

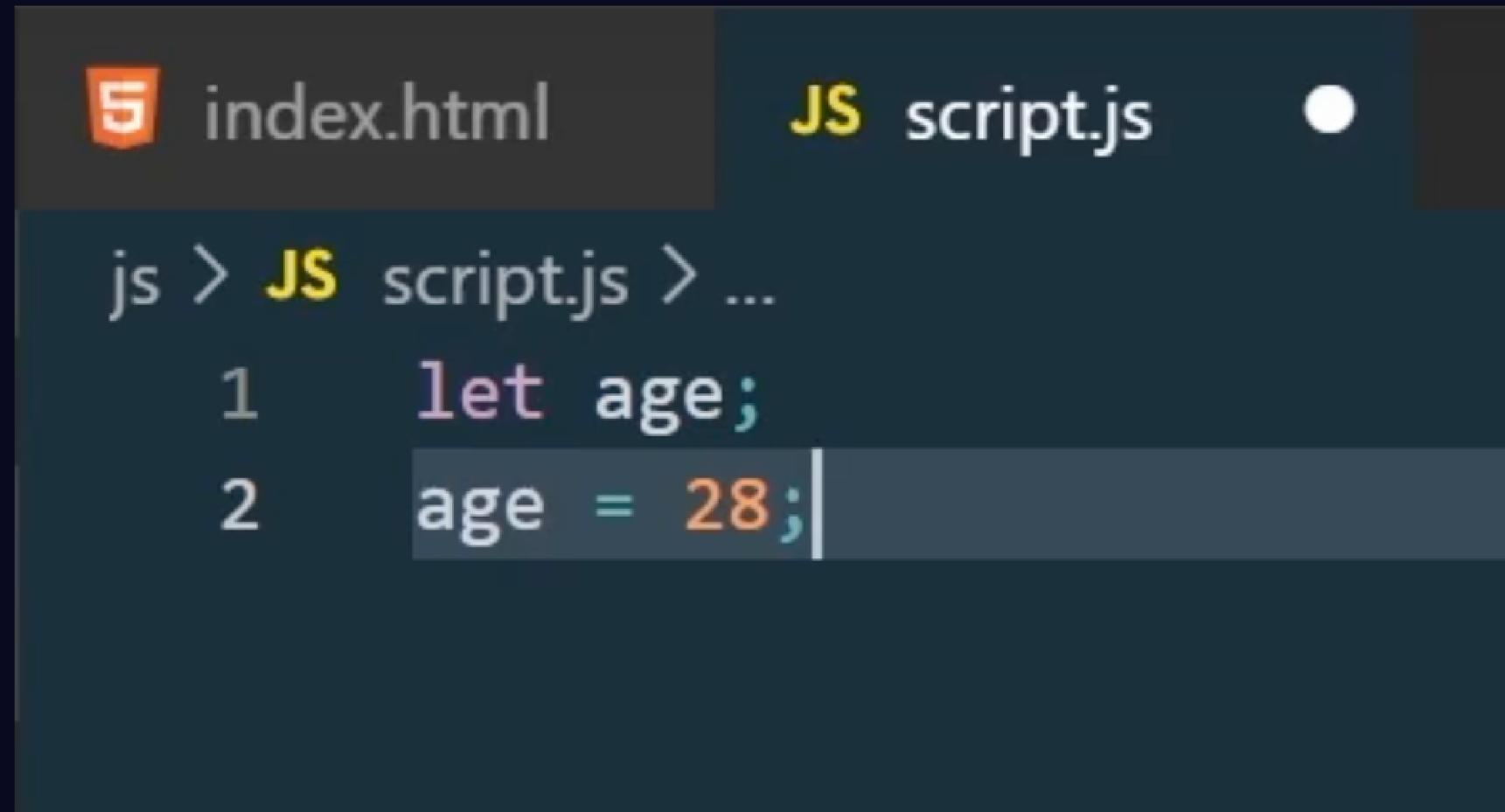
# переменные в JavaScript

# Переменные - это

именованное хранилище данных

Переменные можно использовать  
для хранения товаров, посетителей и пр.

# Переменную можно записать через **let**

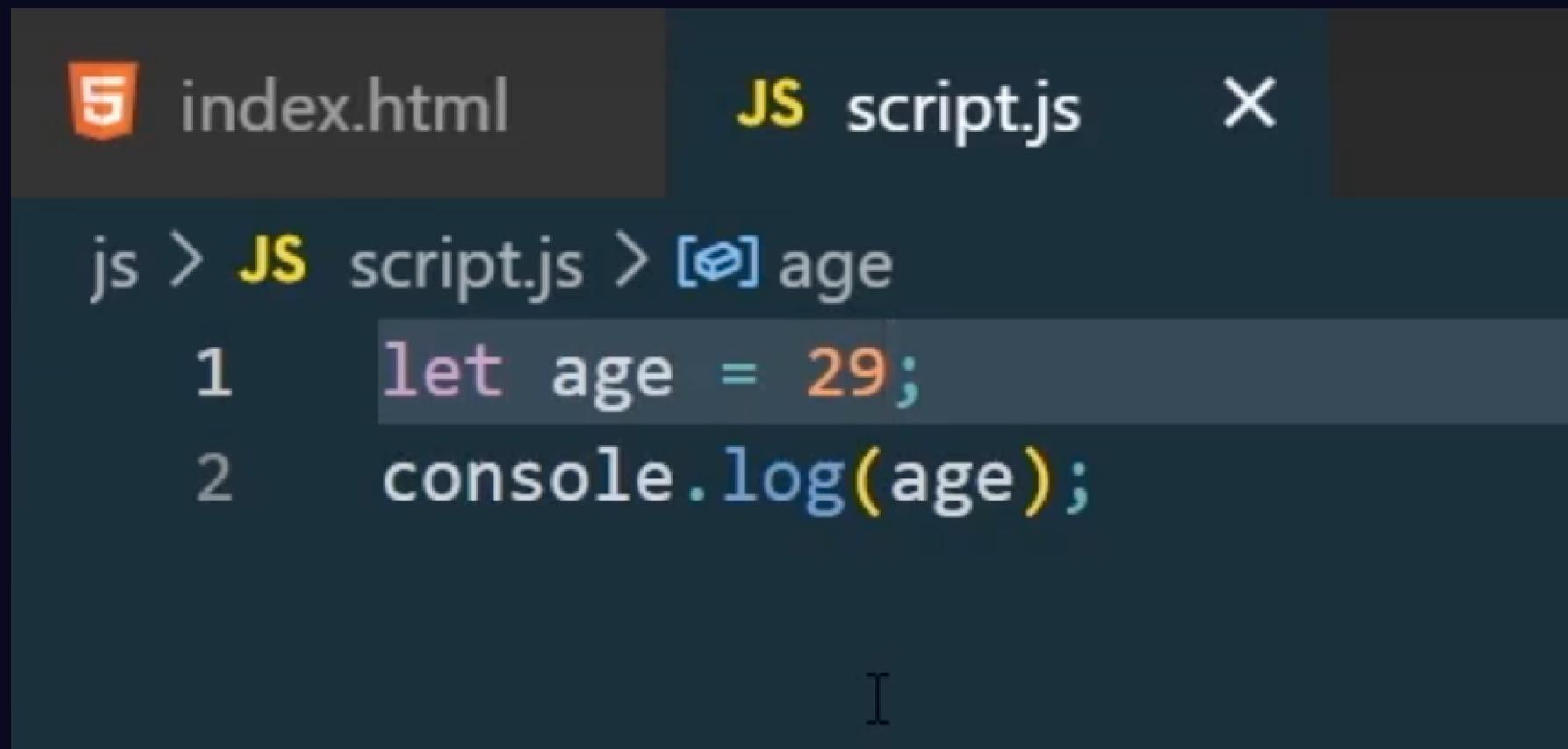


The screenshot shows a code editor interface with two tabs: 'index.html' and 'JS script.js'. The 'script.js' tab is active, indicated by a yellow icon and the word 'JS' in yellow. Below the tabs, the file content is displayed:

```
js > JS script.js > ...
1 let age;
2 age = 28;
```

Строка сохраняется в области памяти, связанной с переменной и мы можем получить к ней доступ, используя имя переменной

# Объявить переменную и задать значение можно в той же строке

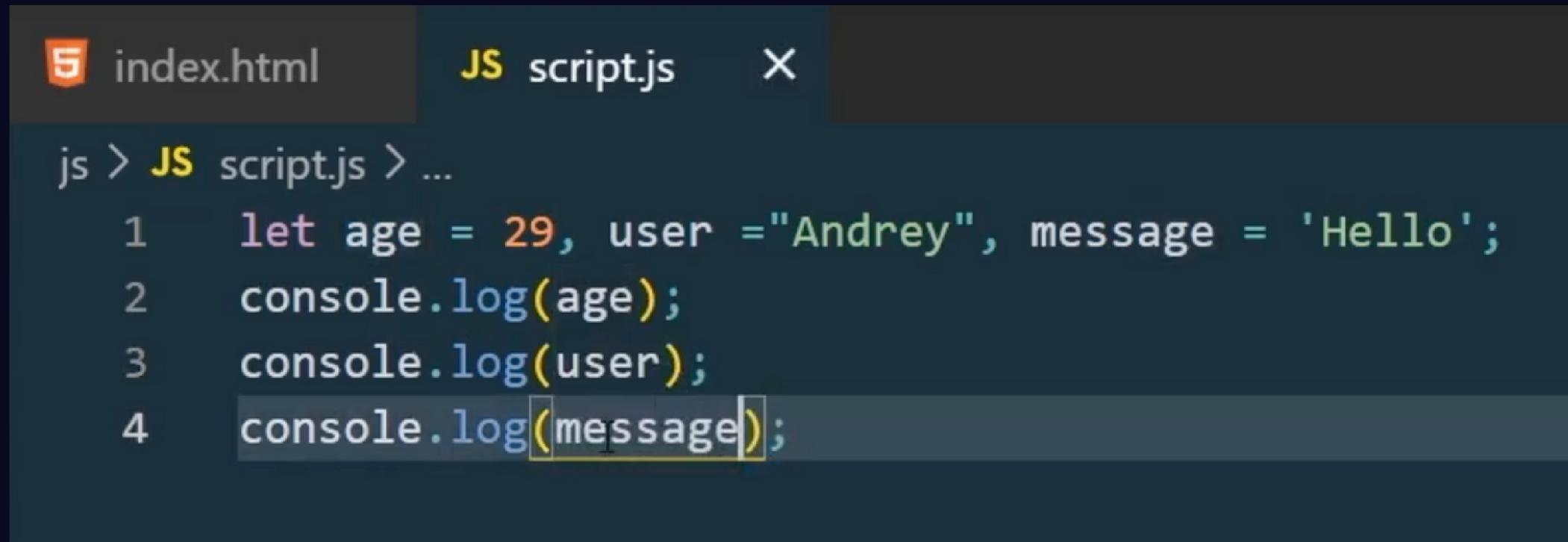


The image shows a dark-themed code editor interface. At the top, there are two tabs: 'index.html' and 'script.js'. The 'script.js' tab is currently selected, indicated by a yellow background and white text. Below the tabs, the file content is displayed in a monospaced font. The code in 'script.js' is:

```
js > JS script.js > [e] age
1 let age = 29;
2 console.log(age);
```

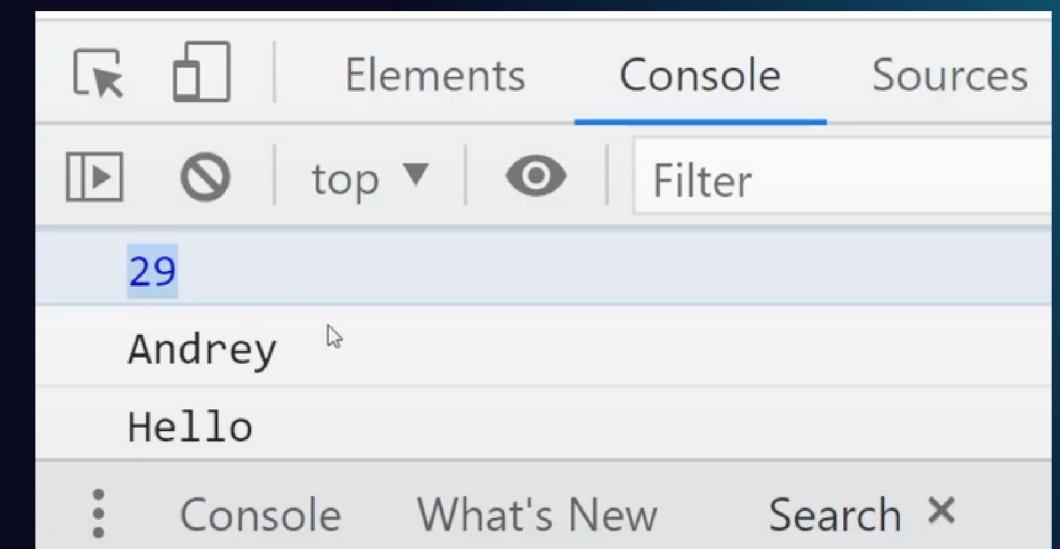
The code consists of two lines. The first line declares a variable 'age' using the 'let' keyword and assigns it the value '29'. The second line uses the 'console.log' function to output the value of 'age' to the console.

# В одной строке можно объявить несколько переменных



```
js > JS script.js > ...
1 let age = 29, user ="Andrey", message = 'Hello';
2 console.log(age);
3 console.log(user);
4 console.log(message);
```

Результат в консоли



Elements    Console    Sources

top    Filter

29
Andrey
Hello

⋮    Console    What's New    Search X

# ✗ Но мы не рекомендуем использовать такой метод

Для читаемости кода лучше объявлять  
переменные в новой строке

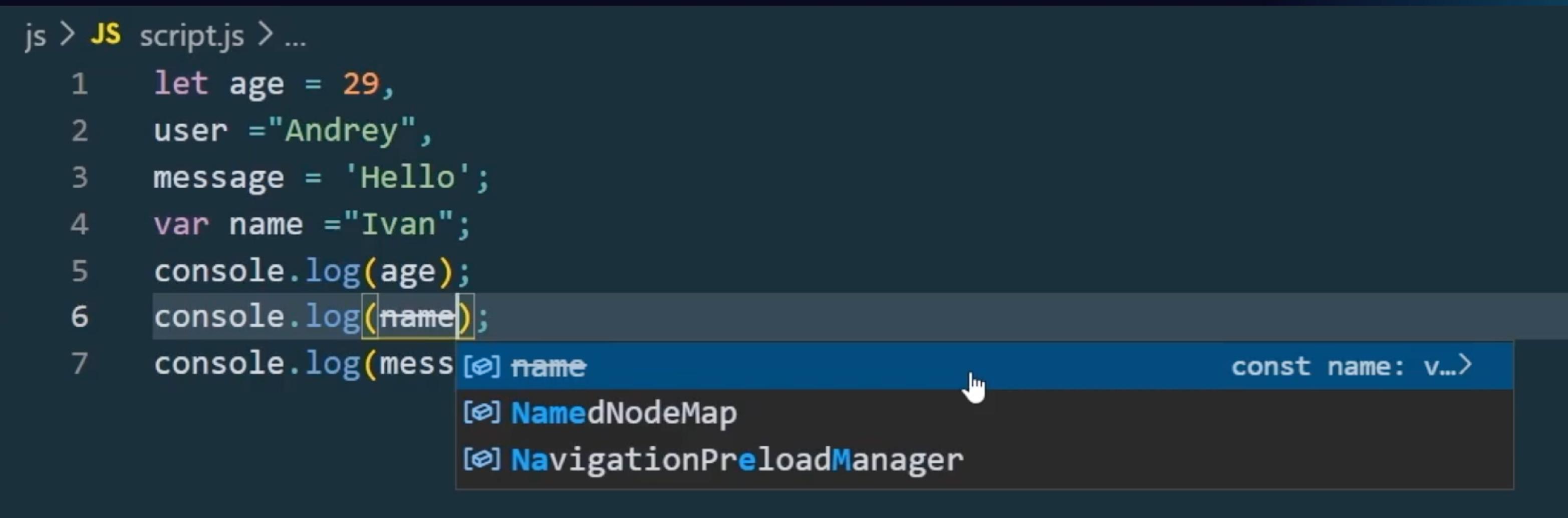
✓ Вот так

```
1 let age = 29;  
2 let user ="Andrey";  
3 let message = 'Hello';  
4 console.log(age);  
5 console.log(user);  
6 console.log(message);
```

**Устаревший метод  
объявления переменных:**

**Использовать `var` вместо `let`**

# Указав var, VS Code сообщает, что желательно указать const



A screenshot of the VS Code interface showing a code editor with the following JavaScript code:

```
js > JS script.js > ...
1 let age = 29,
2 user = "Andrey",
3 message = 'Hello';
4 var name = "Ivan";
5 console.log(age);
6 console.log(name);
7 console.log(message)
```

The cursor is positioned at the end of the word "name" in the sixth line. A code completion dropdown menu is open, displaying three suggestions:

- [?] name
- [?] NamedNodeMap
- [?] NavigationPreloadManager

The suggestion "[?] name" is highlighted with a blue background.

var до сих пор работает, в консоли мы увидим "Ivan"

**Если 2 раза указать значение для одной и той же переменной, то результатом будет последнее значение**

```
3 message = 'Hello';
4 message = 'Привет';
```

Чтение веб-документа  
идёт сверху вниз

**Но если снова объявить  
переменную, которая уже есть  
в документе, то результат  
в консоли - X ошибка**

```
1 let age = 29,  
2 user ="Andrey",  
3 message = 'Hello';  
4 let message = 'Привет';
```

# Имя переменной должно содержать:

- буквы
- цифры
- символы

# Первым символом не может быть цифра

```
js > JS script.js > ...
1 let age = 29,
2 user ="Andrey",
3 1message = 'Hello';
```

# А в конце цифру указать можно

```
js > JS script.js > [6] message1
1   let age = 29,
2   user ="Andrey",
3   message1| = 'Hello';
|
```

**При слиянии слов, каждое  
новое указывается с заглавной**

4

```
firstName = "Pavel";
```

**Но символы между ними  
указывать не стоит**

4 first-name = "Pavel";

# JavaScript чувствителен к регистру

```
1 let age = 29;  
2 let AgE = 28;  
3 console.log(AgE);
```

Результат в консоли = 28

# НЕ латинские буквы разрешены, но не рекомендуются

```
1 let age = 29;  
2 let авто = 28;  
3 console.log(авто);
```

Это связано с дальнейшей поддержкой проекта

# В JavaScript есть **слова**, которые **нельзя использовать** в качестве переменных

- let
- class
- return
- function

**Раньше записывали переменные  
без объявления переменных,  
просто присвоив им значение**

```
1 'use strict';
2 num = 5;
3 console.log(num);
```

Это возможно сделать и сейчас, но указав 'use strict' в начале документа, мы закрываем возможность использовать устаревшие правила

Кроме `let` можно еще  
использовать `const`

# Данный скрипт выведет ошибку

```
2 const num = 5;  
3 num = 6;  
4 console.log(num);
```

Переменная `num` не может изменить значение на 6, потому что `const` фиксируется на значении 5

# Можно передать значение из const

```
const COLOR_RED = "f00";
let color = COLOR_RED;
```

**Название переменной должно  
передать ясный и понятный смысл,  
чтобы другие разработчики тоже  
могли его понять**