

```
let tern = x=> isNaN(x) ? ( x!==x ? 'NaN' : (undefined===x ? 'undefined' :
(isNaN(x) ? 'NaN' : 'qqq' ) ) ) : (x === 0? 'number' : ( isNaN(parseInt(x)) ?
'NaN' : 'number' ) );
```

```
module.exports = tern;
```

```
$ node test1.js
TAP version 13
# ternary function
not ok 1 tern() should return NaN
```

```
---
operator: equal
expected: 'NaN'
actual:   'string'
at: Test.assert (/Users/eliasgoss/Dropbox/p_r_o_j_e_c_t_s/js_testing/unit/test1.js:11:12)
...
```

```
1..1
# tests 1
# pass 0
# fail 1
```

```
→ eliasgoss@ ~/Dropbox/p_r_o_j_e_c_t_s/js_testing/unit
```

```
var test = require('tape'),
    tern = require('./tern');
```

```
test('ternary function', assert => {
  const actual = typeof tern(''),
        expected = 'NaN';
```

```
  assert.equal(actual, expected, 'tern() should return NaN');
```

```
  //провал – почему?
```

```
  //потому что мы ожидаем NaN по смыслу
```

```
  //а функция tern возвращает строку по факту
```

```
  //т.е. мы видим в консоли NaN, но это – строка, а не значение NaN
```

```
  assert.end();
```

```
});
```

UNIT

Next time you write a test, remember to answer all the questions:

What are you testing?

What should it do?

What is the actual output?

What is the expected output?

How can the test be reproduced?