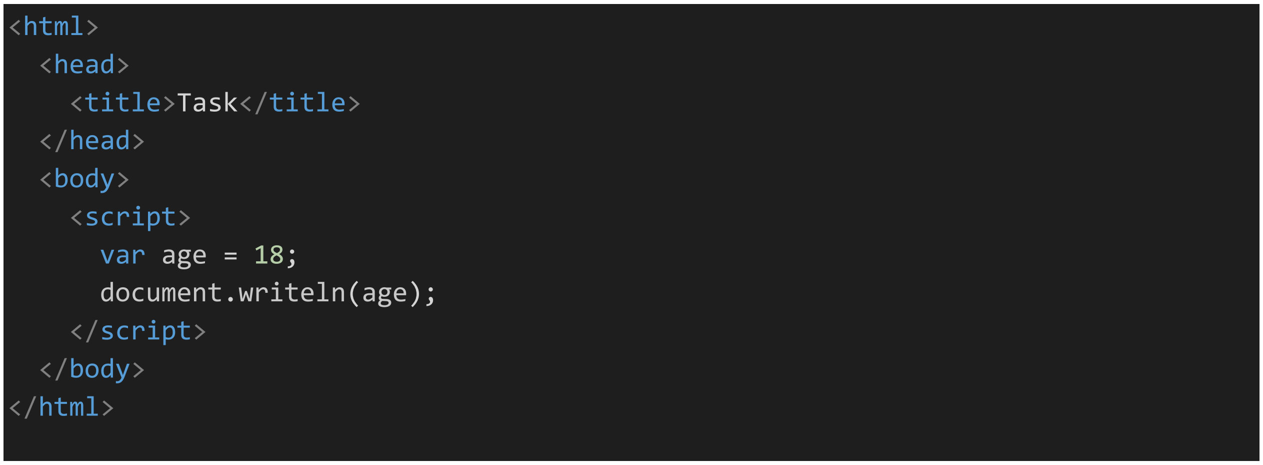
The modern mode, “use strict”, Variables

1. The modern mode, “use strict”:

Task 11

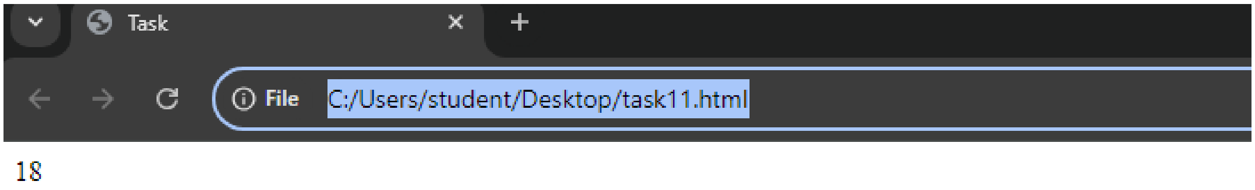
Write a script without using “use strict” and try to assign a value to an undeclared variable. Note the result.

CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      var age = 18;
      document.writeln(age);
    </script>
  </body>
</html>
```

OUTPUT:

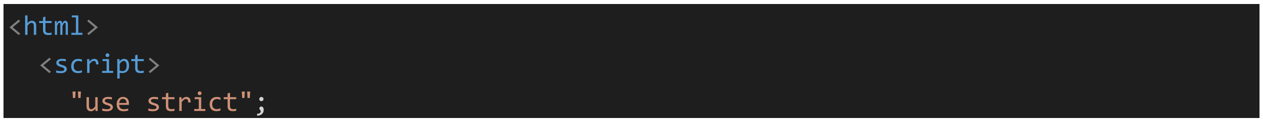


```
18
```

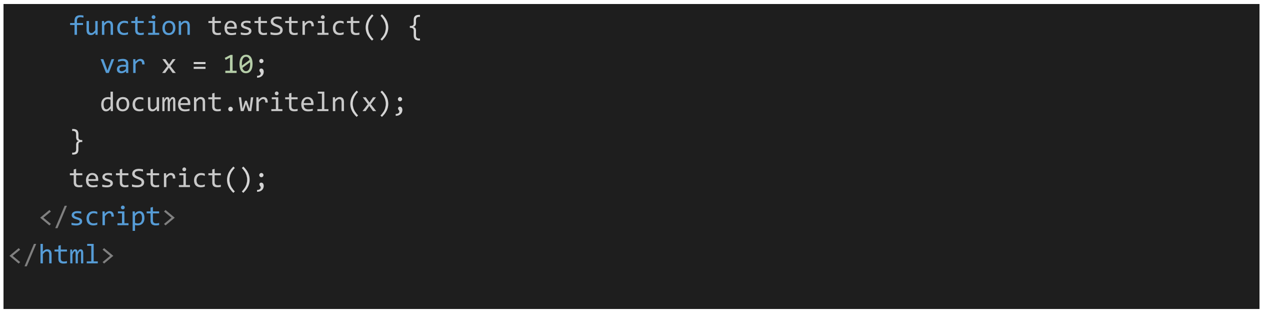
Task 12

Enable “use strict” mode and repeat the above action, noting the difference.

CODE:

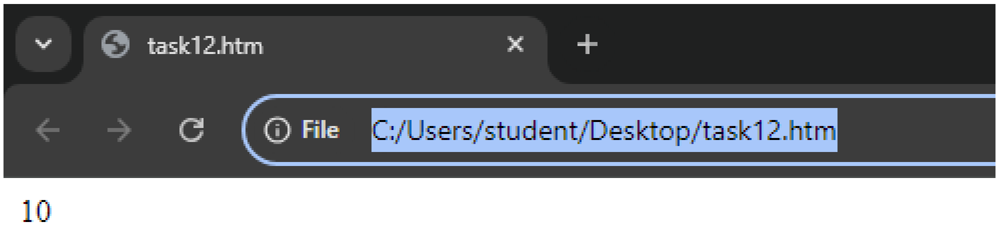


```
<html>
  <script>
    "use strict";
```



```
    function testStrict() {
      var x = 10;
      document.writeln(x);
    }
    testStrict();
  </script>
</html>
```

OUTPUT:

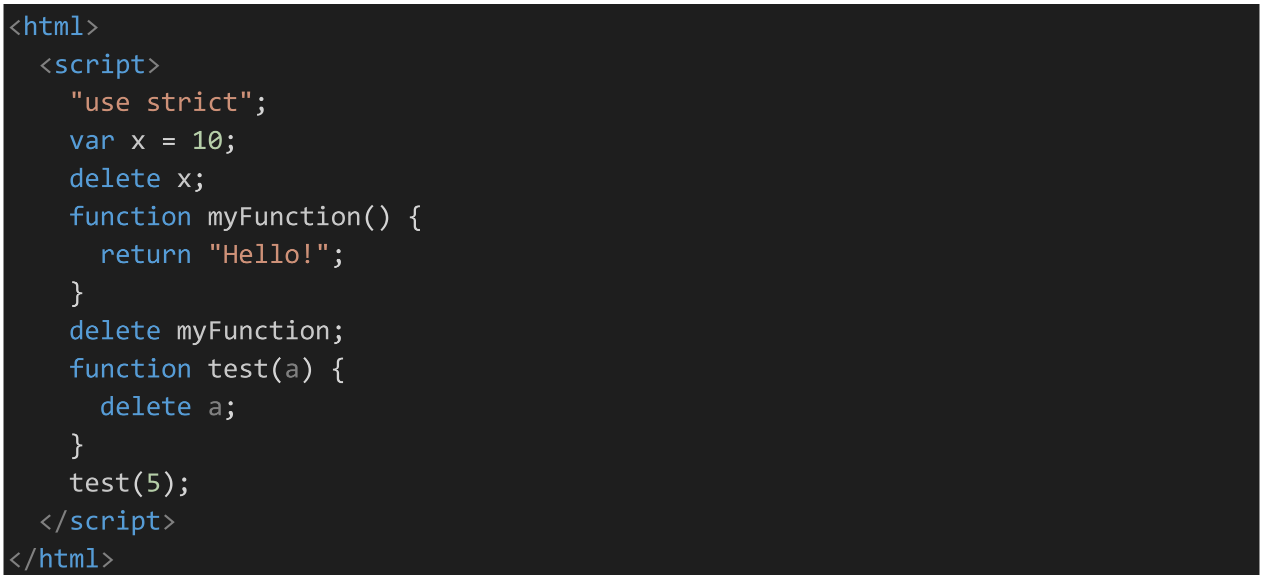


```
10
```

Task 13

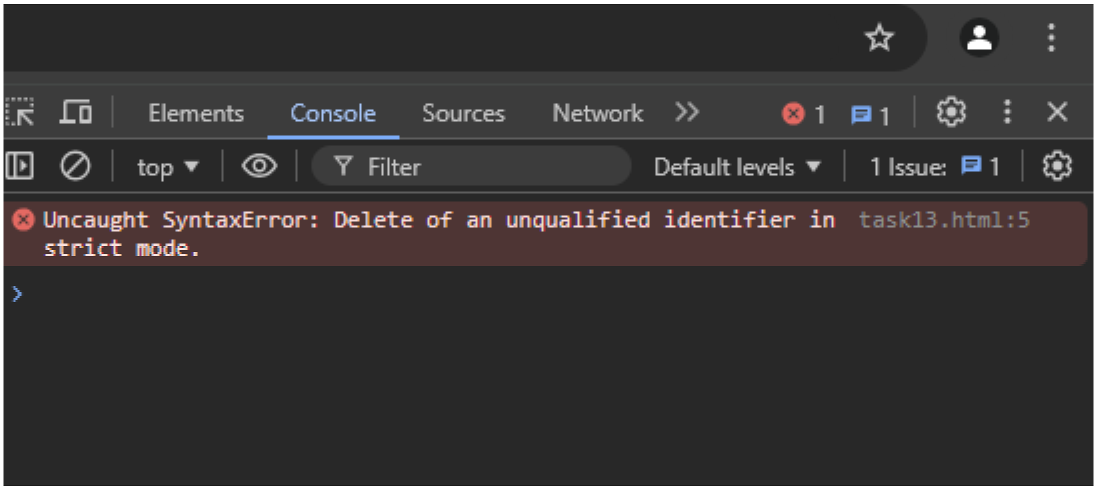
In "use strict" mode, try to delete a variable, function, or function parameter.

CODE:



```
<html>
  <script>
    "use strict";
    var x = 10;
    delete x;
    function myFunction() {
      return "Hello!";
    }
    delete myFunction;
    function test(a) {
      delete a;
    }
    test(5);
  </script>
</html>
```

OUTPUT:



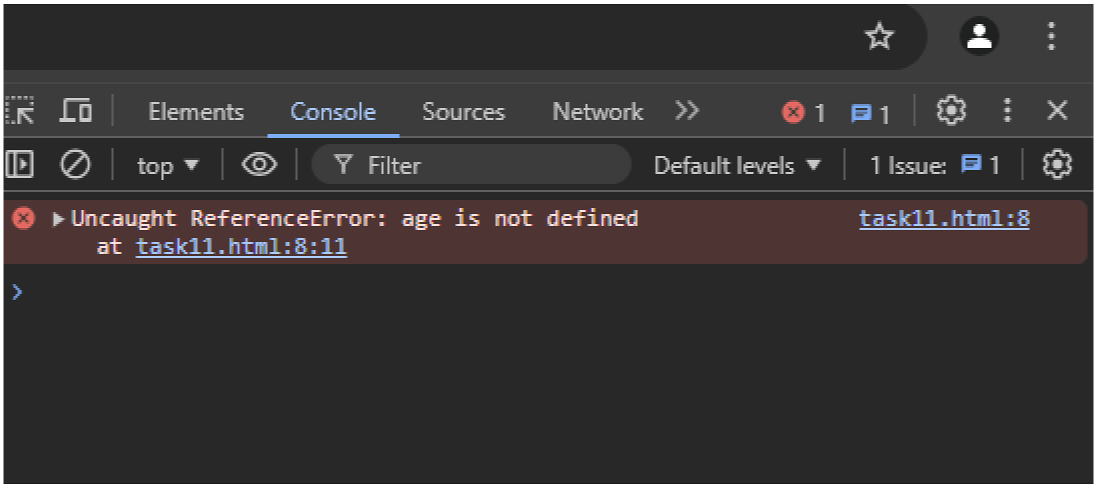
```
Uncaught SyntaxError: Delete of an unqualified identifier in    task13.html:5
strict mode.
```

Task 14: Assign a value to an undeclared variable without “use strict” and then with “use strict”.

CODE:

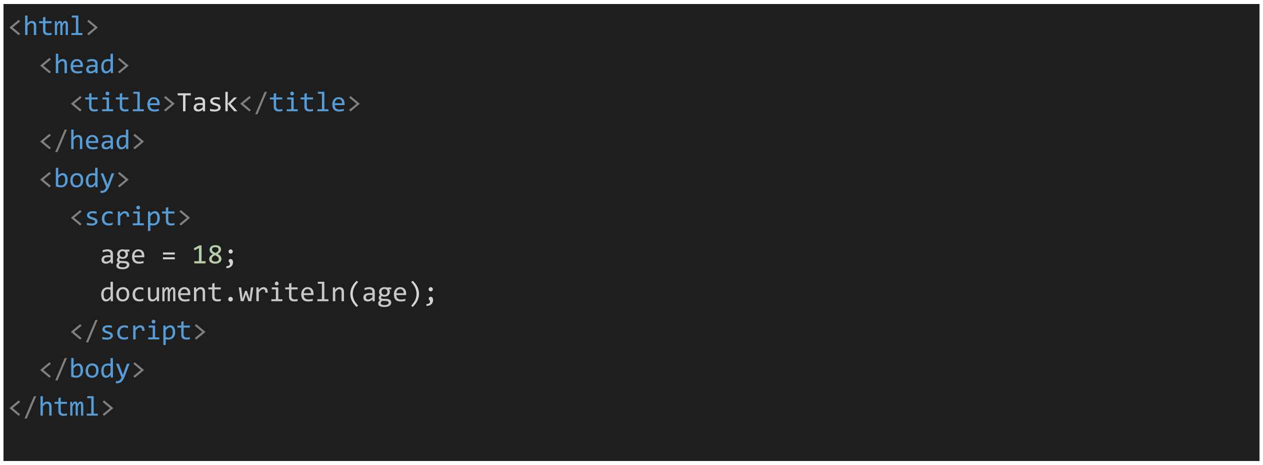
```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
        "use strict";
      age = 18;
      document.writeln(age);
    </script>
  </body>
</html>
```

OUTPUT:



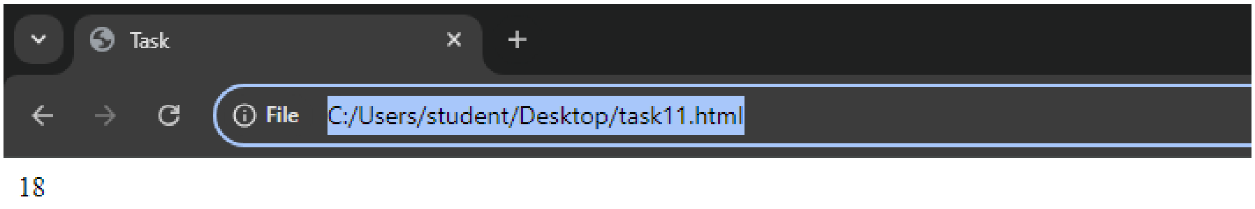
```
Uncaught ReferenceError: age is not defined        task11.html:8
    at task11.html:8:11
```

CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
      age = 18;
      document.writeln(age);
    </script>
  </body>
</html>
```

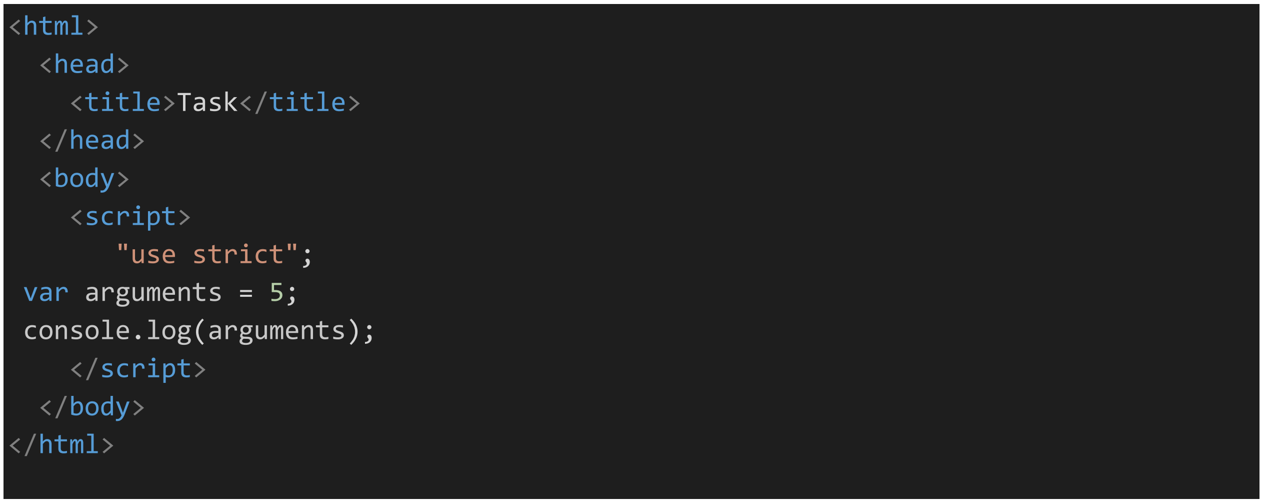
OUTPUT:



18

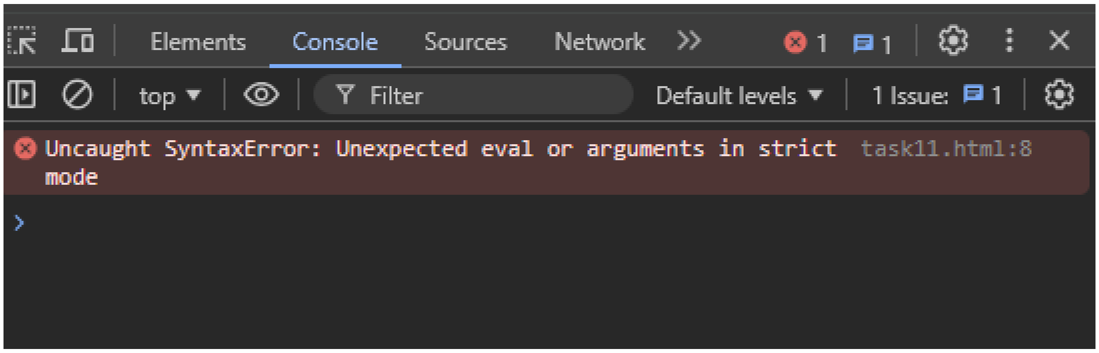
Task 15: Declare a variable with a reserved keyword in “use strict” mode.

CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
       "use strict";
 var arguments = 5;
 console.log(arguments);
    </script>
  </body>
</html>
```

OUTPUT:



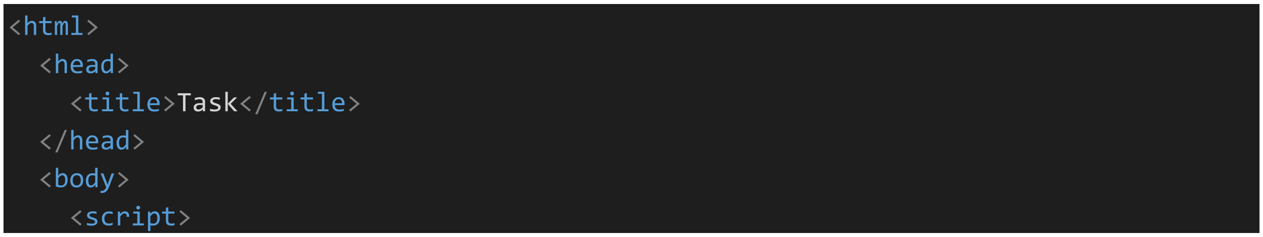
```
Uncaught SyntaxError: Unexpected eval or arguments in strict   task11.html:8
mode
>
```

2. Variables:

Task 16

Declare variables using let, const, and var. Discuss when each should be used.

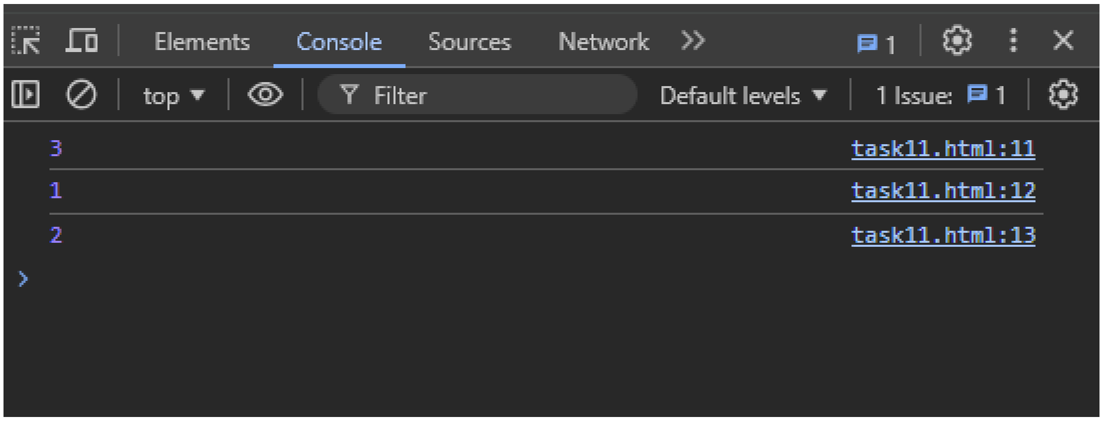
CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
```

```
      let x = 1;
      if (true) {
        const y = 2;
        var z = 3;
        console.log(z);
      console.log(x);
      console.log(y);
      }
    </script>
  </body>
</html>
```

OUTPUT:



```
3                                                    task11.html:11
1                                                    task11.html:12
2                                                    task11.html:13
>
```

Task 17

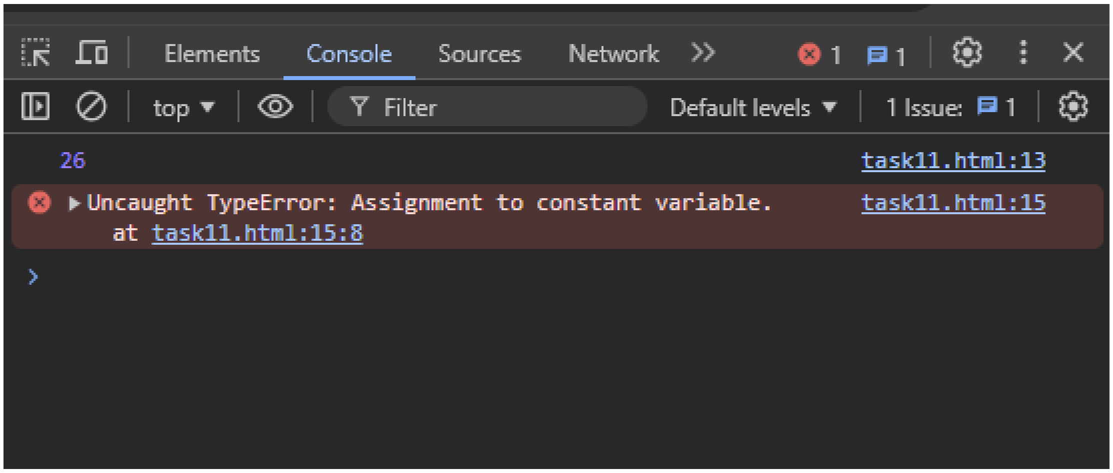
Attempt to reassign a const variable and observe the result.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
     const person = {
  name: 'Alice',
  age: 25
};

person.age = 26;
```

```
console.log(person.age);

person = { name: 'Bob', age: 30 };

    </script>
  </body>
</html>
```

OUTPUT:



```
26                                                          task11.html:13
▶Uncaught TypeError: Assignment to constant variable.       task11.html:15
    at task11.html:15:8
>
```

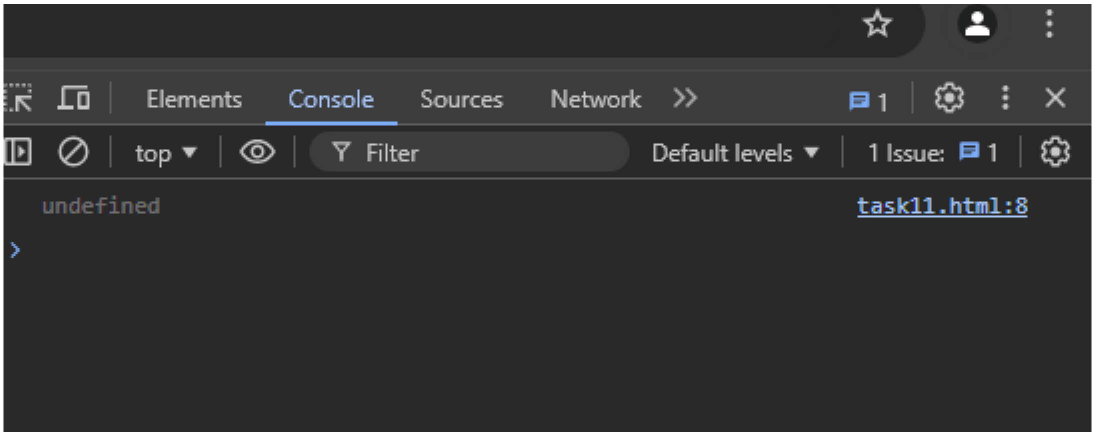
Task 18

Declare a variable without initializing it and print its value.

CODE:

```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
     let a;
console.log(a);

    </script>
  </body>
</html>
```

OUTPUT:

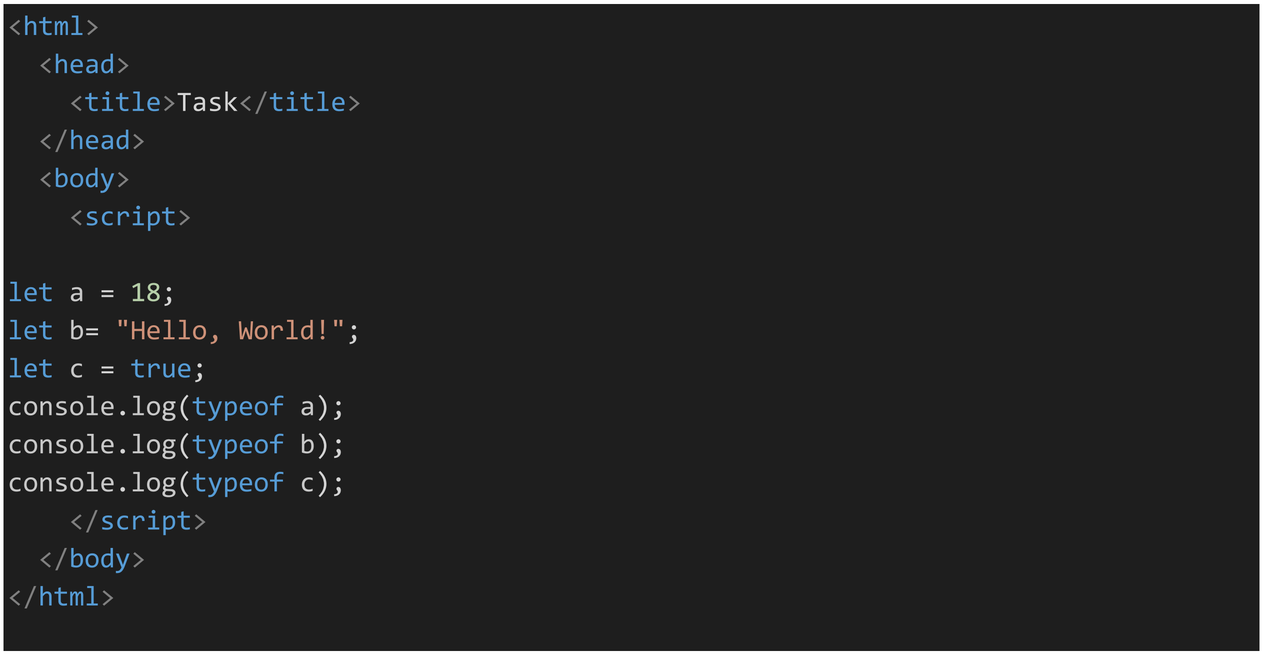


```
undefined                                                    task11.html:8
```

Task 19

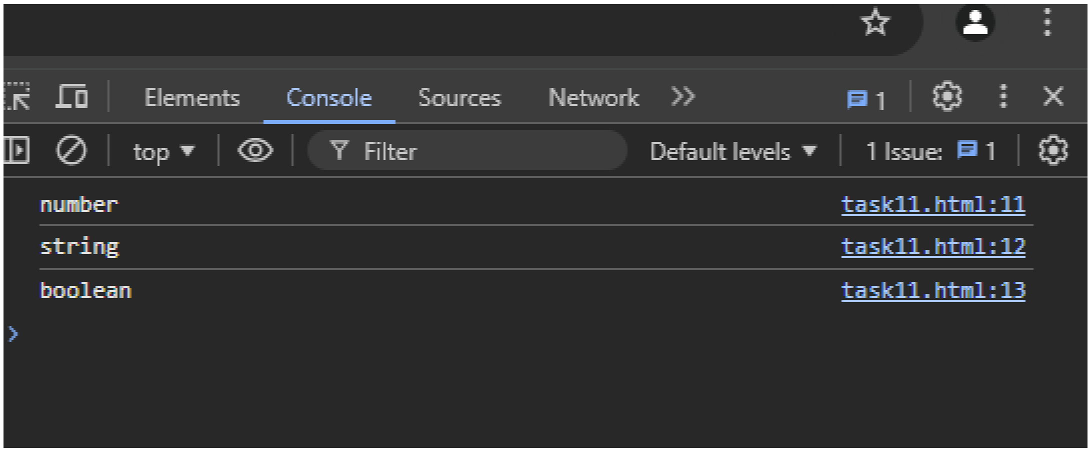
Assign a number, string, and boolean value to a variable and print its type using typeof.

CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>

let a = 18;
let b= "Hello, World!";
let c = true;
console.log(typeof a);
console.log(typeof b);
console.log(typeof c);
    </script>
  </body>
</html>
```

OUTPUT:

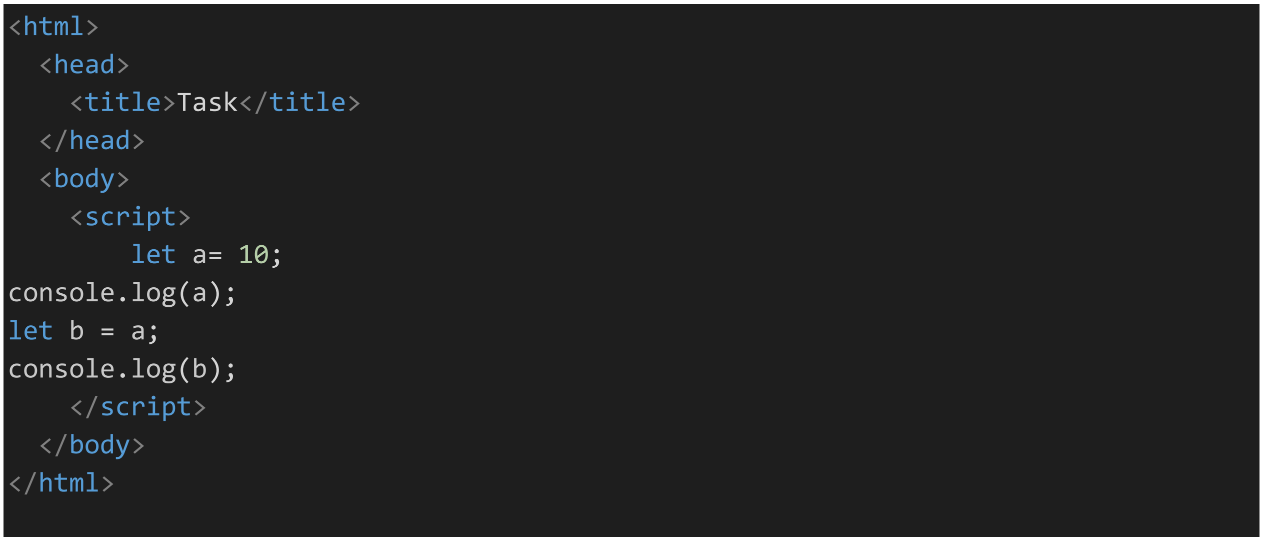


```
number                                                    task11.html:11
string                                                    task11.html:12
boolean                                                   task11.html:13
>
```

Task 20

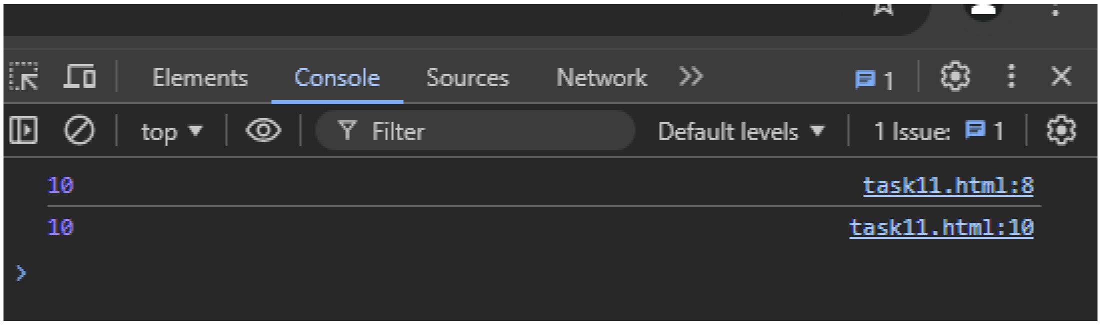
Rename a variable and observe the outcome.

CODE:



```
<html>
  <head>
    <title>Task</title>
  </head>
  <body>
    <script>
        let a= 10;
console.log(a);
let b = a;
console.log(b);
    </script>
  </body>
</html>
```

OUTPUT:



```
10                                                        task11.html:8
10                                                        task11.html:10
>
```