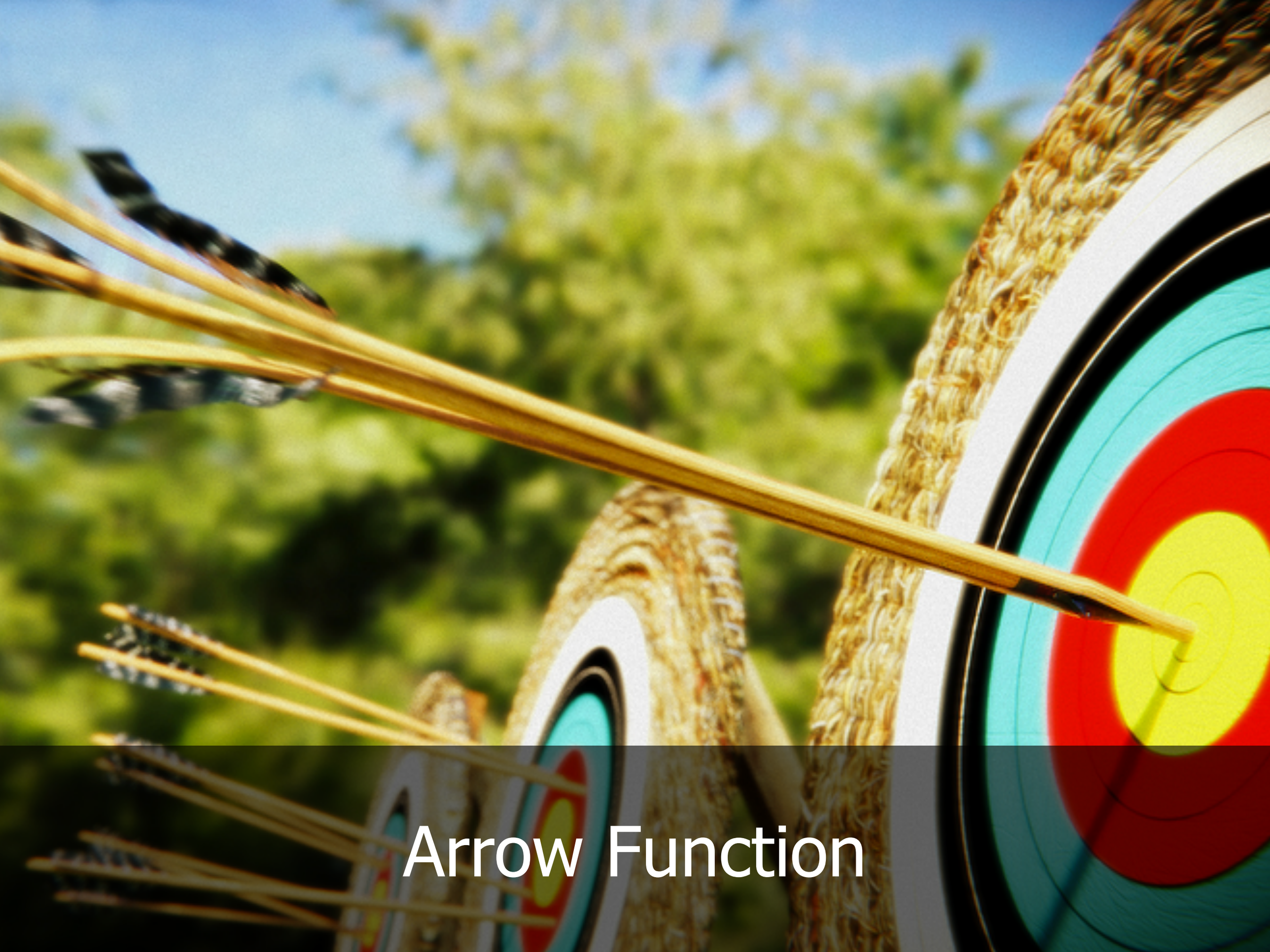


Arrow Function

As arrow functions tem uma abordagem mais simples e direta para escrever uma função e podem melhorar a legibilidade do código em diversas situações

arrow_function_1.js — javascriptmasterclass

JS arrow_function_1.js

```
1   const sum = function(a, b) {
2       return a + b;
3   };
4   const subtract = function(a, b) {
5       return a - b;
6   };
7   const calculator = function(fn) {
8       return function(a, b) {
9           return fn(a, b);
10      };
11  };
12  console.log(calculator(sum)(2, 2));
13  console.log(calculator(subtract)(2, 2));
14
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_1.js
4
0
rodrigobranas:javascriptmasterclass $
```

arrow_function_2.js — javascriptmasterclass

JS arrow_function_2.js

```
1   const sum = (a, b) => {
2       return a + b;
3   };
4   const subtract = (a, b) => {
5       return a - b;
6   };
7   const calculator = (fn) => {
8       return (a, b) => {
9           return fn(a, b);
10      };
11  };
12  console.log(calculator(sum)(2, 2));
13  console.log(calculator(subtract)(2, 2));
14
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_2.js
4
0
rodrigobranas:javascriptmasterclass $
```

arrow_function_3.js — javascriptmasterclass

JS arrow_function_3.js

```
1   const sum = (a, b) => a + b;
2   const subtract = (a, b) => a - b;
3   const calculator = (fn) => (a, b) => fn(a, b);
4   console.log(calculator(sum)(2, 2));
5   console.log(calculator(subtract)(2, 2));
6
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_3.js
4
0
rodrigobranas:javascriptmasterclass $
```

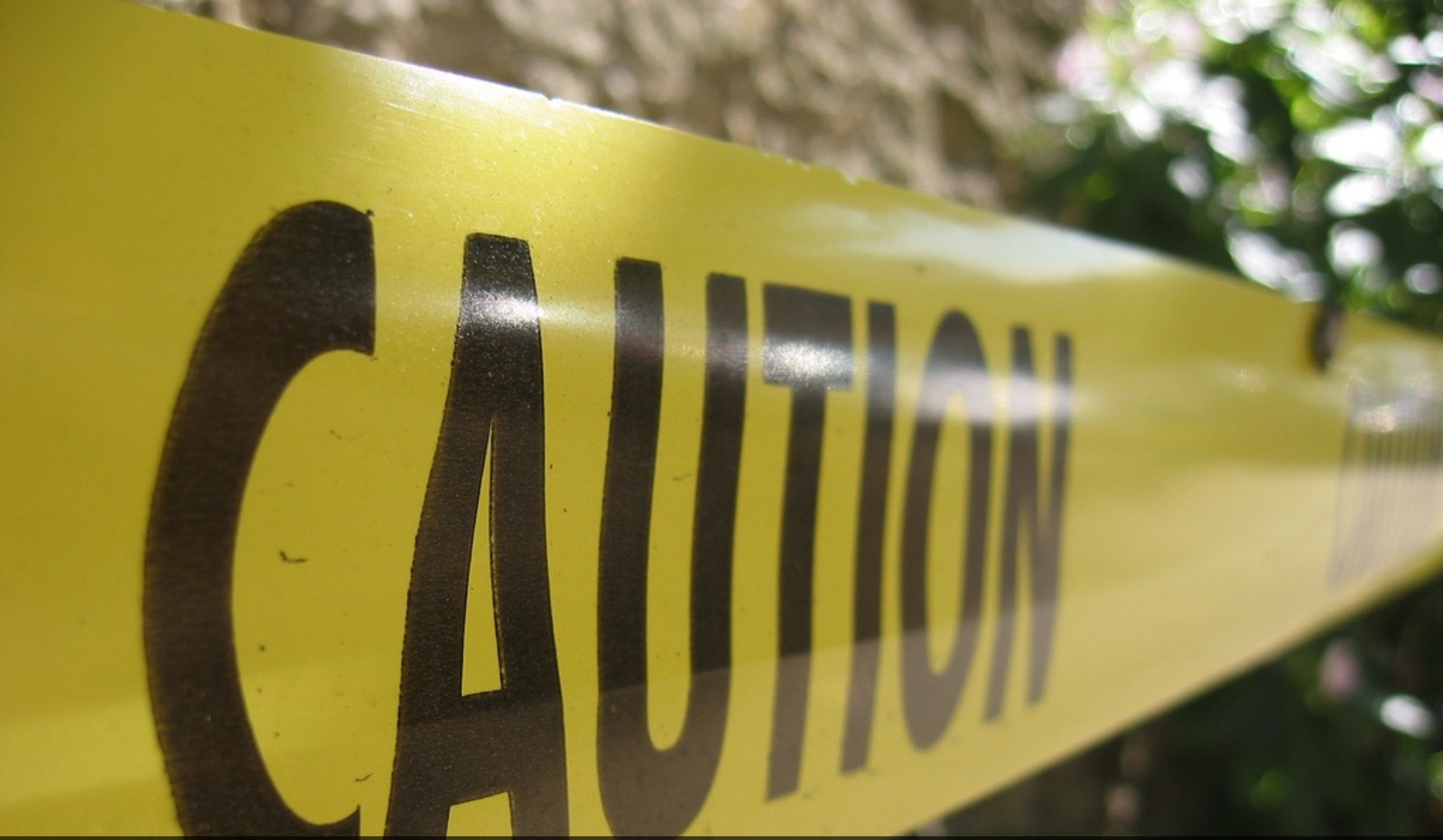
arrow_function_4.js — javascriptmasterclass

JS arrow_function_4.js

```
1   const sum = (a, b) => a + b;
2   const subtract = (a, b) => a - b;
3   const calculator = fn => (a, b) => fn(a, b);
4   console.log(calculator(sum)(2, 2));
5   console.log(calculator(subtract)(2, 2));
6
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_4.js
4
0
rodrigobranas:javascriptmasterclass $
```



As arrow functions não possuem as suas próprias variáveis this e arguments

arrow_function_5.js — javascriptmasterclass

JS arrow_function_5.js

```
1   const person = {
2       name: "James Gosling",
3       city: "Alberta",
4       year: 1955,
5       getAge: function() {
6           return (new Date()).getFullYear() - this.year;
7       }
8   };
9   console.log(person);
10  console.log(person.getAge());
11
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_5.js
{ name: 'James Gosling',
  city: 'Alberta',
  year: 1955,
  getAge: [Function: getAge] }
63
rodrigobranas:javascriptmasterclass $
```

arrow_function_6.js — javascriptmasterclass

JS arrow_function_6.js

```
1   const person = {
2       name: "James Gosling",
3       city: "Alberta",
4       year: 1955,
5       getAge: () => {
6           return (new Date()).getFullYear() - this.year;
7       }
8   };
9   console.log(person);
10  console.log(person.getAge());
11
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_6.js
{ name: 'James Gosling',
  city: 'Alberta',
  year: 1955,
  getAge: [Function: getAge] }
NaN
rodrigobranas:javascriptmasterclass $
```

arrow_function_7.js — javascriptmasterclass

JS arrow_function_7.js

```
1   const sum = function() {
2       let total = 0;
3       for(let argument in arguments) {
4           total += arguments[argument];
5       }
6       return total;
7   };
8   console.log(sum(1,2,3,4,5,6,7,8,9));
9
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_7.js
45
rodrigobranas:javascriptmasterclass $
```

arrow_function_8.js — javascriptmasterclass

JS arrow_function_8.js

```
1   const sum = () => {
2       let total = 0;
3       for(let argument in arguments) {
4           total += arguments[argument];
5       }
6       return total;
7   };
8   console.log(sum(1,2,3,4,5,6,7,8,9));
9
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_8.js
0[object Object]function require(path) {
    try {
      exports.requireDepth += 1;
      return mod.require(path);
    } finally {
      exports.requireDepth -= 1;
    }
  }[object Object]/Users/rodrigobranas/devel
opment/workspace/javascriptmasterclass/arrow
_function/arrow_function_8.js/Users/rodrigob
ranas/development/workspace/javascriptmaster
class/arrow_function
rodrigobranas:javascriptmasterclass $
```

E se um objeto for retornado?

arrow_function_9.js — javascriptmasterclass

JS arrow_function_9.js

```
1   const createPerson = function(name, city, year) {
2       return {
3           name,
4           city,
5           year
6       };
7   };
8   const person = createPerson("Alan Kay", "Springfield", 1940)
9   console.log(person);
10
```

TERMINAL

1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_9.js
{ name: 'Alan Kay', city: 'Springfield', yea
r: 1940 }
rodrigobranas:javascriptmasterclass $
```

arrow_function_10.js — javascriptmasterclass

JS arrow_function_10.js

```
1   const createPerson = (name, city, year) => {name, city, year
2   const person = createPerson("Alan Kay", "Springfield", 1940)
3   console.log(person);
4
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_10.js
undefined
rodrigobranas:javascriptmasterclass $
```

arrow_function_11.js — javascriptmasterclass

JS arrow_function_11.js

```
1   const createPerson = (name, city, year) => ({name, city, yea
2   const person = createPerson("Alan Kay", "Springfield", 1940)
3   console.log(person);
4
```

TERMINAL 1: bash

```
rodrigobranas:javascriptmasterclass $ node a
rrow_function/arrow_function_11.js
{ name: 'Alan Kay', city: 'Springfield', yea
r: 1940 }
rodrigobranas:javascriptmasterclass $
```