

Arrow Function Expressions

based on MDN web docs
(Arrow Function Expressions)

by Dr. K

Arrow Function Expressions

Two factors influenced the introduction of arrow functions:

- the need for **shorter functions**
- the behavior of the **this keyword**

Arrows: Shorter Functions

The following code portions are equivalent:

```
const elements = [  
  "Hydrogen",  
  "Helium",  
  "Lithium",  
  "Beryllium"  
];
```

```
elements.map(function(element) {  
  return element.length;  
});
```

returns [8, 6, 7, 9]
when `elements` is the
array on the left

```
elements.map((element) => {  
  return element.length;  
});
```

the function above can
be rewritten with the
arrow function here

```
elements.map(element => {  
  return element.length;  
});
```

when there is only one
parameter, we can
remove the parens

```
elements.map(element => element.length);
```

when the only statement is
a return, we can remove the
return and the curly braces

Arrows: *this* keyword

Before arrow functions, every new function defined its own *this* value based on how the function was called.

An arrow function does not have its own *this*, so the *this* value of the enclosing lexical scope is used.