

Arrow Functions

41

→ One of the most famous feature in modern JavaScript is the arrow function. ES6 arrow function provide us an alternative way to write a more concise and shorter syntax compared to the traditional function expression Syntax.

LET myFunc = (ARG1, ARG2, ... ARGN) => EXPRESSION

EX: LET ADDITION = FUNCTION (X, Y) {
 RETURN X + Y;
}

CONSOLE.LOG (ADDITION (10, 10));

→ Same code in Arrow function:

LET ADDITION = (X, Y) => X + Y;

CONSOLE.LOG (ADDITION (10, 10));

→ In arrow function if the ~~statement~~ function has one statement, and the

statement returns a value or string, we can remove the brackets and the return keyword.

→ Limitations of Arrow Functions:

→ An arrow function is an alternative to a traditional function expression, but there are some limitations:

→ Arrow functions do not have its own bindings to this or super, and should not be used as methods.

→ It is not suitable for the call, apply and bind methods, which generally rely on establishing a scope.

→ Arrow functions cannot be used as constructors.

→ Arrow functions are a powerful addition to ES6, but we have to be careful while using them. There are some places where arrow functions are not suitable, and this can cause difficulty for us to track errors, especially if we do not understand how they really work. Arrow functions are the best choice when working with closures or callbacks, but it is not a good choice when working with objects methods or constructors.