

Guess the output:

```
let x = 10;
{
  let x = 20;
  console.log(x);
}
console.log(x);
```

20

10

```
const obj = { name: "Alice" };
obj.name = "Bob";
console.log(obj.name);
```

Bob

```
const x = 10;
x = 20;
console.log(x);
```

TypeError

Error :constant cannot be reassigned

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

20

20

```
let x = 10;
if (true) {
  let x = 20;
  console.log(x);
}
console.log(x);
```

20

10

```
function greet() {
  console.log(this.name);
}
const person = { name: "Bob" };
const boundGreet = greet.bind(person);
boundGreet();
```

Bob “person’s object”

```
function greet(a, b) {
  console.log(this.name, a, b);
}
```

Bob Hello World

```
const person = { name: "Bob" };
greet.apply(person, ["Hello", "World"]);
```

```
function greet() {
  console.log(this.name);    Bob
}
const person = { name: "Bob" };
greet.call(person);
```

```
function greet() {
  console.log(this.name);    Bob
}
const person = { name: "Bob" };
greet.call(person);
```

```
const obj = {
  name: "Alice",
  greet() {
    setTimeout(() => {
      console.log(this.name);    Alice
    }, 1000);
  },
};
obj.greet();
```

```
const obj = {
  name: "Alice",
  greet() {
    setTimeout(function () {
      console.log(this.name);    //undefined as it's callback function and anonymous function
                                   // this refers to the global object
    }, 1000);
  },
};
obj.greet();
```

```
const obj = {
  name: "Alice",
  greet() {
    console.log(this.name);    Alice
  },
};
obj.greet();
```

```
"use strict";
function myFunction() {
  console.log(this);    Undefined
}
```

```
myFunction();
```

```
function myFunction() {  
  console.log(this);    Object "global"  
}  
myFunction();
```