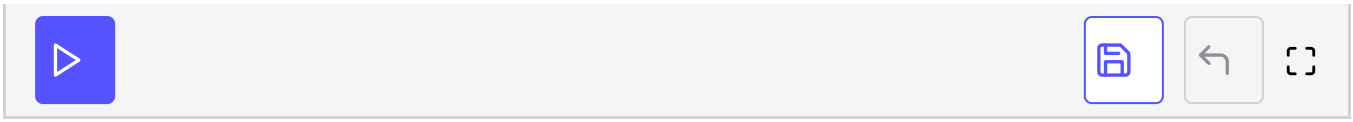
Global Symbol Registry

relationship between symbol values and their string keys

ES6 has a global resource for creating symbols: the symbol registry. The symbol registry provides us with a one-to-one relationship between strings and symbols. The registry returns symbols using `Symbol.for( key )`.

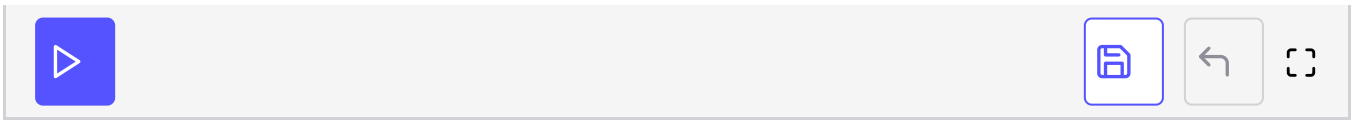
`Symbol.for( key1 ) === Symbol.for( key2 )` whenever `key1 === key2`. This correspondance works even across service workers and iframes.

```
let privateProperty1 = Symbol.for( 'firstName' );
let privateProperty2 = Symbol.for( 'firstName' );

myObject[ privateProperty1 ] = 'Dave';
myObject[ privateProperty2 ] = 'Zsolt';

console.log( myObject[ privateProperty1 ] );
// Zsolt
```



As there is a one-to-one correspondence between symbol values and their string keys in the symbol registry, it is also possible to retrieve the string key. Use the `Symbol.keyFor` method.

```
console.log(Symbol.keyFor( privateProperty1 ));
//> "firstName"

console.log(Symbol.keyFor( Symbol() ));
//> undefined
```



Now, let's talk about making symbols private.