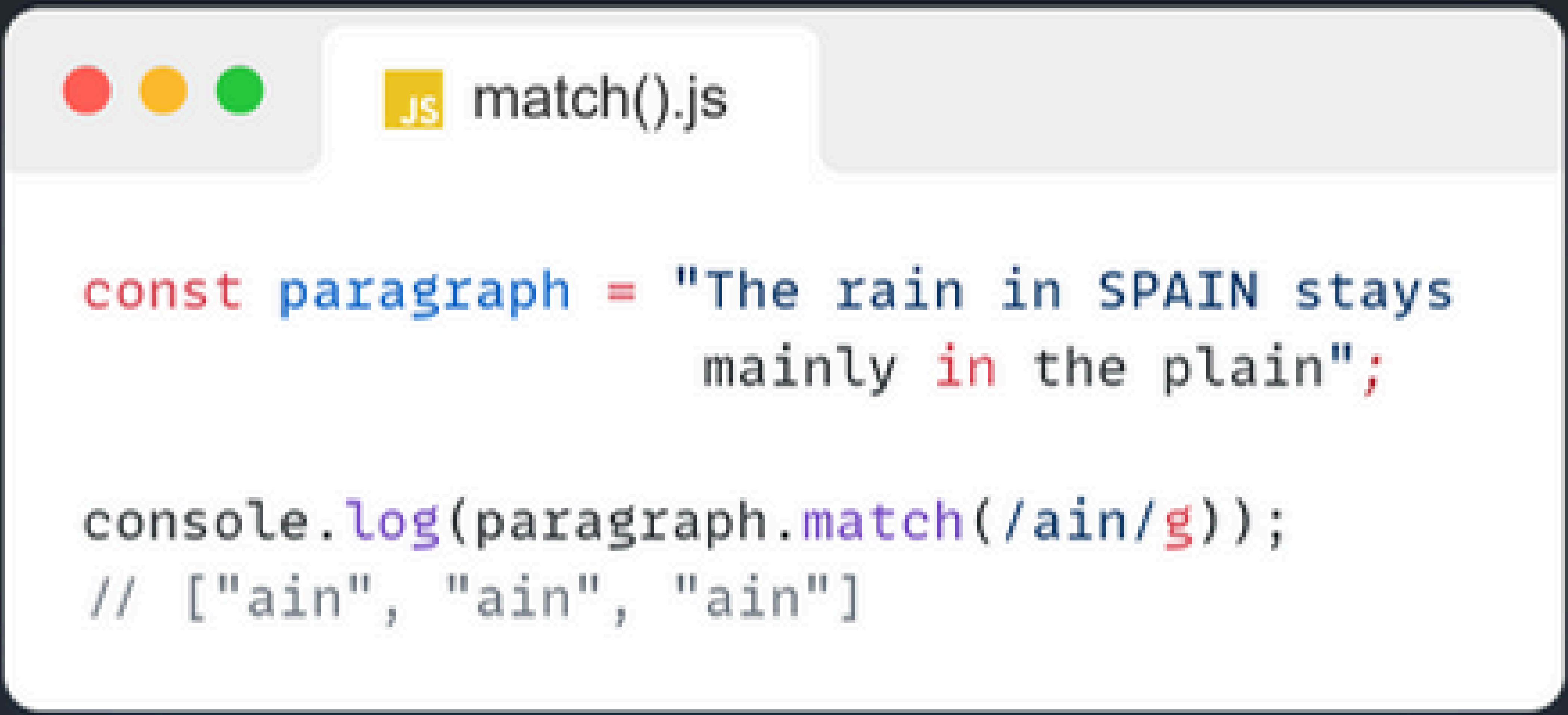


localeCompare().js

```
console.log("a".localeCompare("b"));
// -1 (a comes before b)

console.log("b".localeCompare("a"));
 // 1 (b comes after a)

console.log("a".localeCompare("a"));
// 0 (equal)
```

12. match()

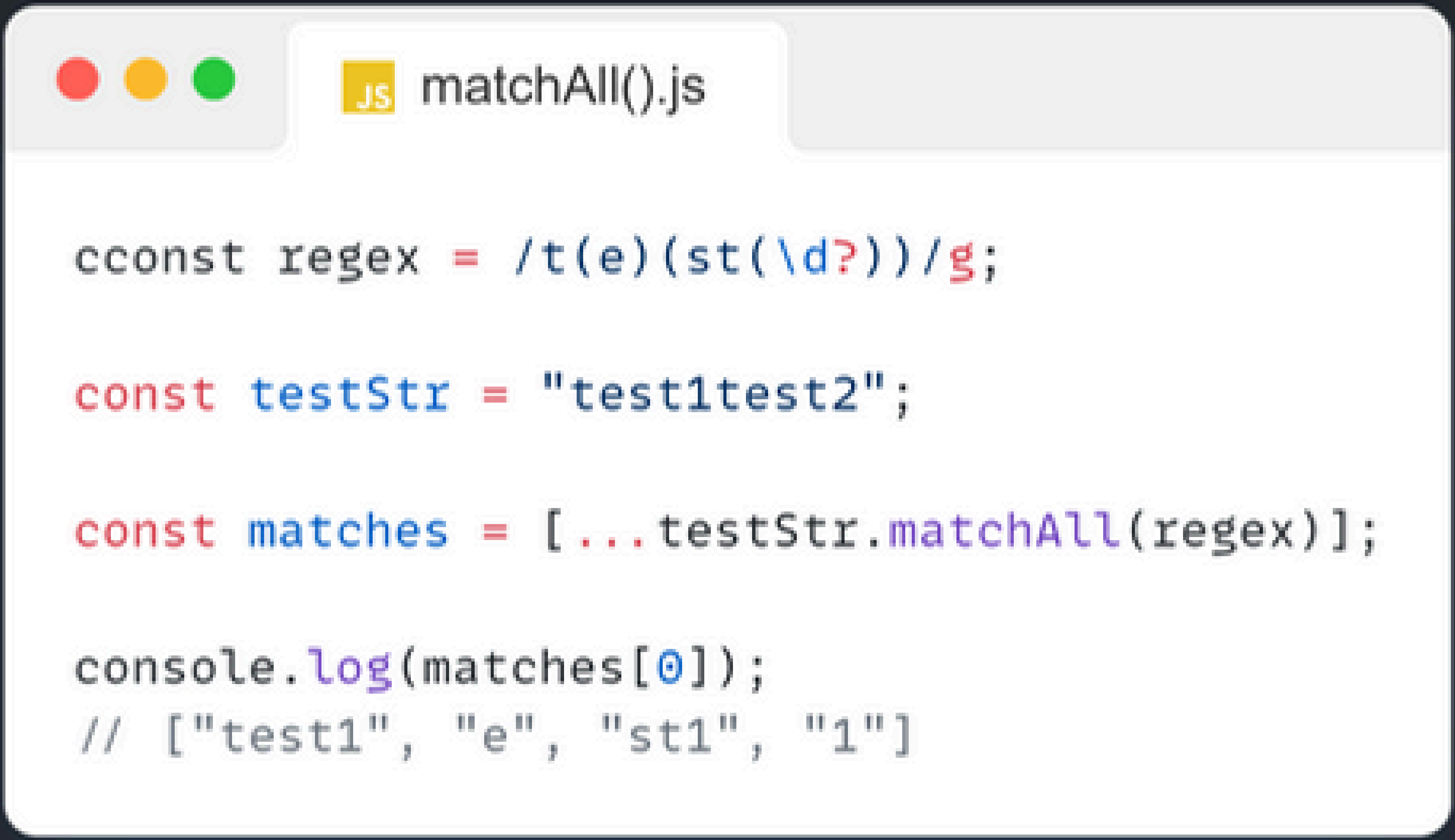
Searches for match against a regular expression



```
const paragraph = "The rain in SPAIN stays
                   mainly in the plain";

console.log(paragraph.match(/ain/g));
// ["ain", "ain", "ain"]
```

13. matchAll()

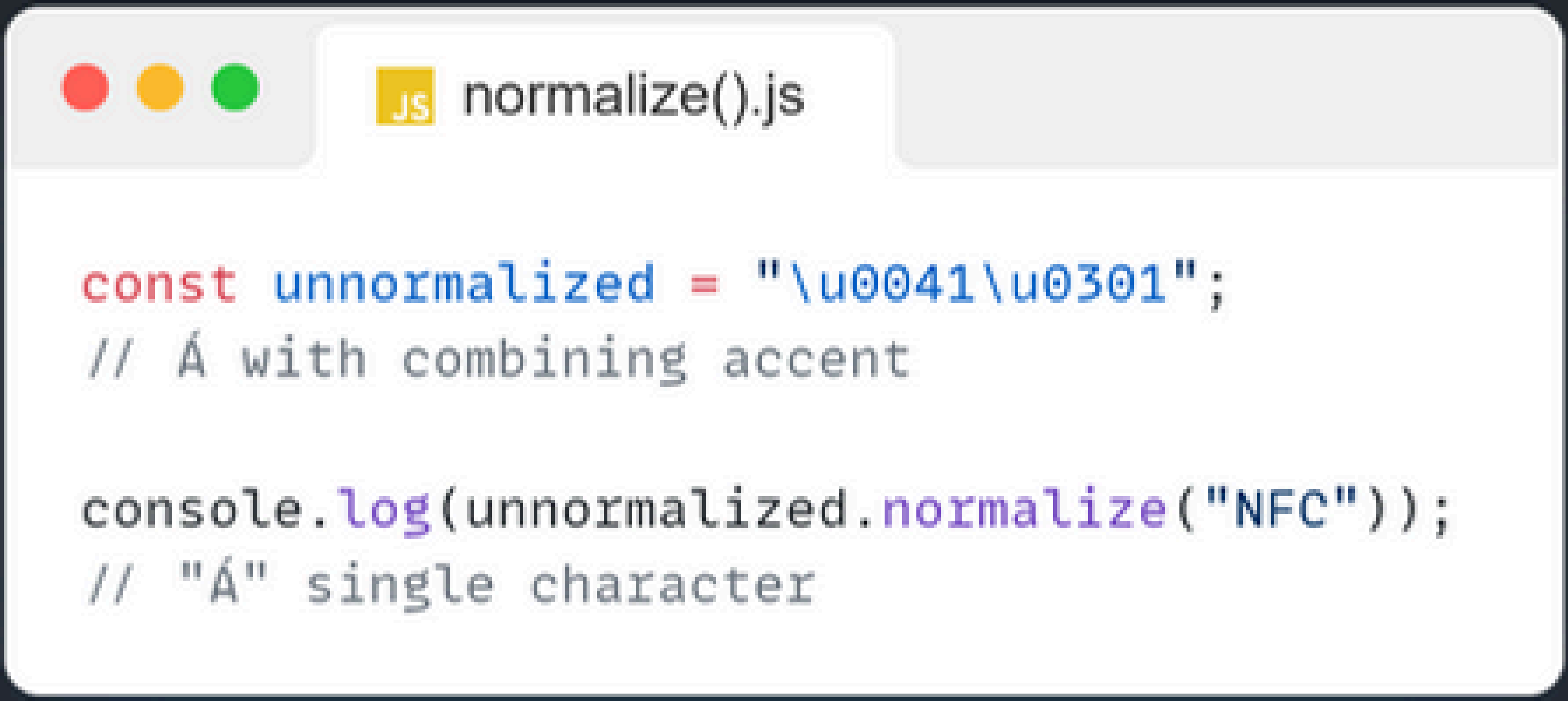
Returns iterator of all matches against a regular expression



```
cconst regex = /t(e)(st(\d?))/g;

const testStr = "test1test2";

const matches = [...testStr.matchAll(regex)];

console.log(matches[0]);
// ["test1", "e", "st1", "1"]
```

14. normalize()

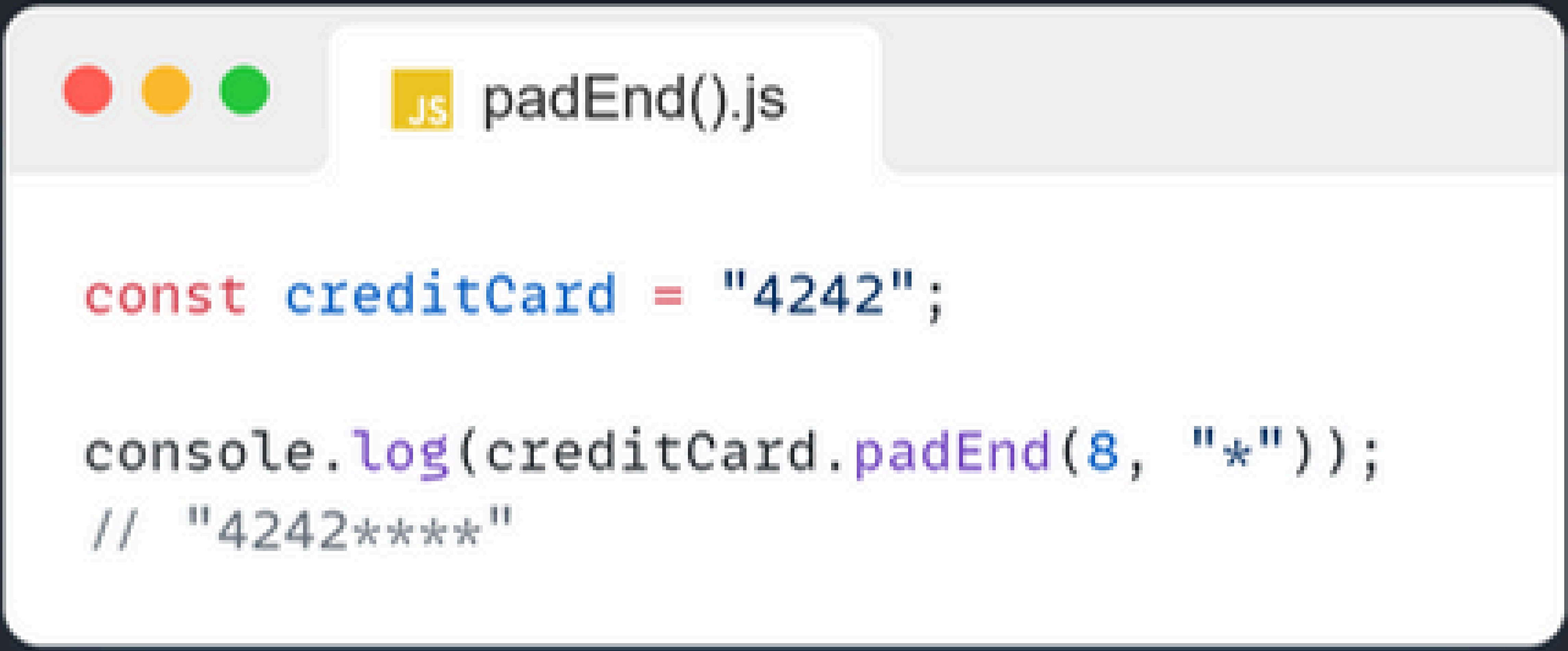
Returns Unicode Normalization Form



```
const unnormalized = "\u0041\u0301";
// Á with combining accent

console.log(unnormalized.normalize("NFC"));
// "Á" single character
```

15. padEnd()

Pads the string at the end

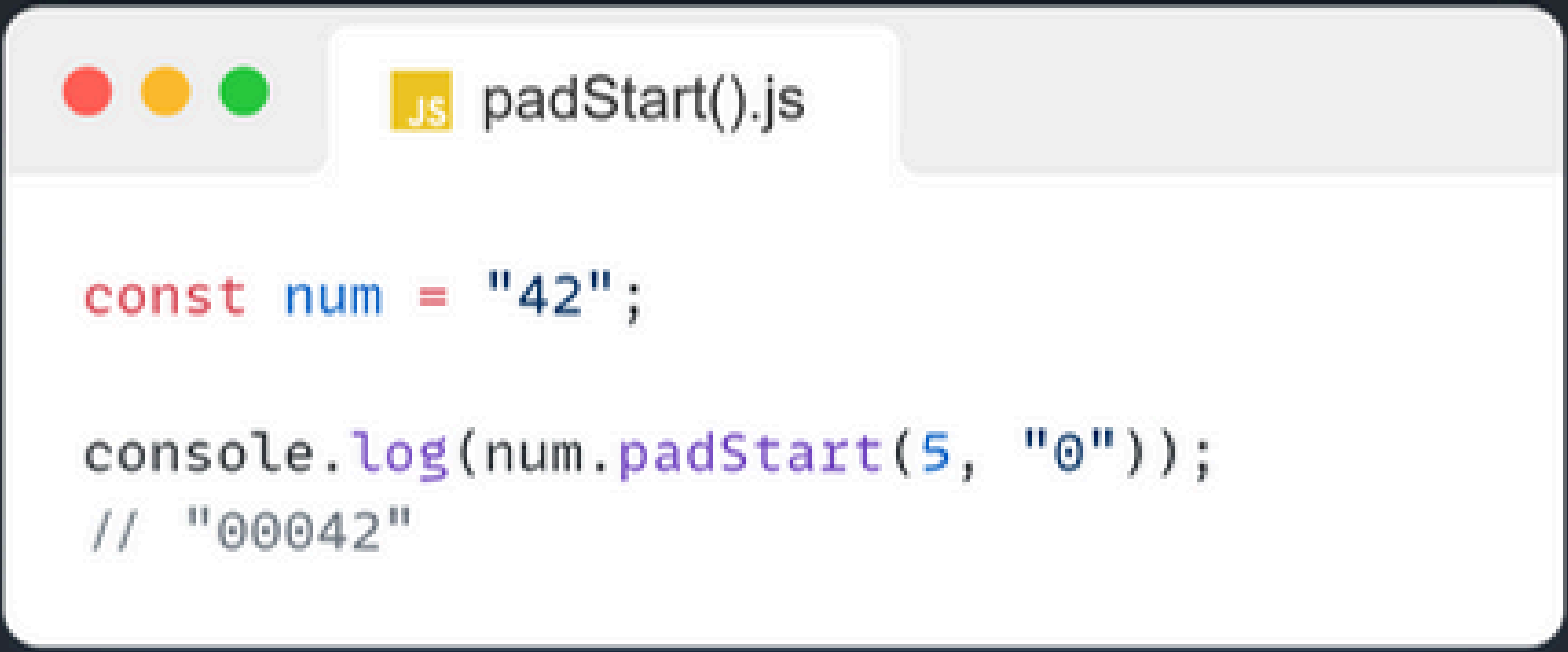


padEnd().js

```
const creditCard = "4242";

console.log(creditCard.padEnd(8, "*"));
// "4242****"
```

16. padStart()

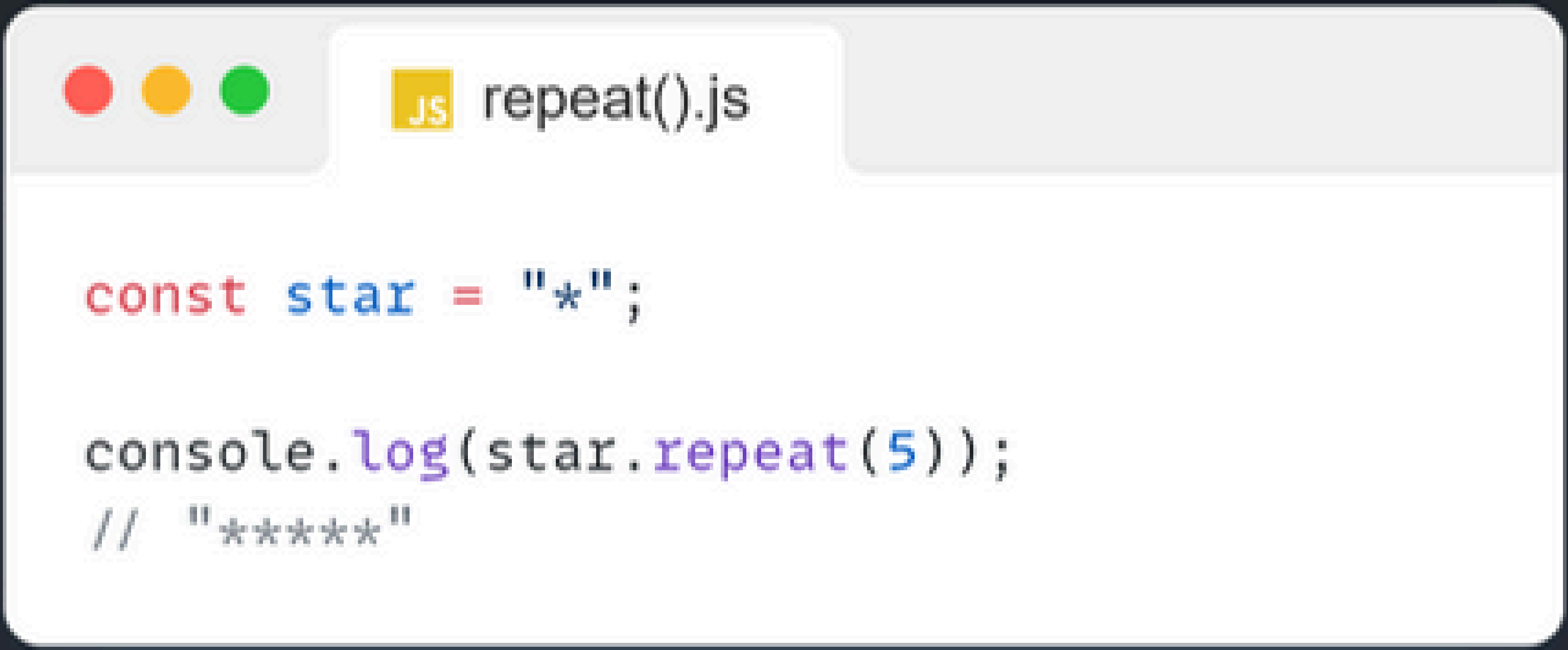
Pads the string at the beginning



```
const num = "42";

console.log(num.padStart(5, "0"));
// "00042"
```

17. repeat()

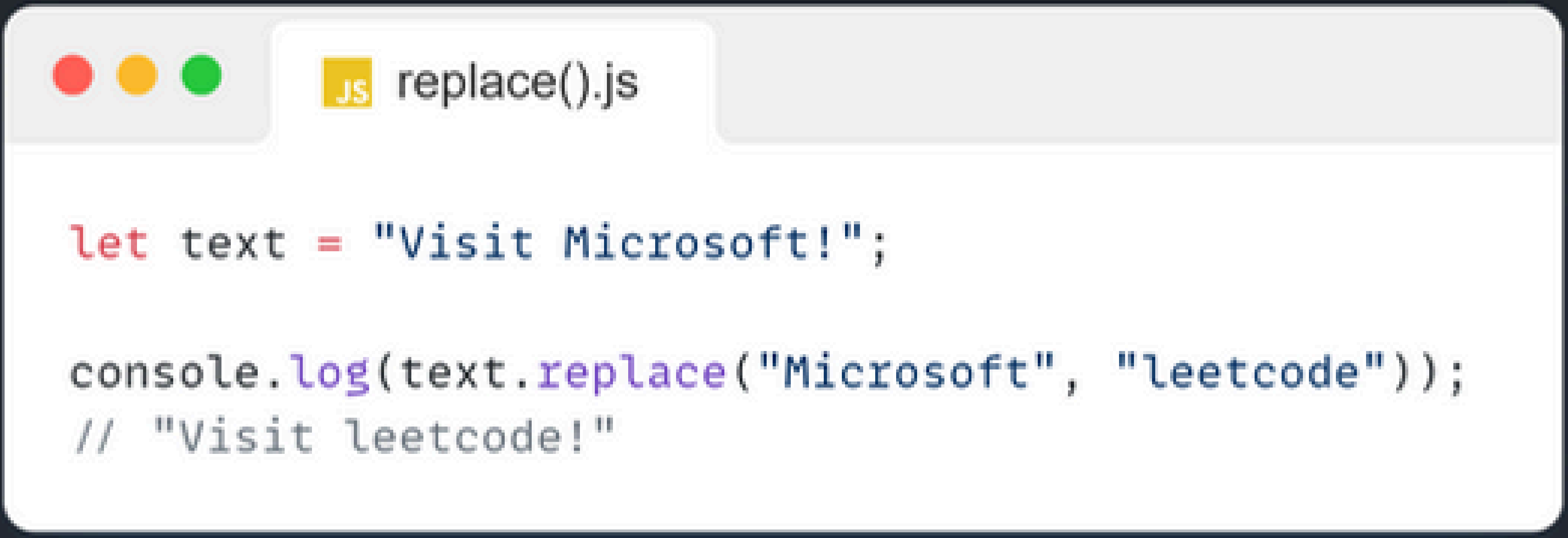
Returns a string repeated n times



```
const star = "*";

console.log(star.repeat(5));
// "*****"
```

18. replace()

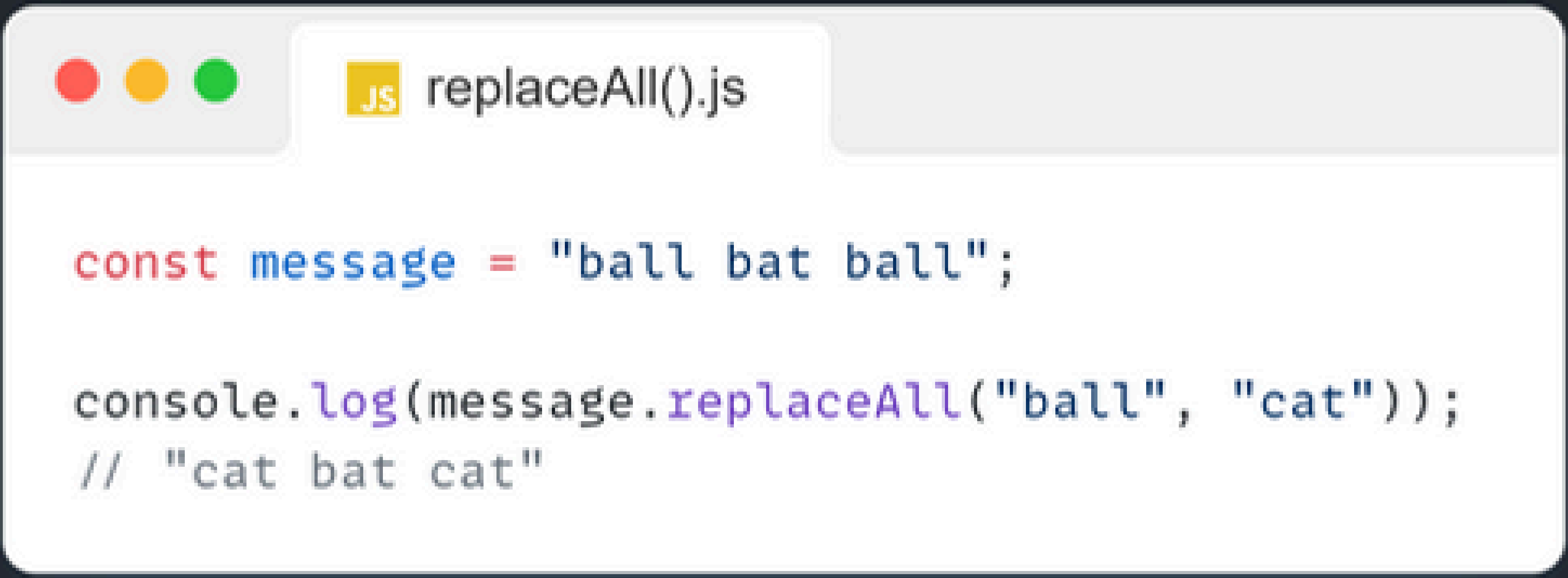
Replaces the first occurrence



```js
let text = "Visit Microsoft!";

console.log(text.replace("Microsoft", "leetcode"));
// "Visit leetcode!"
```

19. replaceAll()

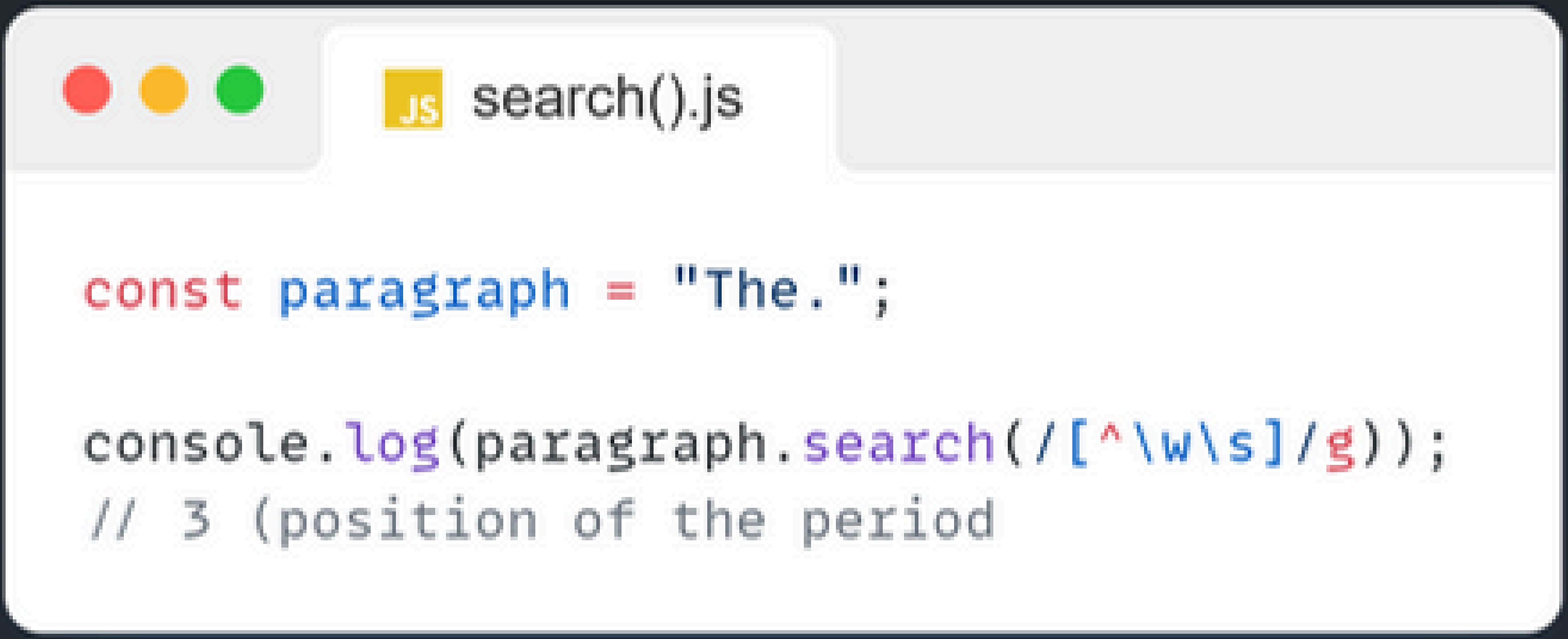
Replaces all occurrences



```javascript
const message = "ball bat ball";

console.log(message.replaceAll("ball", "cat"));
// "cat bat cat"
```

20. search()

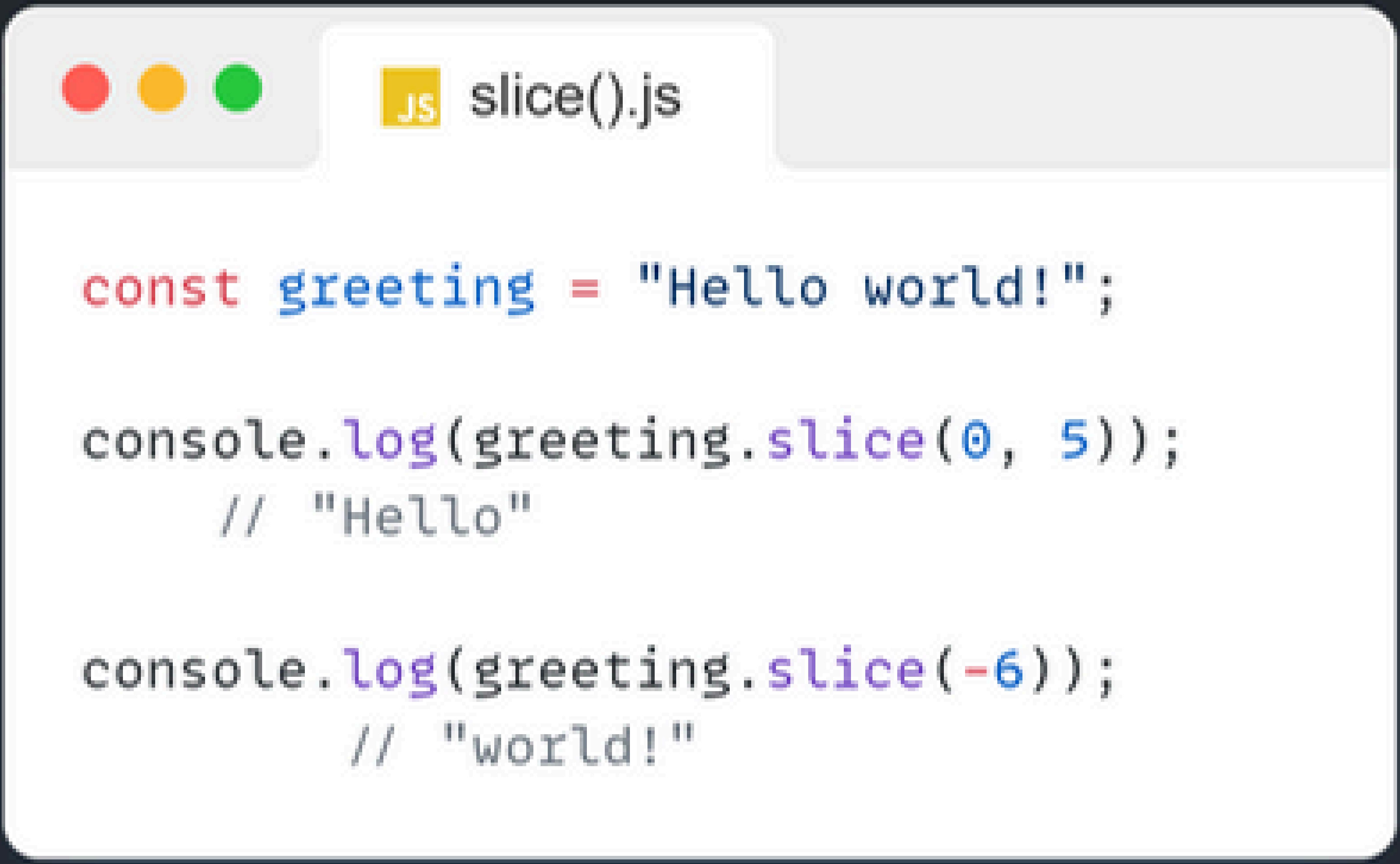
Searches for a match against a regex



```
const paragraph = "The.";

console.log(paragraph.search(/[^\w\s]/g));
// 3 (position of the period
```

21. slice()

Extracts a part of a string

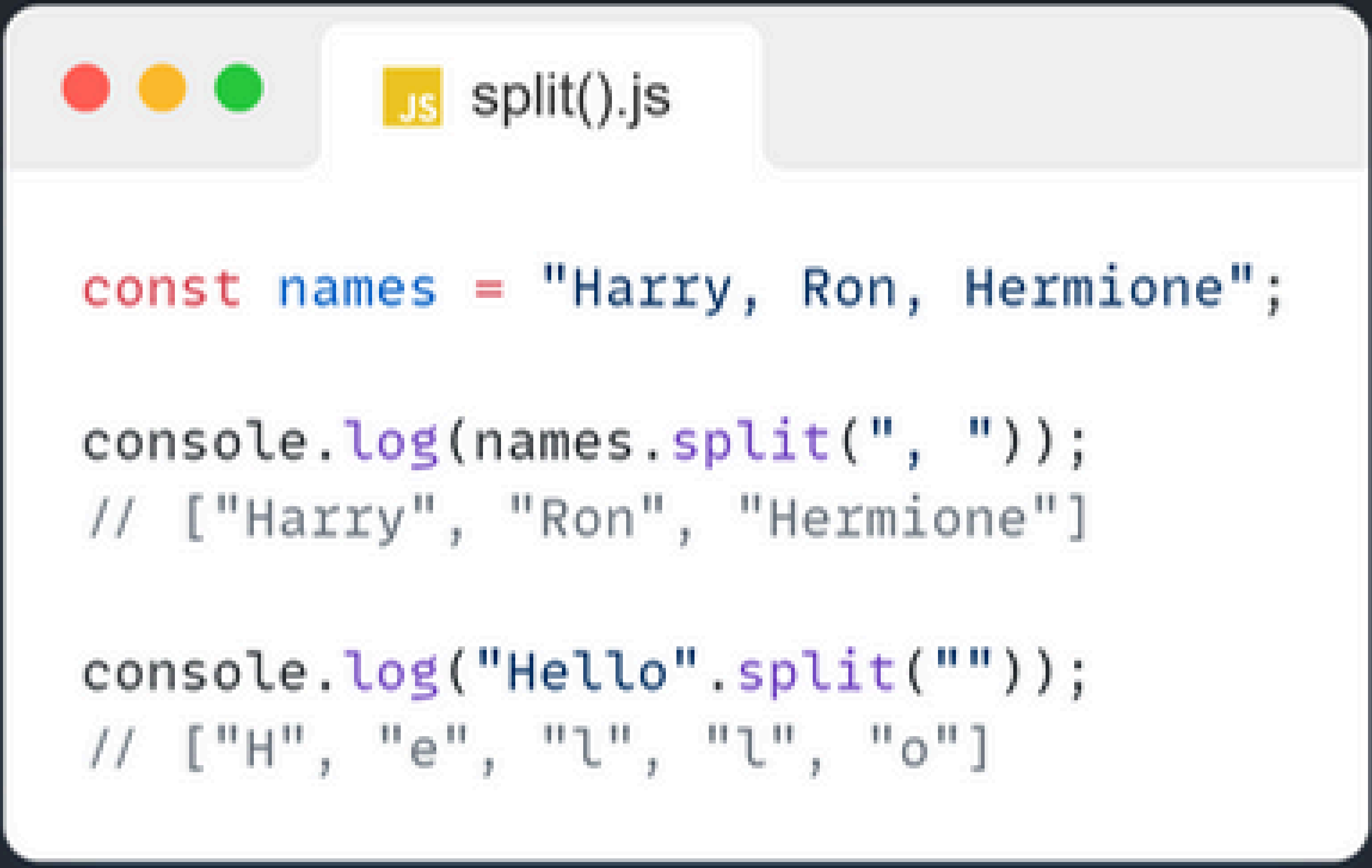


slice().js

```
const greeting = "Hello world!";

console.log(greeting.slice(0, 5));
    // "Hello"

console.log(greeting.slice(-6));
        // "world!"
```

22. split()

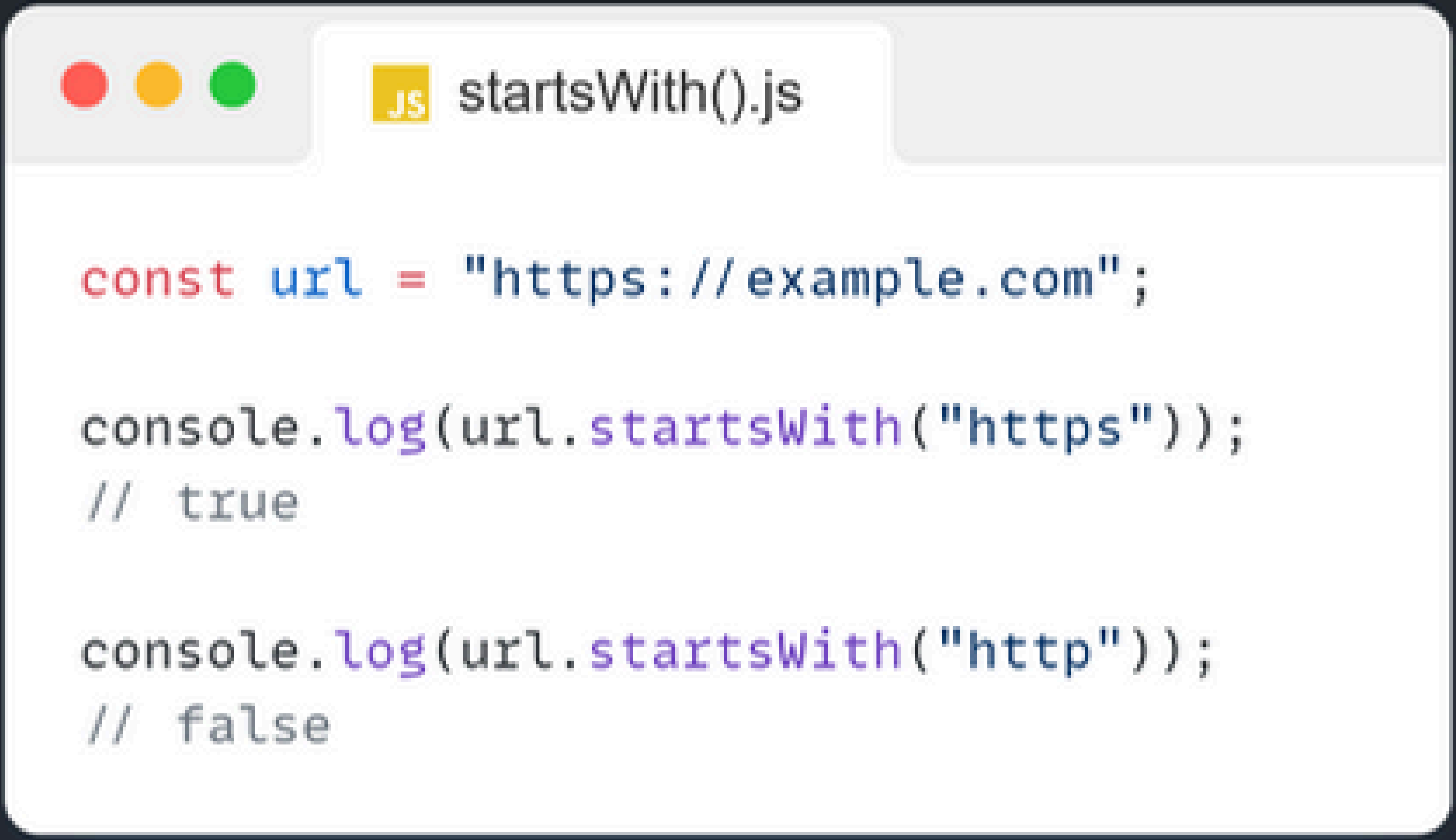
Splits a string into an array of substrings



```
const names = "Harry, Ron, Hermione";

console.log(names.split(", "));
// ["Harry", "Ron", "Hermione"]

console.log("Hello".split(""));
// ["H", "e", "l", "l", "o"]
```

23. startsWith()

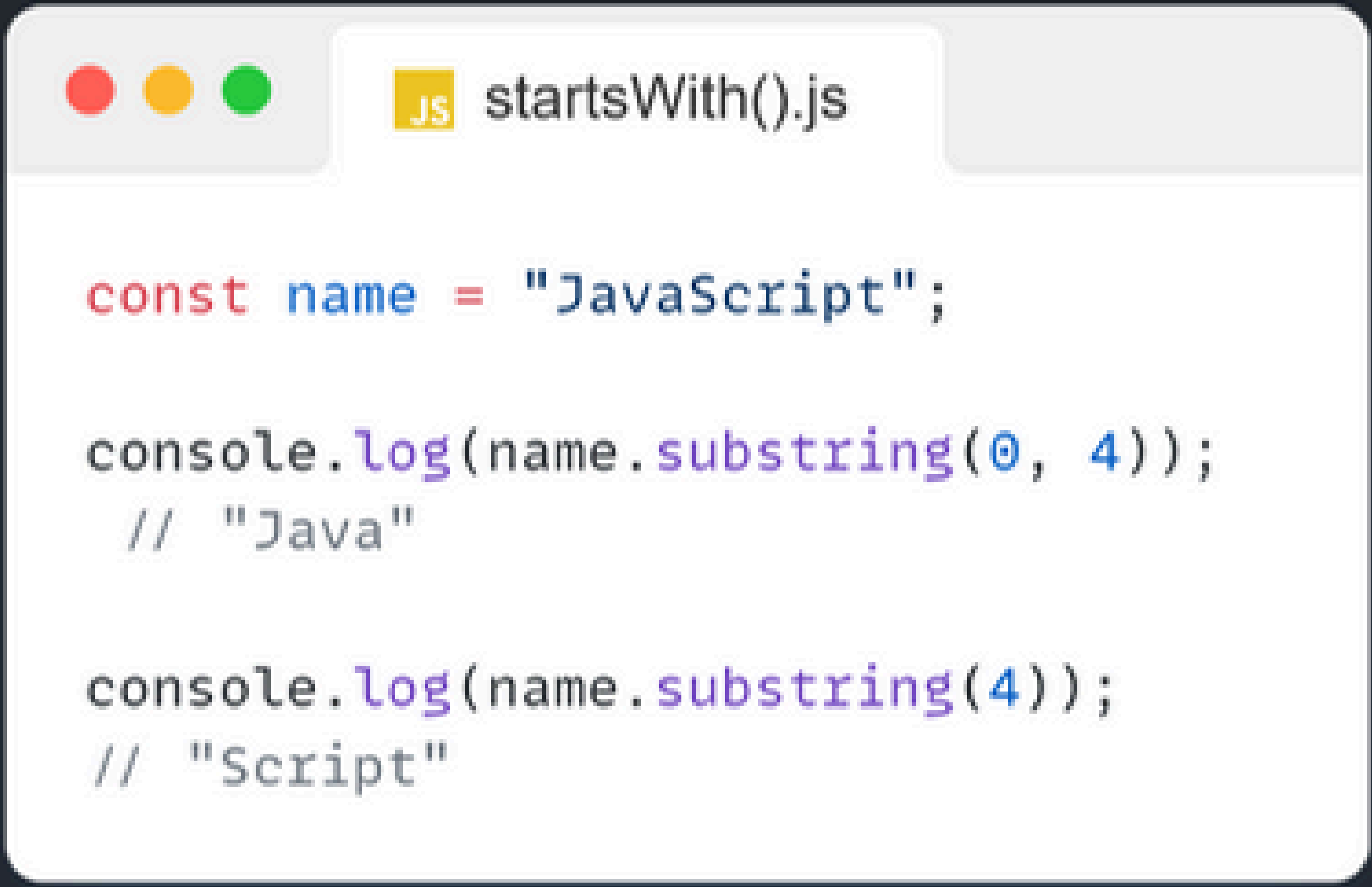
Checks if string starts with specified characters



```
const url = "https://example.com";

console.log(url.startsWith("https"));
// true

console.log(url.startsWith("http"));
// false
```

24. substring()

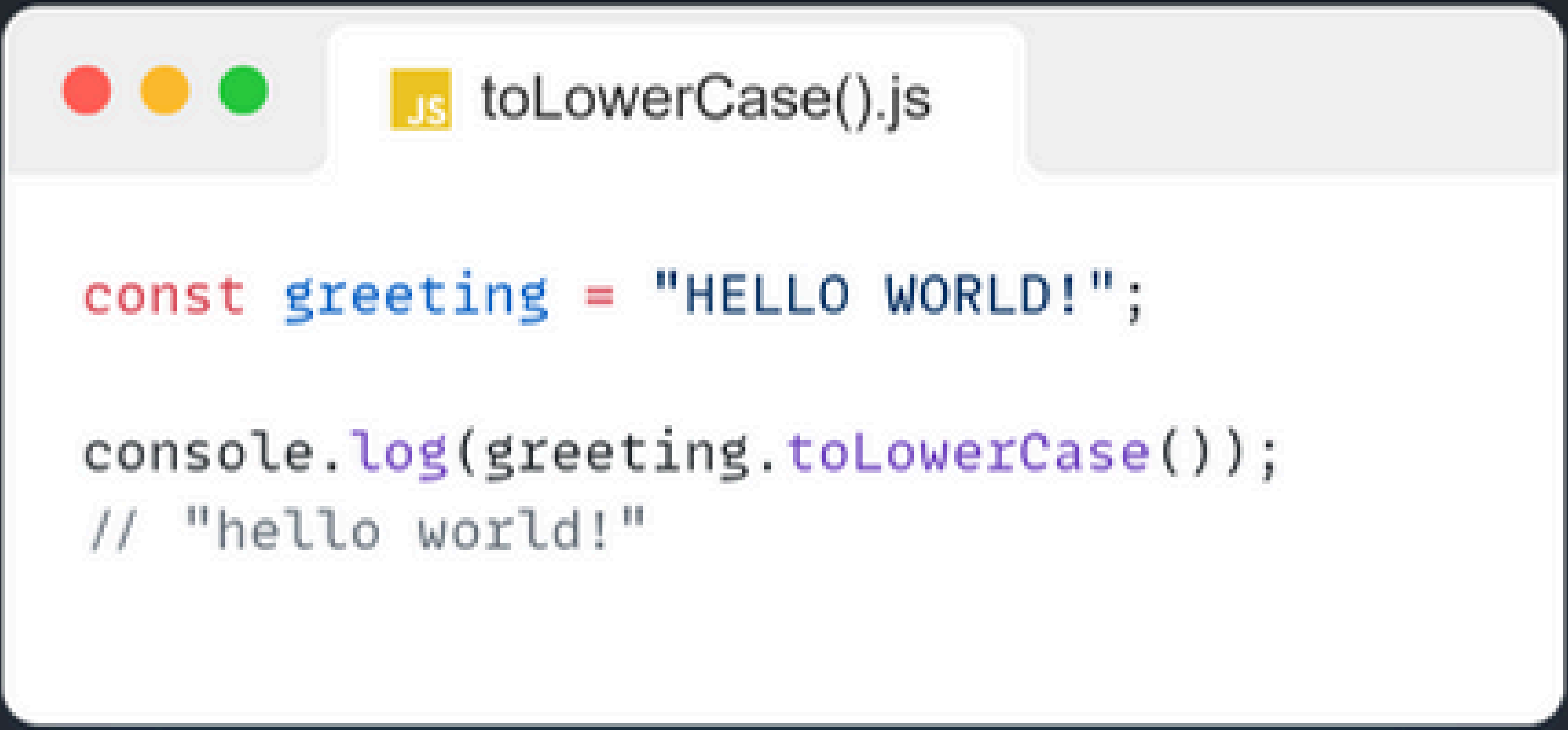
Extracts characters between two indices



```javascript
const name = "JavaScript";

console.log(name.substring(0, 4));
 // "Java"

console.log(name.substring(4));
// "Script"
```

25. toLowerCase()

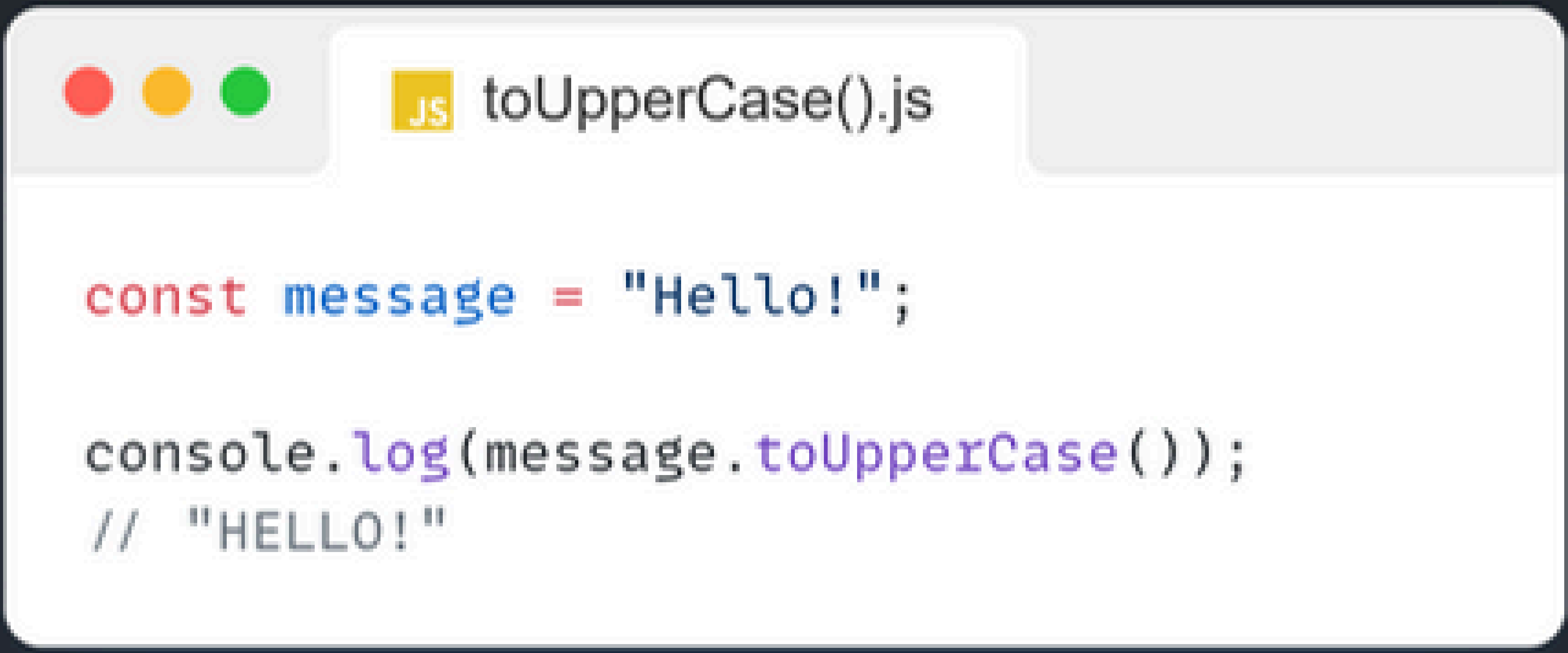
Converts a string to lowercase



```
const greeting = "HELLO WORLD!";

console.log(greeting.toLowerCase());
// "hello world!"
```

26. toUpperCase()

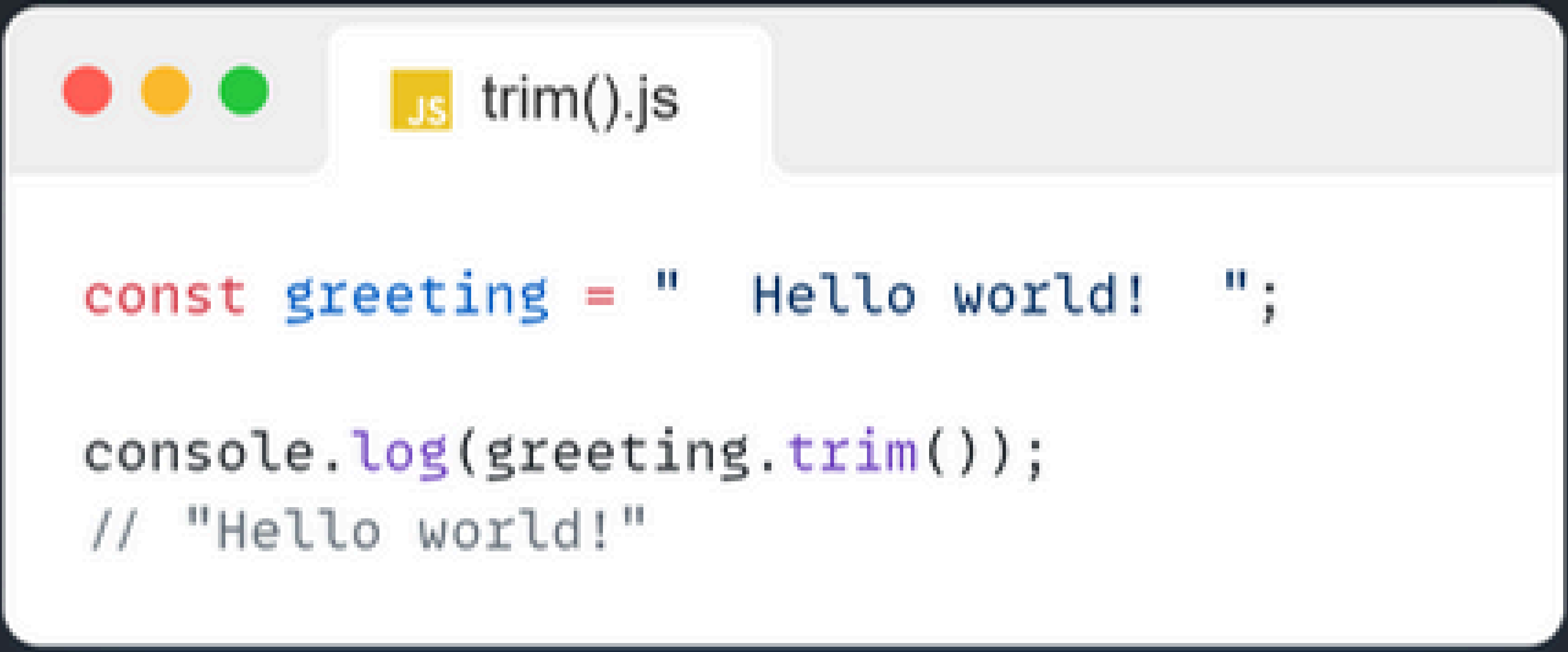
Converts a string to uppercase



```
const message = "Hello!";

console.log(message.toUpperCase());
// "HELLO!"
```

27. trim()

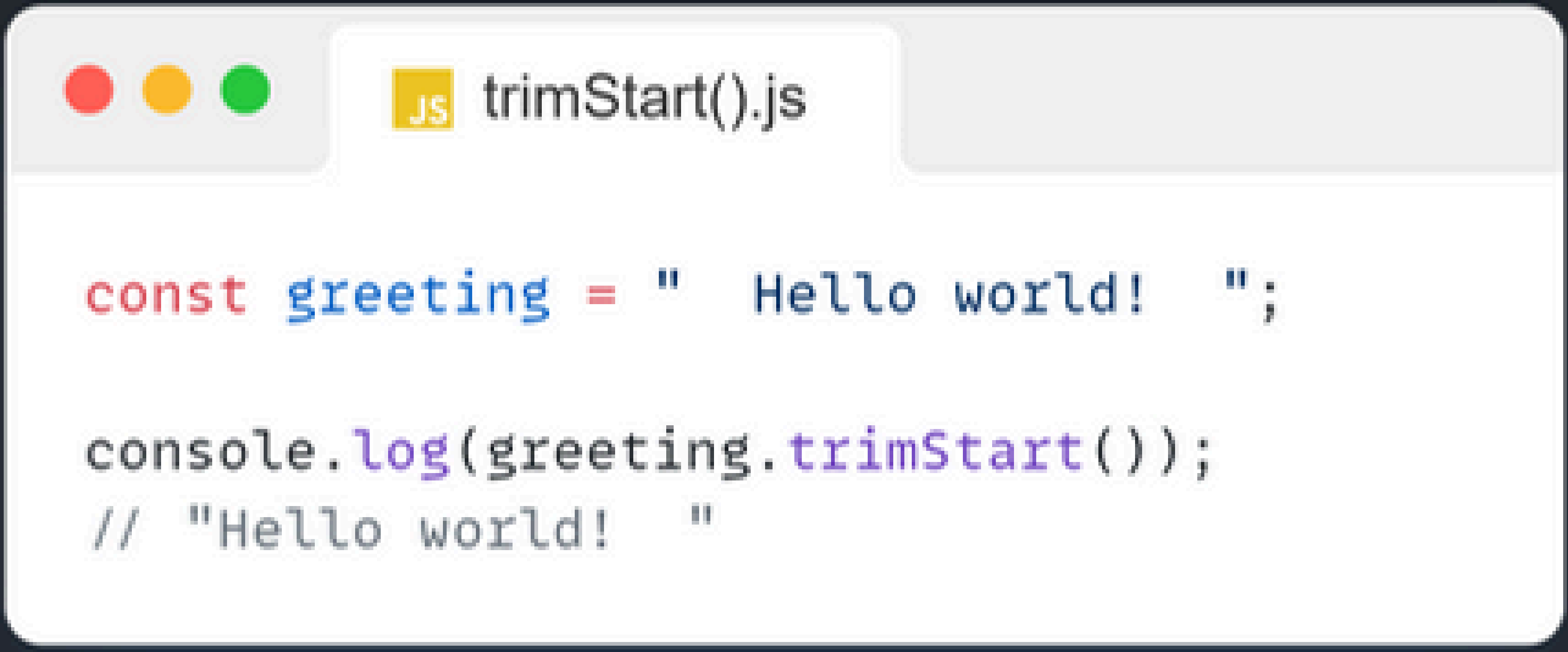
Removes whitespace from both ends



```javascript
const greeting = "  Hello world!  ";

console.log(greeting.trim());
// "Hello world!"
```

28. trimStart()

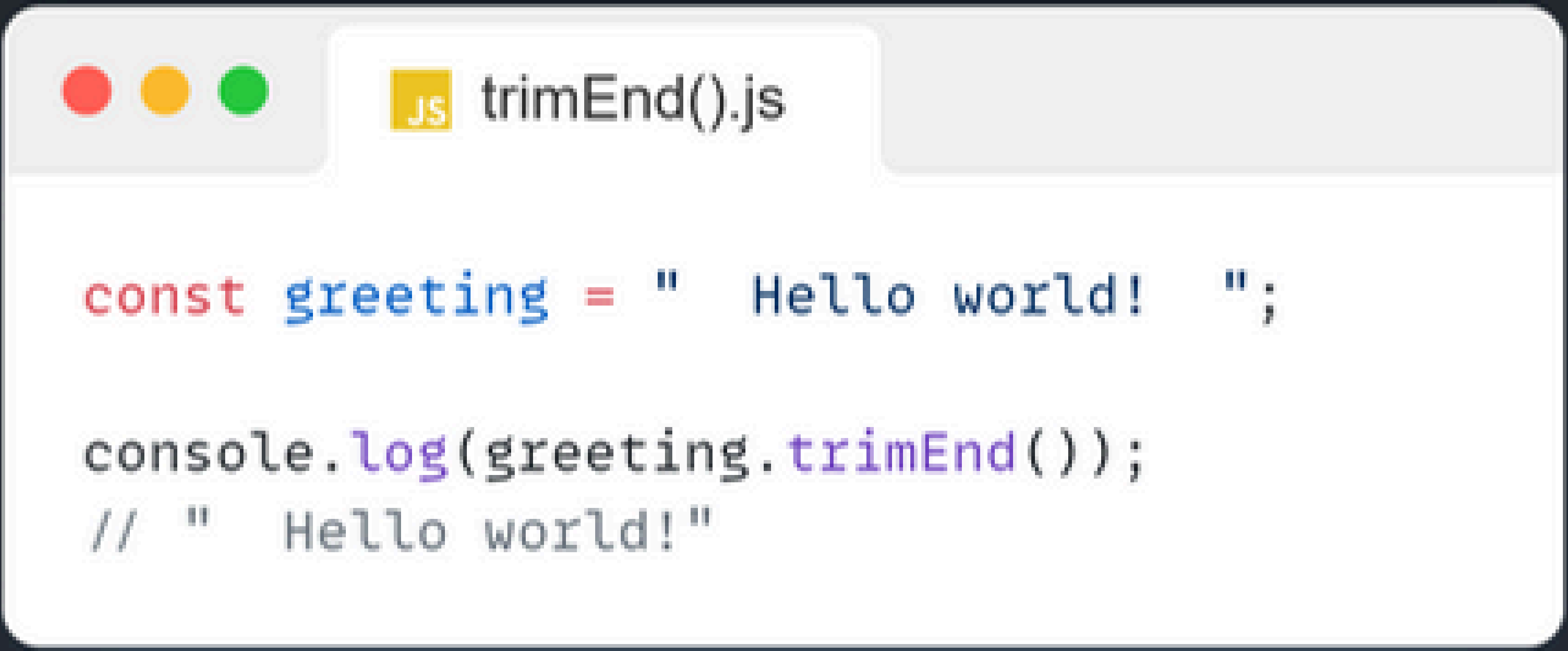
Removes whitespace from the beginning



```
const greeting = "  Hello world!  ";

console.log(greeting.trimStart());
// "Hello world!  "
```

29. trimEnd()

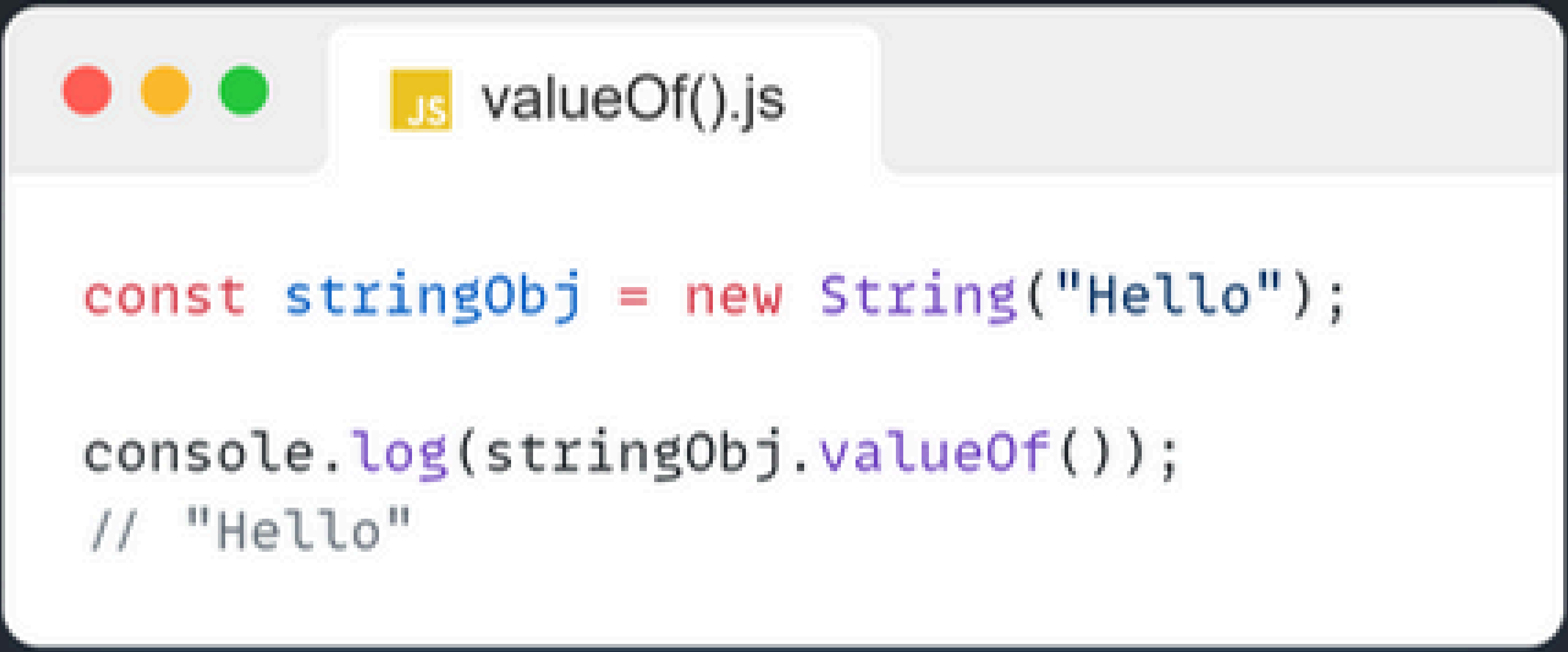
Removes whitespace from the end



```javascript
const greeting = "  Hello world!  ";

console.log(greeting.trimEnd());
// "  Hello world!"
```

30. valueOf()

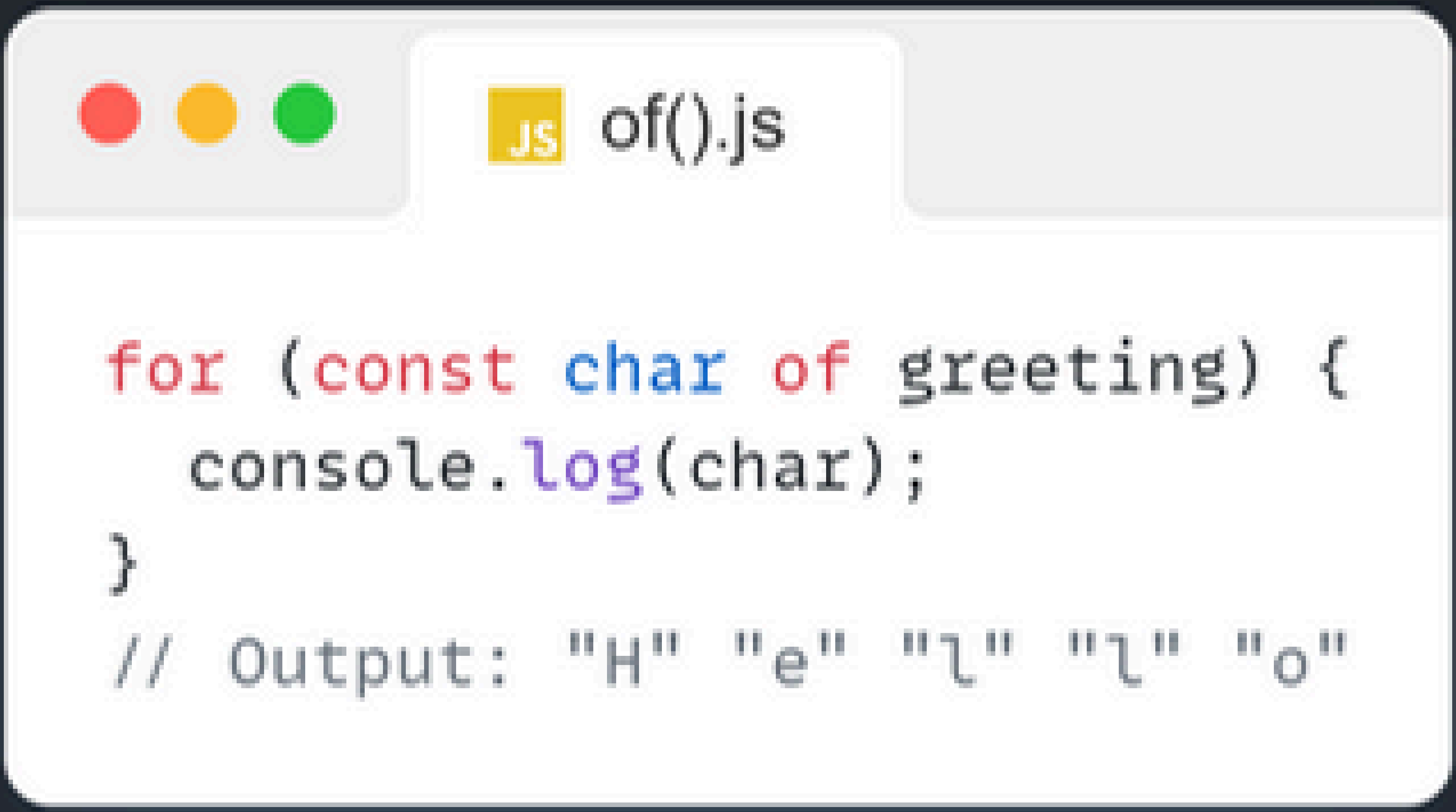
Returns the primitive value of a String object



```
const stringObj = new String("Hello");

console.log(stringObj.valueOf());
// "Hello"
```

31. [Symbol.iterator]

Makes strings iterable



```
for (const char of greeting) {
  console.log(char);
}
// Output: "H" "e" "l" "l" "o"
```

Save
Post

https://github.com/aravindFrontEnd

Aravind G