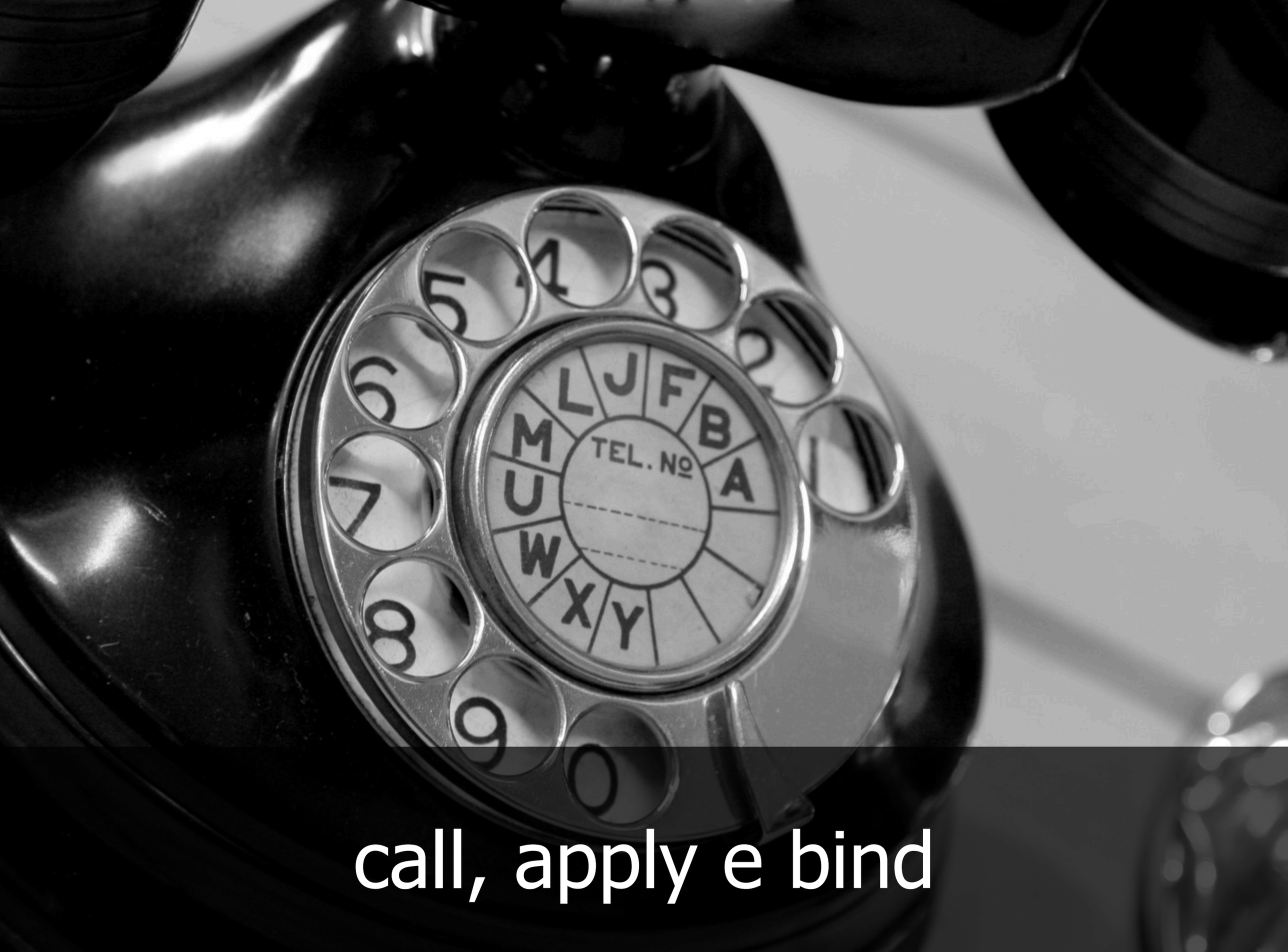

call, apply e bind

Por meio das operações call e apply é possível invocar uma função passando o this por parâmetro

call_apply_bind_1.js — javascriptmasterclass

call_apply_bind_1.js

```
1  const circle = {
2      radius: 10,
3      calculateArea() {
4          return Math.PI * Math.pow(this.radius, 2);
5      }
6  };
7  console.log(circle.calculateArea());
8
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_1.js
314.1592653589793
rodrigobranas:javascriptmasterclass $
```

call_apply_bind_2.js — javascriptmasterclass

call_apply_bind_2.js

```
1  const calculateArea = function() {
2      return Math.PI * Math.pow(this.radius, 2);
3  };
4  const circle = {
5      radius: 10,
6      calculateArea
7  };
8  console.log(circle.calculateArea());
9
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_2.js
314.1592653589793
rodrigobranas:javascriptmasterclass $
```

call_apply_bind_3.js — javascriptmasterclass

call_apply_bind_3.js

```
1  const calculateArea = function() {
2      return Math.PI * Math.pow(this.radius, 2);
3  };
4  const circle = {
5      radius: 10,
6      calculateArea
7  };
8  console.log(calculateArea.call(circle));
9
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_3.js
314.1592653589793
rodrigobranas:javascriptmasterclass $
```

call_apply_bind_4.js — javascriptmasterclass

call_apply_bind_4.js

```
1  const calculateArea = function() {
2      return Math.PI * Math.pow(this.radius, 2);
3  };
4  const circle = {
5      radius: 10,
6      calculateArea
7  };
8  console.log(calculateArea.apply(circle));
9
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_4.js
314.1592653589793
rodrigobranas:javascriptmasterclass $
```

Qual é a diferença entre call e apply?

call_apply_bind_5.js — javascriptmasterclass

call_apply_bind_5.js

```
1   const calculateArea = function(fn) {
2       return fn(Math.PI * Math.pow(this.radius, 2));
3   };
4   const circle = {
5       radius: 10,
6       calculateArea
7   };
8   console.log(calculateArea.call(circle, Math.round));
9   console.log(calculateArea.apply(circle, [Math.ceil]));
10
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_5.js
314
315
rodrigobranas:javascriptmasterclass $
```

A operação bind permite encapsular o this dentro da função, retornando-a

call_apply_bind_6.js — javascriptmasterclass

call_apply_bind_6.js

```
1   const calculateArea = function(fn) {
2       return fn(Math.PI * Math.pow(this.radius, 2));
3   };
4   const circle = {
5       radius: 10,
6       calculateArea
7   };
8   const calculateAreaForCircle = calculateArea.bind(circle);
9   console.log(calculateAreaForCircle(Math.round));
10  console.log(calculateAreaForCircle(Math.ceil));
11
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node c
all_apply_bind/call_apply_bind_6.js
314
315
rodrigobranas:javascriptmasterclass $
```