





Working with variables

Learn how to harness console.log to output our code so we can see what's going on. Learn more about using and copying variables.

We'll cover the following

- Printing
- Reassigning Variables
- Assigning Using Variables
- Quiz

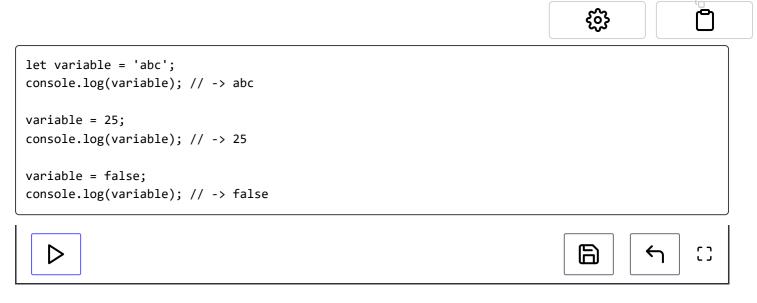
Printing#

To print something out, we use something called <code>console.log()</code>. We write that phrase with either a value or a variable name inside the parentheses. We see that item being printed out when we run our code. This is a very valuable tool as it lets us see what our code is doing.

```
1 let bool = true;
2 let string = 'abc';
3
4 console.log('Hello!'); // -> Hello!
5 console.log(false); // -> false
6 console.log(bool); // -> true
7 console.log(string); // -> abc
```

Reassigning Variables#

We can give a variable a new value if we like.



Note that we don't use the let keyword again. We use it once the first time we create a variable and then we can reassign values without it.

If we use let again, we'll get an error.

```
let variable = 'abc';
console.log(variable);

let variable = 25; // -> Compilation Error
console.log(variable);

let variable = false;
console.log(variable);

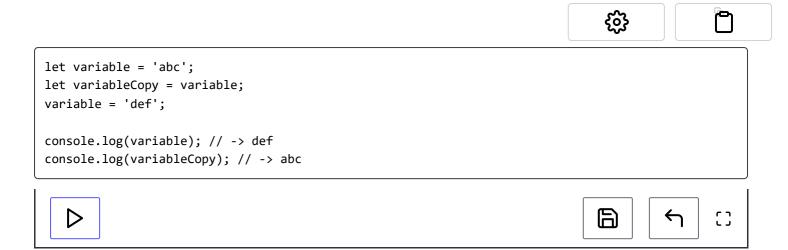
\[ \begin{align*} \begin{al
```

Assigning Using Variables#

We can copy the value of a variable into another variable using = .

```
let variable = 'abc';
let variableCopy = variable;
console.log(variableCopy); // -> abc
```

Once we copy a variable, the new variable is distinct from the old one. They have no relation to each other. If we change one, the other remains unchanged.



Now we know how to reassign variables and have seen how they work when we run our code.

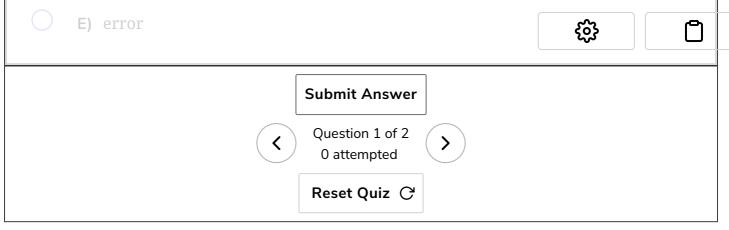
Quiz#

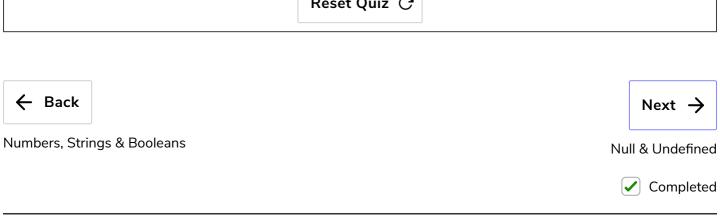
Feel free to test your understanding.

1 What will this code print?

let variable = 'abc';
let variableCopy = variable;
variableCopy = 'def';
variable = variableCopy;
console.log(variable);
A) variable

- B) variableCopy
- C) abc
- O) def





! Report an Issue