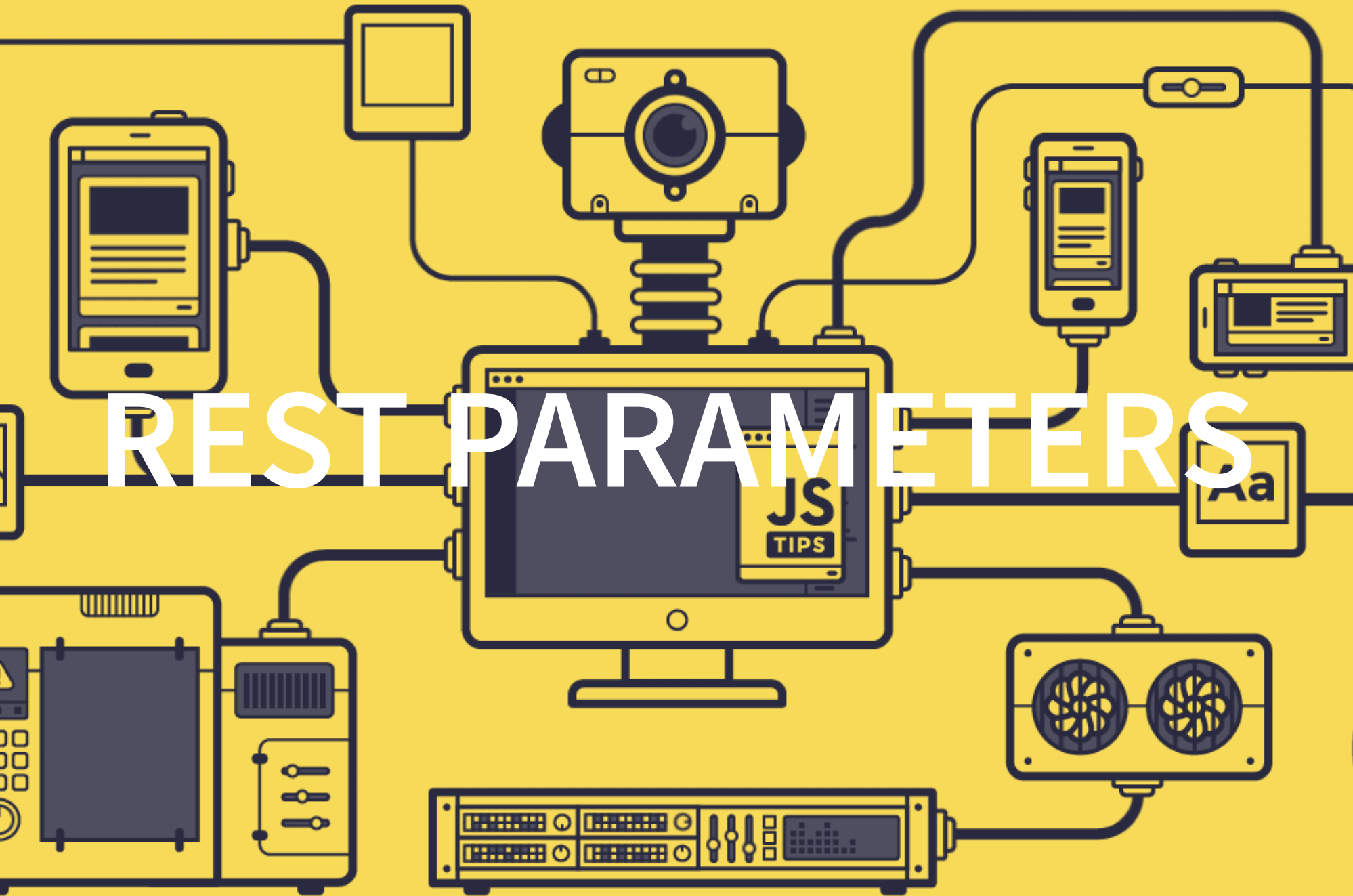


# REST PARAMETERS



# WHAT IS IT?

Great syntax sugar, which helps to work with indefinite amount of parameters

## Raw example

```
1 function func (...rest) {  
2     console.log(rest);  
3 }  
4  
5 func(1, 2, 3);  
6 // [1, 2, 3]
```

Most common – as the arguments, which are left

```
1 function func (a, b, ...rest) {  
2     // `a` and `b` are extracted from `rest`  
3     console.log(rest);  
4 }  
5  
6 func('needed', 'crucial', 'everything else');  
7 // ['everything else']
```

# Always the last

```
1 function func (a, ...rest, b) {  
2     console.log(rest);  
3 }  
4  
5 func('needed', 'crucial', 'everything else');  
6 // SyntaxError: Rest parameter must be last formal parameter
```

# **DRAWBACKS OF OLD SIMILAR APPROACH**

arguments

Even shorter! You may say

```
1 function func () {  
2     console.log(arguments);  
3 }  
4  
5 func(1, 2, 3); // [1, 2, 3]
```

# Always all parameters, NOT only the rest

```
1 function func (a, b) {  
2     console.log(arguments);  
3 }  
4  
5 func(1, 2, 3); // [1, 2, 3]
```



## ~~Smells~~ Looks like an array

```
1 function func (a, b) {  
2     console.log(arguments[0] + arguments[1]);  
3 }  
4  
5 func(1, 2); // 3
```

## But not an Array

```
1 function func (a, b) {  
2     console.log(arguments.map);  
3     console.log(arguments.shift);  
4     console.log(arguments.concat);  
5 }  
6  
7 func(1, 2); // undefined x 3
```

# Oldfags used to use

```
1 function func (a, b) {  
2     var myGreatArrayWithAllMethods =  
3         Array.prototype.slice.call(arguments);  
4     console.log(myGreatArrayWithAllMethods.map);  
5     console.log(myGreatArrayWithAllMethods.shift);  
6     console.log(myGreatArrayWithAllMethods.concat);  
7 }  
8  
9 func(1, 2) // function { [native code] } x 3
```

# Oldfags used to use

```
1 function func (a, b) {  
2     var myGreatArrayWithAllMethods =  
3         Array.prototype.slice.call(arguments);  
4     console.log(myGreatArrayWithAllMethods.map);  
5     console.log(myGreatArrayWithAllMethods.shift);  
6     console.log(myGreatArrayWithAllMethods.concat);  
7 }  
8  
9 func(1, 2) // function { [native code] } x 3
```

# Arrow functions do not have arguments

```
1 const func = () => console.log(arguments)
2
3 func(1, 2) // ReferenceError: arguments is not defined
```

...or worse, they use arguments of a parent

```
1 (function () {  
2     const func = () => console.log(arguments)  
3  
4     func(1, 2) // ['ho-ho-ho', 'xmas']  
5 })('ho-ho-ho', 'xmas');
```

# Hidden dangerous features

```
1 (function callFunction () {  
2     console.log(arguments.callee);  
3 })();  
4 // content of `callFunction`
```

...which are restricted in 'use strict' mode

```
1 (function callFunction () {  
2     'use strict';  
3     console.log(arguments.callee);  
4 })();  
5 // TypeError: 'callee' may not be accessed on strict mode
```



# **ADVANTAGES OF**

rest parameters

**Only** rest parameters (left)  
**Always** an **array**