Say Hello to my Little Friend:

JS

About

Carla Morawicz

McGill

Day Job

Todos

“*JavaScript is* the only language that I’m aware of that people feel they don’t need to learn before they start using it.”

- *Douglas Crockford*

#sotrue

OBJECTS
SPACE RANGER
LIGHTYEAR
OBJECTS EVERYWHERE

you know what else are objects...

FUNCTIONS.

#firstclass

FUNCTION DECLARATION

```
function foo (arg1, arg2,...) {
    //do something!
}
```

VS

FUNCTION EXPRESSION

```
var foo = function (arg1, arg2,...) {
    //do something!
};
```

The same!

But not the same!?!

Consider the following...

```
alert(sum(10, 20));    //30

function sum(num1, num2) {
    return num1 + num2;
}
```

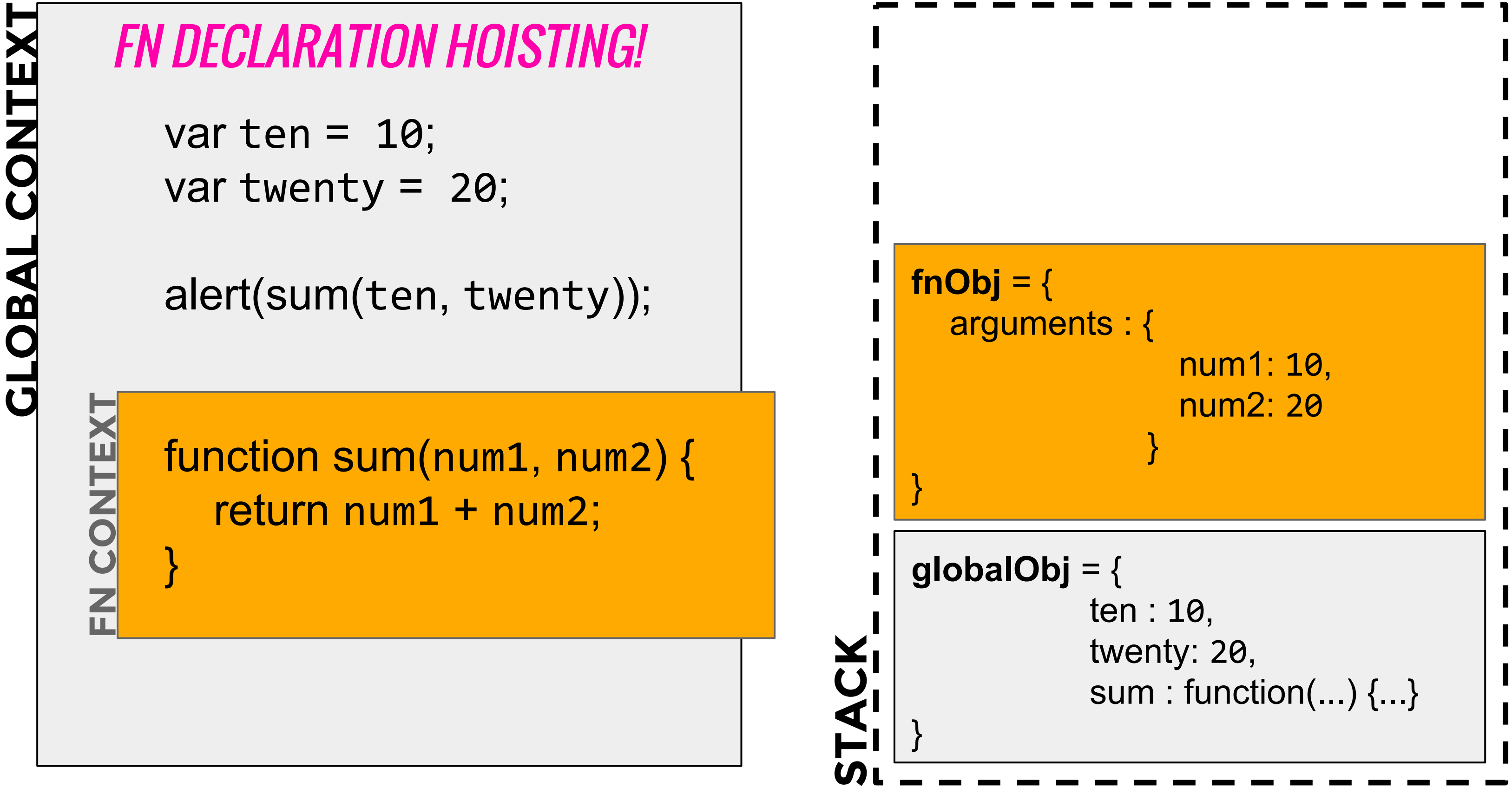
```
alert(sum(10, 20));    //throws error!

var sum = function(num1, num2) {
    return num1 + num2;
};
```

ERRONEOUS!

Context is Everything!

EXECUTION CONTEXT // SCOPE



FN DECLARATION HOISTING!

```
var ten =  10;
var twenty =  20;

alert(sum(ten, twenty));

function sum(num1, num2) {
    return num1 + num2;
}
```

```
fnObj = {
    arguments : {
                        num1: 10,
                        num2: 20
                    }
}
```

```
globalObj = {
                ten : 10,
                twenty: 20,
                sum : function(...) {...}
}
```

“JavaScript does not have *block scope*
(but has closure so let’s call it even?)”

- Eugene Yokota

#seemsfair

NO BLOCK SCOPE // CLOSURE

GLOBAL CONTEXT

```
var ten =  0;

for (var i=0; i < 10; i++) {
      ten++;
}

alert(i);   //10, i lives!!!

alert(sum(ten)());
```

FN CONTEXT

```
function sum(num) {
   var twenty =  20;
   return function () {
     return  num + twenty;
}}
```

STACK

```
anonymObj = {
   arguments : { },
   num : 10,
   twenty: 20
}
```

```
fnObj = {
   arguments : { num: 10 },
   twenty: 20
}
```

```
globalObj = {
            ten : 10,
            i = 10,
            sum : function(...) {...}
}
```

FUNCTIONS as VALUES

"anything var can do, i can do <u>better!</u>"

```
function callFunction(someFunction, someArgument) {
    return someFunction(someArgument);
}
```

Thank you!