

# Debugging Javascript Code in VS-Code

The image shows the Visual Studio Code (VS Code) interface with a JavaScript file named `script.js` open. The code defines a `Workout` class and a `Running` class that extends it.

```
1 'use strict';
2
3 class Workout {
4   date = new Date();
5   id = (Date.now() + '').slice(-10);
6   clicks = 0;
7
8   constructor(coords, distance, duration) {
9     // this.date = ...
10    // this.id = ...
11    this.coords = coords; // [lat, lng]
12    this.distance = distance; // in km
13    this.duration = duration; // in min
14  }
15
16  _setDescription() {
17    // prettier-ignore
18    const months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
19
20    this.description = `${this.type[0].toUpperCase()}${this.type.slice(1)} on ${
21      months[this.date.getMonth()]
22    } ${this.date.getDate()}`;
23  }
24
25  click() {
26    this.clicks++;
27  }
28 }
29
30 class Running extends Workout {
31   type = 'running';
32
33   constructor(coords, distance, duration, cadence) {
34     super(coords, distance, duration);
35     this.cadence = cadence;
36     this.calcPace();
37     this._setDescription();
38   }
39 }
```

The left sidebar shows the **EXPLORER** view with the file `script.js` selected. Below it, the **DEBUG CONSOLE** is visible, showing the **CALL STACK** with two entries: `Debug...` and `Work...`, both marked as **RUNNING**. The **WATCH** and **EVENT LISTENER BREAKPOINTS** sections are also visible.

The right sidebar shows the **DEBUG CONSOLE** with the **CALL STACK** view selected, displaying a list of function calls and their corresponding line numbers in the code.

The status bar at the bottom indicates the current position: **Ln 114, Col 1**, **Spaces: 2**, **UTF-8**, **LF**, **{}** JavaScript, **Go Live**, **Stylelint+**, and **Prettier**.

FileEditSelectionViewGoRunTerminalHelp

←→final

RUN AND DEBUG...JS script.js

RUN

Run and Debug

To customize Run and Debug create a launch.json file.

Show all automatic debug configurations.

JavaScript Debug Terminal

You can use the JavaScript Debug Terminal to debug Node.js processes run on the command line.

BREAKPOINTS

☐ Caught Exceptions

☐ Uncaught Except...

☒ script.js 3

☒ script.js 8

☒ script.js 40

☒ script.js 58

☒ script.js 108

☒ script.js 130

☒ script.js 136

EVE...

☐ Drag / Drop

☐ Geolocation

☒ Keyboard

☒ keydown

☒ keyup

☒ keypress

JS script.js > App > \_loadMap

```
1 'use strict';
2
3 class Workout {
4   date = new Date();
5   id = (Date.now() + '').slice(-10);
6   clicks = 0;
7
8   constructor(coords, distance, duration) {
9     // this.date = ...
10    // this.id = ...
11    this.coords = coords; // [lat, lng]
12    this.distance = distance; // in km
13    this.duration = duration; // in min
14  }
15
16  _setDescription() {
17    // prettier-ignore
18    const months = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'];
19
20    this.description = `${this.type[0].toUpperCase()}${this.type.slice(1)} on ${
21      months[this.date.getMonth()]
22    } ${this.date.getDate()}`;
23  }
24
25  click() {
26    this.clicks++;
27  }
28 }
29
30 class Running extends Workout {
31   type = 'running';
32
33   constructor(coords, distance, duration, cadence) {
34     super(coords, distance, duration);
35     this.cadence = cadence;
36     this.calcPace();
37     this._setDescription();
```

Ln 114, Col 1 Spaces: 2 UTF-8 LF {} JavaScript Go Live Stylelint+ Prettier

File Edit Selection View Go Run Terminal Help

final

Launch Ed... Welcome JS other.js JS script.js JS leaflet.js index.html JS extension.js

VARIABLES

Local: \_getLocalStorage

this: App

Script

Global

WATCH

CALL STACK

PAUSED ON BREAK...

App.\_getLocalStorage

App.constructor

<anonymous> sc...

LOADED SCRIPTS

Launch Edge against l...

Launch Edge against l...

JS script.js

JS unpkg.com/leagl...

<eval>

BREAKPOINTS

Caught Exceptions

Uncaught Except...

script.js 4

script.js 11

script.js 34

script.js 42

JS script.js > App > \_getLocalStorage > data

290 );

291

292 this.#map.setView(workout.coords, this.#mapZoomLevel, {

293   animate: true,

294   pan: {

295     duration: 1,

296   },

297 });

298

299 // using the public interface

300 // workout.click();

301 }

302

303 \_setLocalStorage() {

304   localStorage.setItem('workouts', JSON.stringify(this.#workouts));

305 }

306

307 \_getLocalStorage() {

308   const data = JSON.parse(localStorage.getItem('workouts'));

309

310   if (!data) return;

311

312   this.#workouts = data;

313

314   this.#workouts.forEach(work => {

315     this.\_renderWorkout(work);

316   });

317 }

318

319 reset() {

320   localStorage.removeItem('workouts');

321   location.reload();

322 }

323 }

324

325 const app = new App();

326

Ln 308, Col 18 Spaces: 2 UTF-8 LF JavaScript Go Live Stylelint+ Prettier

The image shows a screenshot of the Visual Studio Code editor interface. The top bar displays the 'Launch Edge' button and the file explorer showing 'index.html' and 'extension.js'. The left sidebar contains the 'VARIABLES' panel with 'Local: App' and 'Script' sections, the 'WATCH' panel, the 'CALL STACK' panel showing 'Launch ...' and 'App.constructor', the 'LOADED SCRIPTS' panel, and the 'BREAKPOINTS' panel. The main editor area shows a JavaScript file named 'script.js' with the following code:

```
JS script.js > App > _loadMap
75 const inputDuration = document.querySelector('.form__input--duration');
76 const inputCadence = document.querySelector('.form__input--cadence');
77 const inputElevation = document.querySelector('.form__input--elevation');
78
79 class App {
80   #map;
81   #mapZoomLevel = 13;
82   #mapEvent;
83   #workouts = [];
84
85   constructor() {
86     this._getPosition();
87
88     this._getLocalStorage();
89
90     form.addEventListener('submit', this._newWorkout.bind(this));
91     inputType.addEventListener('change', this._toggleElevationField);
92     containerWorkouts.addEventListener('click', this._moveToPopup.bind(this));
93   }
94
95   _getPosition() {
96     if (navigator.geolocation)
97       navigator.geolocation.getCurrentPosition(
98         this._loadMap.bind(this),
99         function () {
100           alert('Could not get your position');
101         }
102       );
103   }
104
105   _loadMap(position) {
106     const { latitude } = position.coords;
107     const { longitude } = position.coords;
108
109     const coords = [latitude, longitude];
110
111     this.#map = L.map('map').setView(coords, this.#mapZoomLevel);
```

The bottom status bar shows 'Ln 108, Col 1', 'Spaces: 2', 'UTF-8', 'LF', '{ } JavaScript', 'Go Live', 'Stylelint+', and 'Prettier'.