

## Mongoose: Validation

So, in this section, you learned that:

- When defining a schema, you can set the type of a property to a SchemaType object. You use this object to define the validation requirements for the given property.

### // Adding validation

```
new mongoose.Schema({  
  name: { type: String, required: true }  
})
```

- Validation logic is executed by Mongoose prior to saving a document to the database. You can also trigger it manually by calling the **validate()** method.
- Built-in validators:
  - Strings: **minlength**, **maxlength**, **match**, **enum**
  - Numbers: **min**, **max**
  - Dates: **min**, **max**
  - All types: **required**

### // Custom validation

```
tags: [  
  type: Array,  
  validate: {  
    validator: function(v) { return v && v.length > 0; }, message:  
    'A course should have at least 1 tag.'  
  }  
]
```

- If you need to talk to a database or a remote service to perform the validation, you need to create an async validator:

```
validate: {  
  isAsync: true  
  validator: function(v, callback) {  
    // Do the validation, when the result is ready, call the callback  
    callback(isValid);  
  }  
}
```

- Other useful SchemaType properties:
  - Strings: **lowercase**, **uppercase**, **trim**
  - All types: **get**, **set** (to define a custom getter/setter)

```
price: {  
  type: Number,  
  get: v => Math.round(v),  
  set: v => Math.round(v)  
}
```