

# Trailing Commas

**Call Expressions & Function Parameters**

- Helps with VCS authorship attribution (git/svn blame, hg/cvs annotate, etc)

```
1: function clownsEverywhere(  
2:     param1,  
3:     param2  
4:  
5: ) { /* ... */ }  
6:  
7: clownsEverywhere(  
8:     'foo',  
9:     'bar'  
10:  
11: );
```

- Helps with VCS authorship attribution (git/svn blame, hg/cvs annotate, etc)

```
1: function clownsEverywhere(  
2:     param1,  
3:     param2 ,  
4:     param3  
5: ) { /* ... */ }  
6:  
7: clownsEverywhere(  
8:     'foo',  
9:     'bar' ,  
10:    'baz'  
11: );
```

- Helps with VCS authorship attribution (git/svn blame, hg/cvs annotate, etc)

```
1: function clownseverywhere(  
2:     param1,  
3:     param2,  
4:     param3  
5: ) { /* ... */ }  
6:  
7: clownseverywhere(  
8:     'foo',  
9:     'bar',  
10:    'baz'  
11: );
```

- Helps with VCS authorship attribution (git/svn blame, hg/cvs annotate, etc)

```
1: function clownsEverywhere(  
2:     param1,  
3:     param2,  
4:  
5: ) { /* ... */ }  
6:  
7: clownsEverywhere(  
8:     'foo',  
9:     'bar',  
10:  
11: );
```

- Helps with VCS authorship attribution (git/svn blame, hg/cvs annotate, etc)

```
1: function clownsEverywhere(  
2:     param1,  
3:     param2,  
4:     param3,  
5: ) { /* ... */ }  
6:  
7: clownsEverywhere(  
8:     'foo',  
9:     'bar',  
10:    'baz',  
11: );
```

- Makes code-gen easier (in some cases)
- More consistent with ES5 array/object literals
- Still lintable (if you're not a fan of the style, that's ok!)
- Other langs support it (python, hack, dlang, ...)

# Function Definitions

*FunctionDeclaration*[Yield, Default] :

**function** *BindingIdentifier*[?Yield, ?Default] ( *FormalParameters* ) { *FunctionBody* }

*FunctionExpression* :

**function** *BindingIdentifier*<sub>opt</sub> ( *FormalParameters* ) { *FunctionBody* }

*StrictFormalParameters*[Yield, GeneratorParameter] :

*FormalParameters*[?Yield, ?GeneratorParameter]

*FormalParameters*[Yield, GeneratorParameter] :

[empty]

*FormalParameterList*[?Yield, ?GeneratorParameter]

*FormalParameterList*[Yield, GeneratorParameter] :

*FunctionRestParameter*[?Yield]

*FormalsList*[?Yield, ?GeneratorParameter]

*FormalsList*[?Yield, ?GeneratorParameter], *FunctionRestParameter*[?Yield]



# Function Definitions

*FunctionDeclaration*[Yield, Default] :

**function** *BindingIdentifier*[?Yield, ?Default] ( *FormalParameters* ) { *FunctionBody* }

*FunctionExpression* :

**function** *BindingIdentifier*<sub>opt</sub> ( *FormalParameters* ) { *FunctionBody* }

*StrictFormalParameters*[Yield, GeneratorParameter] :

*FormalParameters*[?Yield, ?GeneratorParameter]

*FormalParameters*[Yield, GeneratorParameter] :

[empty]

*FormalParameterList*[?Yield, ?GeneratorParameter]

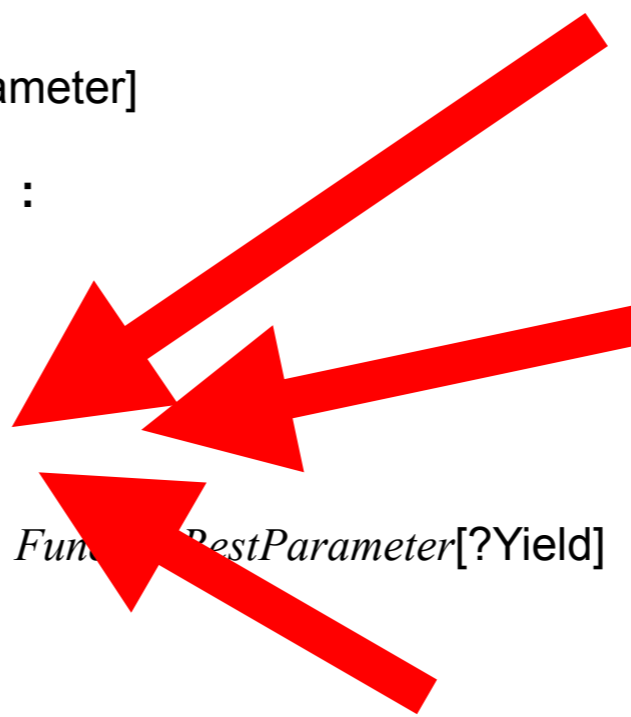
*FormalParameterList*[Yield, GeneratorParameter] :

*FunctionRestParameter*[?Yield]

*FormalsList*[?Yield, ?GeneratorParameter]

*FormalsList*[?Yield, ?GeneratorParameter] ,

*FormalsList*[?Yield, ?GeneratorParameter] , *FunctionRestParameter*[?Yield]



# Call Expressions

*CallExpression* :

*MemberExpression Arguments*

*CallExpression Arguments*

*CallExpression* [ *Expression* ]

*CallExpression* . *IdentifierName*

*Arguments* :

( )

( *ArgumentList* )

# Call Expressions

*CallExpression* :

*MemberExpression Arguments*

*CallExpression Arguments*

*CallExpression* [ *Expression* ]

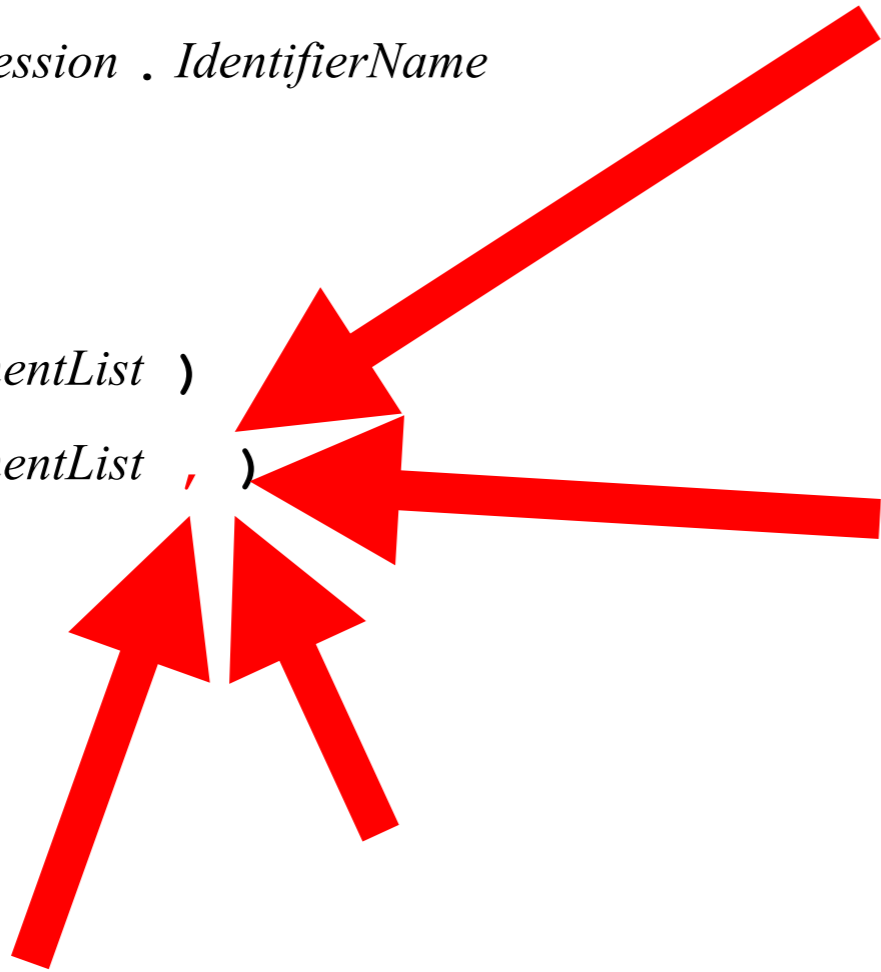
*CallExpression* . *IdentifierName*

*Arguments* :

( )

( *ArgumentList* )

( *ArgumentList* , )



<https://github.com/jeffmo/es-trailing-function-commas>