

ARROW FUNCTIONS

Concepts and Examples

! (INTERNET EXPLORER)

Current aligned	Usage relative	Date relative	Apply filters	Show all	?	
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *
		2 - 21	4 - 44	3.1 - 9.1	10 - 31	3.2 - 9.3
6 - 10	12 - 17	22 - 63	45 - 70	10 - 11.1	32 - 56	10 - 11.4
11	18	64	71	12	57	12.1
		65 - 66	72 - 74	TP		

WHAT ARE ARROW FUNCTIONS

A notation that is a shorter form of the standard function expression.

```
( modes ) => {  
  fetch(modes)  
    .then( raw    => { return raw.json(); } )  
    .then( modes => { this.setState(modes); } );  
}
```

SYNTAX

(param1, param2, ...paramN) => { expressions; }

oneParam => expression; // concise body syntax

(oneParam) => expression;

(oneParam) => { expression(s); } // block body syntax

() => { expressions; }

NOTE:

Arrow functions do not have the concept of **this**

this - the context of where something (a variable, function, object etc.. was used or invoked. In other words, the surrounding context

NOT USABLE IN CONSTRUCTORS


Arrow functions cannot be used as constructors



```
let AuditComponent = () => {  
  this.createDate = new Date();  
  this.logger = LoggerFactory.getDefaultLogger();  
}
```

DON'T PUT A LINE BREAK

```
let BreakingNewsComponent = (  
  => {  
    this.article = BreakingNewsService.fetch();  
  }  
)
```



Developer hits the
return key on their
keyboard.

EMPTY ARROW FUNCTIONS

Returns
undefined

$() \Rightarrow \{ \}$

PARSING ORDER

```
let callback;  
  
callback = callback || function() {}; // ok  
  
callback = callback || () => {};  
// SyntaxError: invalid arrow-function arguments  
  
callback = callback || (() => {}); // ok
```

NO .PROTOTYPE

Arrow functions do not have the .prototype property



```
let MenuComponent = () => {  
    console.log(MenuComponent.prototype);  
    //undefined  
}
```

RETURNING OBJECT LITERALS

You need to enclose (or wrap) the literal with parentheses.

```
let getDefaultGreetings = ( ) => {  
    return ( {en_us: 'Hello', it: 'Caio', ru: 'Privet'} );  
}
```

JQUERY

Safe:

```
$('#target').on('click', event =>  
    $(event.currentTarget).hide(); );
```

Not Going to Work:

```
$('#broken').on('error', ( ) =>  
    $(this).hide(); );
```