

Qunit

```
Qunit.test( "      ", function() {  
    assert (function (), "      ");  
}):
```

Assert

Example Test Code

```
function add(a, b){  
    return a+b;  
}
```

ok()

```
<sxh js> Qunit.test("ok_test", function(){
```

```
    ok(add(1,2) == 3, "true"); //      Logic      true      가  
    ok(add(1,2) == 4, "false");
```

```
}); </sxh>
```

equal()

```
<sxh js> Qunit.test("equal_test", function(){
```

```
    equal(add(1,1), 2, "true"); //      Logic      value      .  
    equal(add(1,1), 3, "false");
```

```
}); </sxh>
```

deepEqual()

```
<sxh js> Qunit.test("deepequal/test", function(){
```

```
deepEqual(add(1,1),2,"true"); // Logic value type
deepEqual(add(1,1),"2","false");

}); </sxh>
```

strictEqual()

```
<sxh js> Qunit.test("strictequal/test", function(){
    strictEqual(add(1,1), 2, "true"); //value type
    strictEqual(add(1,1), "2", "false");

}); </sxh>
```

expect()

```
<sxh js> Qunit.test("expect_test", function(){
    expect(4); //assert가 가 ,
    ok(add(1,2) == 3, "true");
    equal(add(1,1), 2, "ture");
    deepEqual(add(1,1),"2","false");
    strictEqual(add(1,1), "2", "false");

}); </sxh>
```

module()

```
<sxh js> module("Group A"); Qunit.test("oktest", function(){ ok(add(1,2) == 3,
"true"); }); Qunit.test("equaltest", function(){ equal(add(1,1), 2, "true"); }); module("Group B");
Qunit.test("deepequaltest", function(){ deepEqual(add(1,1),"2","false"); });
Qunit.test("strictequaltest", function(){ strictEqual(add(1,1), "2", "false"); }); </sxh> — ##
```

Plugin Backlinks:

From:
<https://moro.kr/> - **Various Ways**

Permanent link:
<https://moro.kr/open/qunit>

Last update: **2020/06/02 09:25**

