

Esercizi con test unitari

Abbiamo preparato degli esercizi organizzati a difficoltà crescente per permettervi di verificare in autonomia la vostra preparazione durante il corso.

[Disclaimer](#)

[Come utilizzare gli esercizi](#)

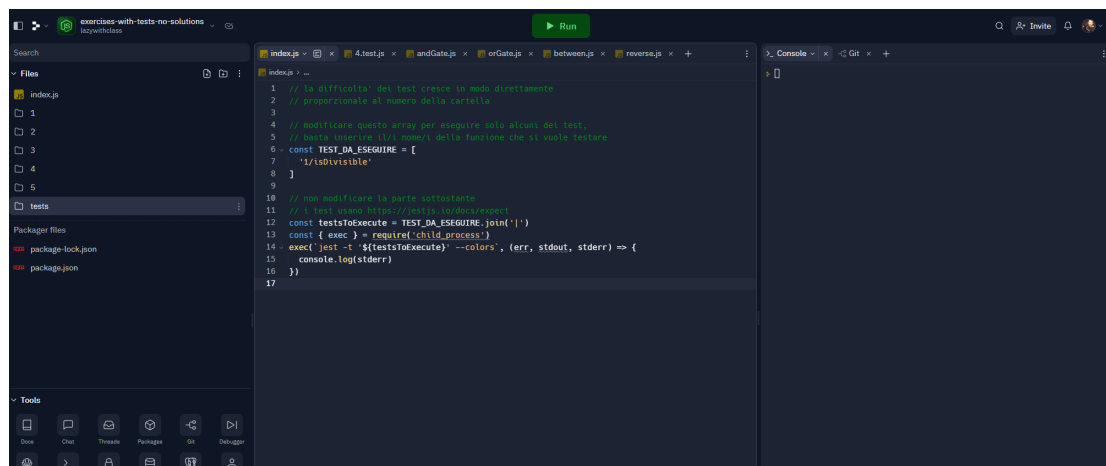
[Aggiornare gli esercizi](#)

Disclaimer

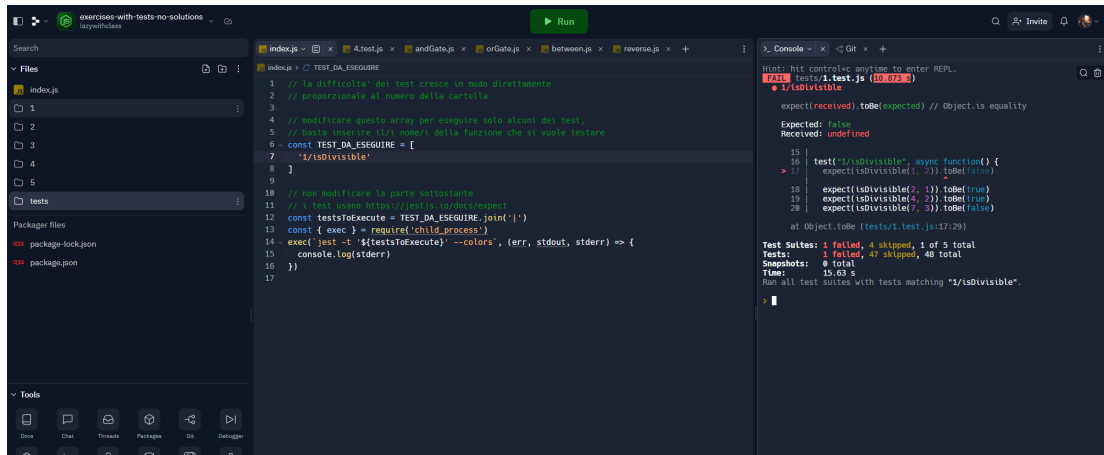
E' molto facile trovare online aiuto per risolvere i seguenti esercizi senza sforzo, vi suggeriamo di farli quanto più possibile in autonomia.

Come utilizzare gli esercizi

1. registrarsi su <https://replit.com>
2. andare a <https://replit.com/@lazywithclass/exercises-with-tests-no-solutions>
3. premere sul pulsante "Fork Repl", così da aggiungere al proprio account la repl con tutti gli esercizi
4. ora dovrete avere una situazione come nello screenshot: a sinistra l'elenco delle cartelle che contengono gli esercizi a difficoltà crescente e in centro il codice; la cartella tests contiene appunto i test unitari, non vi servira' modificare nulla li dentro



5. ora vi basta premere sul pulsante “Run” in verde (oppure premere CTRL e ENTER assieme) e verranno eseguiti i test relativi all’esercizio chiamato “1/isDivisible”



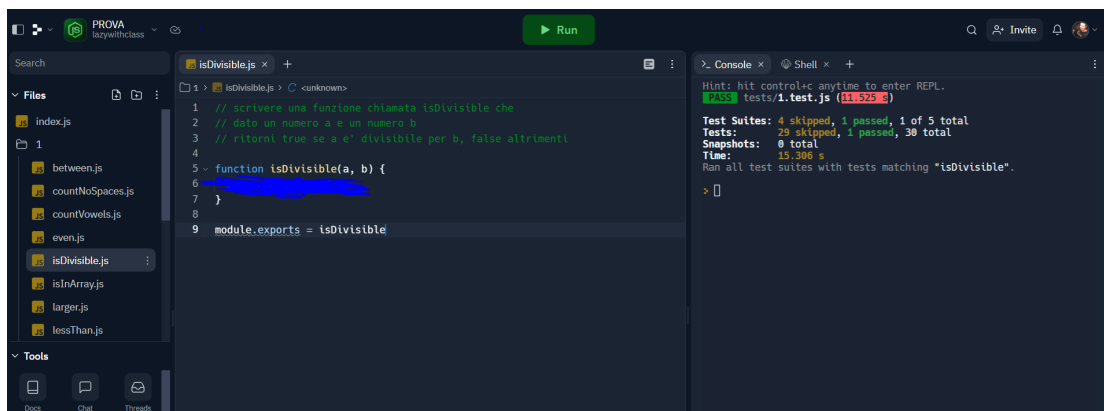
The screenshot shows the VS Code interface with a file explorer on the left, a code editor in the center, and a console on the right. The code editor displays a file named '1.test.js' with the following content:

```
1 // la definizione dei test viene in modo direttissimo
2 // proporzionata al numero della cartella
3
4 // modificare questa array per eseguire solo alcuni dei test
5 // deve essere il nome della funzione che si vuole testare
6 const TEST_DA_ESEGUIRE = [
7   '1/isDivisible'
8 ]
9
10 // non modificare la parte sottostante
11 // i test vanno https://jestjs.io/docs/en/expect
12 const testsToExecute = TEST_DA_ESEGUIRE.join(',')
13 const { exec } = require('child_process')
14 - exec(`jest -i "${testsToExecute}" --colors`, {err, stdout, stderr}) => {
15   console.log(stderr)
16 }
17
```

The console on the right shows the output of the test run, indicating a failure:

```
Hint: hit control+c anytime to enter REPL.
FAIL tests/1.test.js (10.673 s)
  ● 1/isDivisible
    expect(received).toBe(expected) // Object.is equality
    Expected: false
    Received: undefined
  15 |
  16 |   test('1/isDivisible', async function() {
  17 |     expect(isDivisible(1, 2)).toBe(false)
  18 |
  19 |     expect(isDivisible(2, 1)).toBe(true)
  20 |     expect(isDivisible(4, 2)).toBe(true)
  21 |     expect(isDivisible(7, 3)).toBe(false)
  22 |   })
  at Object.toBe (tests/1.test.js:17:29)
Test Suites: 1 failed, 4 skipped, 1 of 5 total
Tests:      1 failed, 47 skipped, 48 total
Snapshots:  0 total
Time:       15.63 s
Run all test suites with tests matching "1/isDivisible".
```

6. a questo punto siete pronti per programmare, implementate la funzione “isDivisible” che si trova nella cartella 1 e rieseguite i test, se tutto e’ andato a buon fine dovreste vedere “PASS” in verde sulla destra, che vuol dire che i test sono passati e la vostra implementazione e’ corretta.



The screenshot shows the VS Code interface with a file explorer on the left, a code editor in the center, and a console on the right. The code editor displays a file named 'isDivisible.js' with the following content:

```
1 // scrivere una funzione chiamata isDivisible che
2 // dato un numero a e un numero b
3 // ritorni true se a e' divisibile per b, false altrimenti
4
5 function isDivisible(a, b) {
6   // implementare la funzione qui
7 }
8
9 module.exports = isDivisible
```

The console on the right shows the output of the test run, indicating a success:

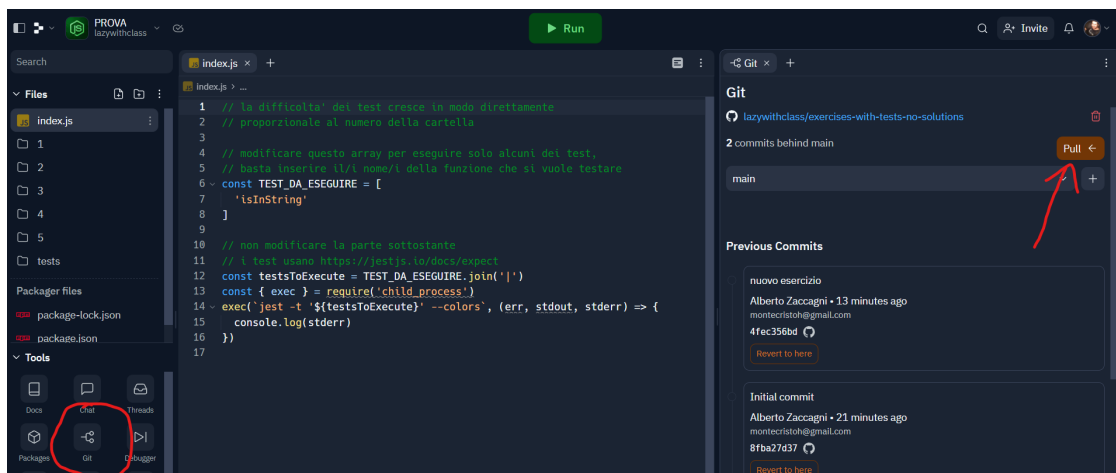
```
Hint: hit control+c anytime to enter REPL.
PASS tests/1.test.js (11.525 s)
Test Suites: 4 skipped, 1 passed, 1 of 5 total
Tests:      29 skipped, 1 passed, 30 total
Snapshots:  0 total
Time:       15.306 s
Run all test suites with tests matching "isDivisible".
```

7. se volete eseguire i test di altri esercizi vi bastera’ scrivere il nome delle relative funzioni nell’array TEST_DA_ESEGUIRE, se ad esempio volete eseguire isDivisible e alfabetoStrambo allora metterete [“1/isDivisible”, “2/alfabetoStrambo”]

Aggiornare gli esercizi

Ogni tot aggiungiamo nuovi esercizi con i loro test, potete aggiornare la vostra repl per ottenerli in modo molto semplice:

1. nella sezione “Tools” in basso a sinistra, cliccate su “Git”, vi apparirà una schermata come la seguente: se ci sono nuovi esercizi vedrete il pulsante “Pull” come da screenshot



2. e' possibile abbiate dei conflitti, nel qual caso e' bene scegliere di sovrascrivere la vostra repl con quella base, questo vuol dire che perderete le modifiche che avete fatto alla repl, ma non e' un problema se salvate i vostri esercizi in un progetto sul vostro PC