Front-end development: Tricky Parts
JavaScript – closures

The following exercise contains the following subjects:

- closures

Instructions

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

Block 1

```javascript
(function immediateA(a) {
  return (function immediateB(b) {
    console.log(a);
  })(1);
})(0);
```

Block 2

```
let count = 0;
(function immediate() {
  if (count === 0) {
    let count = 1;
    console.log(count);
  }
  console.log(count);
})();
```

Block 3

```
var module = (function () {
  var foo = 'foo'
  function addToFoo (bam) {
    foo = bam;
    return foo;
  }
  var publicInterface = {
    bar: function () {
      return 'bar';
    },
  bam: function () {
      return addToFoo('bam')
    }
  }
  return publicInterface
})()
console.log(module.bar())
console.log(module.bam())
```

Block 4

```javascript
function createIncrement() {
  let count = 0;
  function increment() {
    count++;
  }
  let message = `Count is ${count}`;
  function log() {
    console.log(message);
  }
  return [increment, log];
}
const [increment, log] = createIncrement();
increment();
increment();
increment();
log();
```

Block 5

```javascript
var x = 10;
console.log(x);
if (true) {
  (function() {
    var x = 20;
    console.log(x);
  })();
}
console.log(x);
```

Block 6

```javascript
var x = 10;
console.log(x);
function test(){
  var x = 20;
  console.log(x);
  if (x > 10) {
    let x = 30;
    console.log(x);
  }
  console.log(x);
}
test();
console.log(x);
```

Block 7

```javascript
var x;
x = 10;
function test(){
  var x;
  if (x > 20) {
    x = 50;
  }
  console.log(x);
}
test();
```

Block 8

```javascript
function test(){
  var x, y;
  if (false) {
    x = 50;
  }
  console.log(x);
  console.log(y);
  y = 100;
  console.log(y);
}
test();
```

Block 9

```javascript
function test(){
  foo();
  bar();
  function foo(){
    console.log('foo');
  }
  var bar = function() {
    console.log('bar');
  }
}
test();
```