



Front-end development: Tricky Parts

JavaScript – scope & hoisting

The following exercise contains the following subjects:

- scope
- hoisting

Instructions

Without running the code below, explain in your own words what the result of each block of code will be and why.

If there are any faulty outputs, please write on how we can fix them.

Block 1

```
function funcA() {  
  console.log(a);  
  console.log(foo());  
  var a = 1;  
  function foo() {  
    return 2;  
  }  
}  
funcA();
```

Block 2

```
var fullName = 'John Doe';
var obj = {
  fullName: 'Colin Ihrig',
  prop: {
    fullName: 'Aurelio De Rosa',
    getFullName: function () {
      return this.fullName;
    }
  }
};
console.log(obj.prop.getFullName());

var test = obj.prop.getFullName;
console.log(test());
```

Block 3

```
function funcB() {
  let a = b = 0;
  a++;
  return a;
}
funcB();
console.log(typeof a);
console.log(typeof b);
```

Block 4

```
function funcC() {  
    console.log("1");  
}  
funcC();  
function funcC() {  
    console.log("2");  
}  
funcC();
```

Block 5

```
function funcD1() {  
    d = 1;  
}  
funcD1();  
console.log(d);  
  
function funcD2() {  
    var e = 1;  
}  
funcD2();  
console.log(e);
```

Block 6

```
function funcE() {  
    console.log("Value of f in local scope: ", f);  
}  
console.log("Value of f in global scope: ", f);  
var f = 1;  
funcE();
```