JavaScript sessions work differently. Generally, a session expires when the user closes their browser (but bear in mind that the concept of “closing a browser” is not clear on mobile devices). When a session expires, all data stored in the session storage of a Web browser is removed.

There’s no need to explicitly initialize a session because in JavaScript a session takes the form of the global sessionStorage object and is always present.

It’s up to us to write data into the current session.

Session data comes in the form of key-value pairs, and the value of each key may contain only strings.

To write data, we can use the sessionStorage.setItem(name, value) method:

```
sessionStorage.setItem( "total", 120 );
```

This value will be available until the session expires, unless we use sessionStorage.removeItem(“total”) to remove the named key or we call sessionStorage.clear() to entirely remove all keys and values from the session storage.

We can also update its value by using sessionStorage.setItem() again with a new value.

Objects may be stored in session storage by first turning them into JSON strings (with JSON.stringify()) and then back into JavaScript objects (with JSON.parse())