Tools

**DEADLINE**: 17/04/2018

## FOLDER STRUCTURE

homework\_10/

└─ index.html

└─ app.js

## TASK

Implement pattern decorator to add validation functionality to the Input instances:

* You have basic Input class, with can have its subclasses like NumberInput or TextInput
* after you create Input class(or its subclasses) instance, decorate(add additional functionality) it with validation decorators. After applying at least on of them, “valid” property should be added to an instance, with correct boolean flag, based on current input value.
* write AddRequiredValidation decorator, which will set “valid” property to false, if you set the input value to an empty string or undefined
* write AddMaxLengthValidation decorator, which will set “valid” property to false, if you set the input value, with length bigger then maxLength parameter provided to the decorator
* write AddNumberValidation decorator, which will set “valid” property to false, if you set the input value that’s not of type “number”
* You can apply this decorators together to one input instance or separately to different, but applying next decorator should not break the effect of previous.
* Additional you can add validation messages to an object, dependent on what validation didn’t pass.
* Make at least on example of input with all three decorators applied.

## RESTRICTIONS

* It is restricted to use experimental ES2016 decorator feature like @decorator

## BEFORE SUBMIT

* Make at least on example of input with all three decorators applied, set different values to it and print “valid” property state to console to show correct results

## USEFUL LINKS

* <https://addyosmani.com/blog/decorator-pattern/>
* <https://www.joezimjs.com/javascript/javascript-design-patterns-decorator/>