

# Unit Testing

So, in this section, you learned that:

- Automated testing is the practice of writing code to test our code.
- Automated tests help us deliver software with fewer bugs and of better quality. They also help us refactor our code with confidence.
- Jest is a new trending popular testing framework recommended by Facebook. It comes with everything you need to write automated tests.
- We have 3 types of automated tests:
  - **Unit tests:** test a unit of an application without external resources (eg db)
  - **Integration tests:** test the application with external resources.
  - **Functional or end-to-end tests:** test the application through its UI.
- Tests should not be too general nor too specific. If they're too general, they don't give you much confidence that your code works. If they're too specific, they become fragile and can break easily. As you write code, you have to spend extra unnecessary time to fix these broken tests.
- Mocking is replacing a real implementation of a function with a fake or mock function. It allows us to isolate our application code from its external resources.
- Popular Jest matcher functions:

## // Equality

```
expect(...).toBe();  
expect(...).toEqual();
```

## // Truthiness

```
expect(...).toBeDefined();  
expect(...).toBeNull();  
expect(...).toBeTruthy();
```

```
expect(...).toBeFalsy();
```

### // Numbers

```
expect(...).toBeGreaterThan();
```

```
expect(...).toBeGreaterThanOrEqual();
```

```
expect(...).toBeLessThan();
```

```
expect(...).toBeLessThanOrEqual();
```

### // Strings

```
expect(...).toMatch(/regularExp/);
```

### // Arrays

```
expect(...).toContain();
```

### // Objects

```
expect(...).toBe(); // check for the equality of object references
```

```
expect(...).toEqual(); // check for the equality of properties
```

```
expect(...).toMatchObject();
```

### // Exceptions

```
expect(() => { someCode }).toThrow();
```