ES NEXT

**DEADLINE:** 01/04/2018

## FOLDER STRUCTURE

homework\_05/

└─ tower-of-babel/

└─ babel\_setup.js

└─ class.js

└─ class\_extend.js

└─ modules\_with\_name.js

└─ modules\_default\_export.js

└─ modules\_default\_export\_math.js

└─ solution.js

└─ computed\_property.js

└─ iterator\_for\_of.js

└─ generator.js

└─ destructure.js

└─ arrow\_function.js

└─ rest\_and\_spread.js

└─ learn-generators/

└─ catch\_error.js

└─ delegating\_generators.js

└─ generator\_iterator.js

└─ Look\_sync\_do\_async.js

└─ look\_sync\_make\_promise.js

└─ run\_stop\_run.js

└─ async-await/

└─ async\_task.js

## TASK 1

Follow instructions at <http://nodeschool.io/> to install and run **tower-of-babel.**

Follow exercises and verify that all your files pass corresponding tasks tests.

Pass **learn-generators** workshop.

## TASK 2

* create doAsyncMagic function using async/await that logs to console results of asyncFn() called 4 times asynchronously (see “CODE” section below);
* create iterateRange function that returns sum of rangeGen() yields using async iterators.

## RESTRICTIONS

* Block scope exercise fails one test if the file is named different than solution.js.

## CODE

const waitFewSec = (msec, triggerFail) => {

return new Promise((resolve, reject) => {

setTimeout(() => {

if (triggerFail) {

reject(false);

return;

}

resolve(true);

}, msec);

});

};

const asyncFn = async () => {

const result = await waitFewSec(1000);

return result;

};

/\* Your code here \*/

doAsyncMagic(); // [true, true, true, true]

async function\* rangeGen() {

for (let i = 1; i <= 15; i++) {

yield i;

}

}

/\* Your code here \*/

iterateRange(); // Promise {<resolved>: 120}

## BEFORE SUBMIT

* Verify that all your files pass corresponding tasks tests

## USEFUL LINKS

* <http://nodeschool.io/>
* <http://cmder.net/>

* <https://nodejs.org/en/>