Unit Testing ReactJS Apps (Jest & Enzyme)

Q1. What is the main purpose of Jest in React projects?  
 a) State management  
 b) Unit testing framework  
 c) CSS styling tool  
 d) Routing library  
 Correct answer is Unit testing framework – Option b

Q2. Which command is commonly used to run Jest tests?  
 a) npm run start  
 b) npm run test  
 c) npm run build  
 d) npm run lint  
 Correct answer is npm run test – Option b

Q3. Enzyme is primarily used for:  
 a) API requests  
 b) Component testing  
 c) Redux store management  
 d) Internationalization  
 Correct answer is Component testing – Option b

Q4. Which Jest function is used to group related test cases?  
 a) test()  
 b) expect()  
 c) describe()  
 d) it()  
 Correct answer is describe() – Option c

Q5. In Jest, which function is used to make assertions?  
 a) test()  
 b) expect()  
 c) render()  
 d) shallow()  
 Correct answer is expect() – Option b

Q6. What does the shallow() method from Enzyme do?  
 a) Fully renders the component tree  
 b) Renders only the given component without its children  
 c) Tests API endpoints  
 d) Creates Redux store mocks  
 Correct answer is Renders only the given component without its children – Option b

Q7. Which function is used to render a component with full DOM in Enzyme?  
 a) shallow()  
 b) render()  
 c) mount()  
 d) snapshot()  
 Correct answer is mount() – Option c

Q8. Which Jest matcher is used to check strict equality?  
 a) toBe()  
 b) toEqual()  
 c) toMatch()  
 d) toContain()  
 Correct answer is toBe() – Option a

Q9. What does Jest's toEqual() check for?  
 a) Reference equality  
 b) Deep equality of objects/arrays  
 c) String matching  
 d) Boolean comparison  
 Correct answer is Deep equality of objects/arrays – Option b

Q10. Which Enzyme method simulates user interactions?  
 a) simulate()  
 b) trigger()  
 c) act()  
 d) mockEvent()  
 Correct answer is simulate() – Option a

Q11. Jest's beforeEach() function is used to:  
 a) Run once before all tests  
 b) Run before every test case  
 c) Run once after all tests  
 d) Run only if tests fail  
 Correct answer is Run before every test case – Option b

Q12. Snapshot testing in Jest ensures that:  
 a) API responses match schema  
 b) Component output hasn’t unexpectedly changed  
 c) Redux store is updated  
 d) Database schema is consistent  
 Correct answer is Component output hasn’t unexpectedly changed – Option b

Q13. Which file extension is commonly used for Jest test files?  
 a) .jsx  
 b) .test.js  
 c) .snap.js  
 d) .mock.js  
 Correct answer is .test.js – Option b

Q14. Which Jest function is used to mock dependencies?  
 a) jest.fn()  
 b) jest.mock()  
 c) jest.spyOn()  
 d) All of the above  
 Correct answer is All of the above – Option d

Q15. What does enzyme-adapter-react-16 (or latest) do?  
 a) Adds Redux support  
 b) Bridges Enzyme with React version  
 c) Enables Jest matchers  
 d) Creates snapshots  
 Correct answer is Bridges Enzyme with React version – Option b

Q16. In Jest, which function is used to run a single test file?  
 a) npm test fileName.test.js  
 b) jest --runSingle  
 c) npm run build fileName  
 d) testOnly fileName  
 Correct answer is npm test fileName.test.js – Option a

Q17. Enzyme’s find() method is used to:  
 a) Locate a DOM node or component  
 b) Simulate user clicks  
 c) Create a snapshot  
 d) Mount the component  
 Correct answer is Locate a DOM node or component – Option a

Q18. Which Jest configuration file is commonly used?  
 a) jest.config.json  
 b) jestfile.json  
 c) test.config.js  
 d) enzyme.config.js  
 Correct answer is jest.config.json – Option a

Q19. Which Jest function is used to check if a function is called?  
 a) toBeCalled()  
 b) toHaveBeenCalled()  
 c) wasCalled()  
 d) fnCalled()  
 Correct answer is toHaveBeenCalled() – Option b

Q20. In Jest, which option enables coverage reports?  
 a) npm test --verbose  
 b) npm test --coverage  
 c) npm test --watch  
 d) npm test --report  
 Correct answer is npm test --coverage – Option b

E2E Testing using Cypress

Q21. Cypress is mainly used for:  
 a) Backend testing  
 b) API testing  
 c) End-to-End testing  
 d) Database testing  
 Correct answer is End-to-End testing – Option c

Q22. Cypress tests run directly inside the:  
 a) Browser  
 b) Server  
 c) Database  
 d) IDE  
 Correct answer is Browser – Option a

Q23. Which Cypress command is used to visit a page?  
 a) cy.load()  
 b) cy.goto()  
 c) cy.visit()  
 d) cy.open()  
 Correct answer is cy.visit() – Option c

Q24. Cypress test files are usually written in:  
 a) HTML  
 b) JavaScript/TypeScript  
 c) Python  
 d) C#  
 Correct answer is JavaScript/TypeScript – Option b

Q25. Cypress default folder for integration tests is:  
 a) /cypress/tests/  
 b) /cypress/integration/  
 c) /tests/integration/  
 d) /src/tests/  
 Correct answer is /cypress/integration/ – Option b

Q26. Cypress supports which type of testing?  
 a) Unit  
 b) Component  
 c) Integration  
 d) All of the above  
 Correct answer is All of the above – Option d

Q27. To get an element by ID in Cypress:  
 a) cy.getElement('#id')  
 b) cy.find('#id')  
 c) cy.get('#id')  
 d) cy.locate('#id')  
 Correct answer is cy.get('#id') – Option c

Q28. Which Cypress command is used to simulate a click?  
 a) cy.press()  
 b) cy.tap()  
 c) cy.click()  
 d) cy.triggerClick()  
 Correct answer is cy.click() – Option c

Q29. Cypress provides:  
 a) Real browser environment  
 b) Virtual DOM only  
 c) API mocks only  
 d) Only CLI support  
 Correct answer is Real browser environment – Option a

Q30. Cypress runs asynchronously but provides:  
 a) Callbacks  
 b) Automatic waiting  
 c) Promises only  
 d) Manual waits  
 Correct answer is Automatic waiting – Option b

Q31. Cypress command to type text into an input:  
 a) cy.enter('text')  
 b) cy.type('text')  
 c) cy.write('text')  
 d) cy.input('text')  
 Correct answer is cy.type('text') – Option b

Q32. Cypress command to assert visibility:  
 a) cy.get().visible()  
 b) cy.should('be.visible')  
 c) cy.expect('visible')  
 d) cy.assert('visible')  
 Correct answer is cy.should('be.visible') – Option b

Q33. Cypress supports test retries with:  
 a) Retry block  
 b) Cypress.config({ retries })  
 c) cy.retry()  
 d) cy.loop()  
 Correct answer is Cypress.config({ retries }) – Option b

Q34. Cypress dashboard is used for:  
 a) Real-time test results & analytics  
 b) Code compilation  
 c) Storing test code  
 d) Database queries  
 Correct answer is Real-time test results & analytics – Option a

Q35. Cypress automatically handles:  
 a) Async code & waits  
 b) SQL queries  
 c) Cloud deployments  
 d) Webpack builds  
 Correct answer is Async code & waits – Option a

Q31. Cypress command to type text into an input:  
 a) cy.enter('text')  
 b) cy.type('text')  
 c) cy.write('text')  
 d) cy.input('text')  
 Correct answer is cy.type('text') – Option b.

Q32. Cypress command to assert visibility:  
 a) cy.get().visible()  
 b) cy.should('be.visible')  
 c) cy.expect('visible')  
 d) cy.assert('visible')  
 Correct answer is cy.should('be.visible') – Option b.

Q33. Cypress supports test retries with:  
 a) Retry block  
 b) Cypress.config({ retries })  
 c) cy.retry()  
 d) cy.loop()  
 Correct answer is Cypress.config({ retries }) – Option b.

Q34. Cypress dashboard is used for:  
 a) Real-time test results & analytics  
 b) Code compilation  
 c) Storing test code  
 d) Database queries  
 Correct answer is Real-time test results & analytics – Option a.

Q35. Cypress automatically handles:  
 a) Async code & waits  
 b) SQL queries  
 c) Cloud deployments  
 d) Webpack builds  
 Correct answer is Async code & waits – Option a.

Q36. In Redux, the state must always be treated as:  
 a) Mutable  
 b) Immutable  
 c) Temporary  
 d) None of the above  
 Correct answer is Immutable – Option b.

Q37. Which function in Redux is used to combine multiple reducers into one?  
 a) createReducer  
 b) applyMiddleware  
 c) combineReducers  
 d) rootReducer  
 Correct answer is combineReducers – Option c.

Q38. Which Redux middleware is commonly used for handling asynchronous operations?  
 a) redux-logger  
 b) redux-thunk  
 c) redux-toolkit  
 d) redux-persist  
 Correct answer is redux-thunk – Option b.

Q39. In Redux, what does dispatch() do?  
 a) Updates the reducer directly  
 b) Sends an action to the store  
 c) Returns the current state  
 d) Initializes the store  
 Correct answer is Sends an action to the store – Option b.

Q40. What is the correct order of Redux data flow?  
 a) Store → Reducer → Action → UI  
 b) UI → Action → Reducer → Store → UI  
 c) Reducer → Store → Action → UI  
 d) Action → UI → Reducer → Store  
 Correct answer is UI → Action → Reducer → Store → UI – Option b.

Q41. When fetching data in Redux, where should the API call usually be placed?  
 a) Inside the reducer  
 b) Inside the action creator (with middleware)  
 c) Inside the store directly  
 d) Inside the component only  
 Correct answer is Inside the action creator (with middleware) – Option b.

Q42. Which hook is often used in React-Redux for accessing state from the store?  
 a) useEffect  
 b) useReducer  
 c) useSelector  
 d) useContext  
 Correct answer is useSelector – Option c.

Q43. Which hook is used in React-Redux to dispatch actions?  
 a) useDispatch  
 b) useSelector  
 c) useReducer  
 d) useAction  
 Correct answer is useDispatch – Option a.

Q44. What does an action in Redux contain?  
 a) Reducers and state  
 b) State and middleware  
 c) Type and payload  
 d) Store and UI  
 Correct answer is Type and payload – Option c.

Q45. What happens if you try to mutate Redux state directly?  
 a) Redux automatically corrects it  
 b) State updates without problems  
 c) It breaks time-travel debugging and immutability principles  
 d) Nothing happens  
 Correct answer is It breaks time-travel debugging and immutability principles – Option c.

Q46. Which of the following is NOT true about Redux store?  
 a) It holds the application state  
 b) It allows direct modification of state  
 c) It is updated only through dispatching actions  
 d) It is created using createStore or configureStore  
 Correct answer is It allows direct modification of state – Option b.

Q47. What is the main purpose of Redux middleware?  
 a) To directly update the reducer  
 b) To log only errors  
 c) To intercept actions before they reach the reducer  
 d) To remove boilerplate code  
 Correct answer is To intercept actions before they reach the reducer – Option c.

Q48. Which library is recommended by the Redux team for writing Redux logic?  
 a) redux-observables  
 b) redux-toolkit  
 c) redux-thunk  
 d) redux-persist  
 Correct answer is redux-toolkit – Option b.

Q49. What does the mapStateToProps function do in React-Redux?  
 a) Maps component props to reducer  
 b) Maps Redux state to React component props  
 c) Maps React component props to Redux store  
 d) Maps dispatch to reducers  
 Correct answer is Maps Redux state to React component props – Option b.

Q50. If you want to initialize state in Redux with server data, which lifecycle stage is best?  
 a) Before rendering the component (useEffect)  
 b) Inside the reducer directly  
 c) Only during store creation  
 d) Inside UI event handlers  
 Correct answer is Before rendering the component (useEffect) – Option a.

RxJS & Redux-Observables, Reducers & Actions

Q51. What does RxJS primarily help with in React applications?  
 a) Managing UI components  
 b) Handling asynchronous data streams  
 c) Styling components  
 d) Building server-side apps  
 Correct answer is Handling asynchronous data streams – Option b.

Q52. What is a Redux-Observable?  
 a) A middleware for handling async logic with RxJS  
 b) A library for styling React apps  
 c) A testing framework for React  
 d) A server-side rendering library  
 Correct answer is A middleware for handling async logic with RxJS – Option a.

Q53. Which RxJS operator is commonly used in Redux-Observable epics for mapping actions?  
 a) switchMap  
 b) mapState  
 c) reduce  
 d) dispatchMap  
 Correct answer is switchMap – Option a.

Q54. In Redux, what is the role of a reducer?  
 a) Dispatching actions  
 b) Describing state changes based on actions  
 c) Fetching data from API  
 d) Handling authentication  
 Correct answer is Describing state changes based on actions – Option b.

Q55. Which of the following best describes Redux actions?  
 a) Functions that modify state directly  
 b) Pure functions  
 c) Plain JavaScript objects describing changes  
 d) Observables that emit state  
 Correct answer is Plain JavaScript objects describing changes – Option c.

Q56. Which operator cancels the previous observable when a new one is emitted?  
 a) mergeMap  
 b) switchMap  
 c) concatMap  
 d) map  
 Correct answer is switchMap – Option b.

Q57. What is the purpose of implementing Undo History in Redux?  
 a) To support rollback of UI state  
 b) To manage API requests  
 c) To improve performance  
 d) To simplify reducers  
 Correct answer is To support rollback of UI state – Option a.

Q58. Which Redux concept is most important for supporting undo/redo?  
 a) Middleware  
 b) Reducer composition  
 c) Immutable state  
 d) Async actions  
 Correct answer is Immutable state – Option c.

Q59. What does ImmutableJS provide?  
 a) Fast rendering  
 b) Persistent, immutable data structures  
 c) Authentication management  
 d) A testing framework  
 Correct answer is Persistent, immutable data structures – Option b.

Q60. Which of the following is a key advantage of ImmutableJS in Redux apps?  
 a) Larger bundle size  
 b) Direct state mutation  
 c) Performance optimization with structural sharing  
 d) Automatic API fetching  
 Correct answer is Performance optimization with structural sharing – Option c.

Q61. In RxJS, which operator is best for handling multiple values sequentially without cancellation?  
 a) switchMap  
 b) concatMap  
 c) mergeMap  
 d) flatMap  
 Correct answer is concatMap – Option b.

Q62. Which function is used in Redux to combine multiple reducers?  
 a) mergeReducers()  
 b) joinReducers()  
 c) combineReducers()  
 d) useReducers()  
 Correct answer is combineReducers() – Option c.

Q63. In Redux-Observable, what is an Epic?  
 a) A reducer that handles async data  
 b) A middleware function that listens for actions and returns new actions  
 c) A UI component that renders state  
 d) A Redux store enhancer  
 Correct answer is A middleware function that listens for actions and returns new actions – Option b.

Q64. ImmutableJS provides which method to update deeply nested data without mutation?  
 a) setDeep()  
 b) updateIn()  
 c) pushState()  
 d) changeIn()  
 Correct answer is updateIn() – Option b.

Q65. In Undo History implementation, which Redux principle ensures previous states can be restored?  
 a) Single source of truth  
 b) Actions must be pure  
 c) State is immutable  
 d) Store is asynchronous  
 Correct answer is State is immutable – Option c.

### Redux-Thunk & Redux-Saga MCQs

Q66. What is the primary purpose of Redux-Thunk?  
 a) To handle routing in React apps  
 b) To write action creators that return functions instead of actions  
 c) To test Redux reducers  
 d) To optimize React rendering  
 Correct answer is To write action creators that return functions instead of actions – Option b.

Q67. In Redux-Thunk, the function returned by an action creator receives:  
 a) Only dispatch  
 b) Only getState  
 c) Both dispatch and getState  
 d) Neither  
 Correct answer is Both dispatch and getState – Option c.

Q68. Which of the following is a use case for Redux-Thunk?  
 a) Testing React components  
 b) Managing asynchronous API calls  
 c) Writing reducers  
 d) Handling routing  
 Correct answer is Managing asynchronous API calls – Option b.

Q69. Redux-Saga is based on which JavaScript feature?  
 a) Promises  
 b) Generators  
 c) Callbacks  
 d) Async/Await  
 Correct answer is Generators – Option b.

Q70. In Redux-Saga, which effect is used to call asynchronous functions?  
 a) put  
 b) call  
 c) take  
 d) delay  
 Correct answer is call – Option b.

Q71. Which Redux middleware is best for complex async workflows like cancellation and sequencing?  
 a) Redux-Thunk  
 b) Redux-Saga  
 c) Redux-Logger  
 d) Redux-Observable  
 Correct answer is Redux-Saga – Option b.

Q72. The put effect in Redux-Saga is used to:  
 a) Dispatch an action  
 b) Call an API  
 c) Cancel a task  
 d) Create a reducer  
 Correct answer is Dispatch an action – Option a.

Q73. In Redux-Saga, the takeEvery effect does what?  
 a) Cancels previous sagas  
 b) Runs a saga for every matched action  
 c) Runs only the latest action saga  
 d) Runs sagas sequentially  
 Correct answer is Runs a saga for every matched action – Option b.

Q74. Which effect in Redux-Saga runs only the latest task and cancels previous ones?  
 a) takeLatest  
 b) takeEvery  
 c) fork  
 d) join  
 Correct answer is takeLatest – Option a.

Q75. What is a key difference between Redux-Thunk and Redux-Saga?  
 a) Thunk uses Promises, Saga uses Generators  
 b) Thunk is synchronous, Saga is asynchronous  
 c) Thunk is faster than Saga  
 d) Saga does not support async  
 Correct answer is Thunk uses Promises, Saga uses Generators – Option a.

Q76. Which middleware allows you to retry failed API calls automatically?  
 a) Redux-Thunk  
 b) Redux-Saga  
 c) Redux-Persist  
 d) Redux-Logger  
 Correct answer is Redux-Saga – Option b.

Q77. In Redux-Saga, yield call(apiFunction) ensures:  
 a) The function is executed asynchronously  
 b) The action is dispatched  
 c) The reducer is updated  
 d) The store is replaced  
 Correct answer is The function is executed asynchronously – Option a.

Q78. Which of these is NOT true about Redux-Thunk?  
 a) It is simpler to learn than Redux-Saga  
 b) It uses functions instead of plain actions  
 c) It is good for small to medium apps  
 d) It requires generator functions  
 Correct answer is It requires generator functions – Option d.

Q79. Which effect is used to pause execution in Redux-Saga for a given time?  
 a) wait  
 b) delay  
 c) timeout  
 d) sleep  
 Correct answer is delay – Option b.

Q80. Which scenario would benefit more from Redux-Saga than Redux-Thunk?  
 a) Simple API calls  
 b) Complex async workflows with cancellation  
 c) Dispatching synchronous actions  
 d) Static state  
 Correct answer is Complex async workflows with cancellation – Option b.

Q81. What is the main purpose of Redux-Thunk?  
 a) Handle asynchronous logic in Redux  
 b) Manage routing in Redux apps  
 c) Optimize rendering performance  
 d) Create reducers automatically  
 Correct answer is Handle asynchronous logic in Redux – Option a.

Q82. Redux-Thunk allows dispatching of:  
 a) Only objects  
 b) Functions and objects  
 c) Only strings  
 d) Reducers directly  
 Correct answer is Functions and objects – Option b.

Q83. Which middleware is required to enable Redux-Thunk?  
 a) redux-saga  
 b) redux-devtools  
 c) redux-thunk  
 d) redux-persist  
 Correct answer is redux-thunk – Option c.

Q84. In Redux-Saga, which effect is used to call an asynchronous function?  
 a) take  
 b) call  
 c) put  
 d) select  
 Correct answer is call – Option b.

Q85. What does the put effect in Redux-Saga do?  
 a) Calls an API  
 b) Dispatches an action  
 c) Waits for an action  
 d) Cancels a task  
 Correct answer is Dispatches an action – Option b.

Q86. Redux-Saga is built on top of:  
 a) Async/Await  
 b) Observables  
 c) Generator functions  
 d) Promises only  
 Correct answer is Generator functions – Option c.

Q87. In Redux-Saga, the takeLatest effect is used to:  
 a) Run all actions in sequence  
 b) Cancel previous tasks and run only the latest one  
 c) Queue all actions without canceling  
 d) Run actions in parallel  
 Correct answer is Cancel previous tasks and run only the latest one – Option b.

Q88. Redux-Thunk is best suited for:  
 a) Complex async flows with cancellation  
 b) Simple async logic like API calls  
 c) State immutability handling  
 d) Reducer composition  
 Correct answer is Simple async logic like API calls – Option b.

Q89. Which effect in Redux-Saga is used to watch for dispatched actions?  
 a) take  
 b) call  
 c) race  
 d) put  
 Correct answer is take – Option a.

Q90. Redux-Saga can handle:  
 a) Only synchronous logic  
 b) Complex asynchronous workflows  
 c) CSS styling  
 d) DOM rendering  
 Correct answer is Complex asynchronous workflows – Option b.

Q91. Which of the following is TRUE about Redux-Thunk?  
 a) It uses generator functions  
 b) It allows writing action creators that return functions  
 c) It replaces reducers  
 d) It is built on RxJS  
 Correct answer is It allows writing action creators that return functions – Option b.

Q92. In Redux-Saga, the select effect is used to:  
 a) Fetch data from an API  
 b) Access the Redux store state  
 c) Dispatch an action  
 d) Cancel an effect  
 Correct answer is Access the Redux store state – Option b.

Q93. Which is an advantage of Redux-Saga over Redux-Thunk?  
 a) Simpler for beginners  
 b) Handles side effects more declaratively  
 c) No need for middleware  
 d) Faster rendering always  
 Correct answer is Handles side effects more declaratively – Option b.

Q94. Which effect in Redux-Saga allows running multiple effects in parallel?  
 a) race  
 b) all  
 c) call  
 d) fork  
 Correct answer is all – Option b.

Q95. In Redux-Saga, what does the fork effect do?  
 a) Run a blocking call  
 b) Spawn a non-blocking task  
 c) Cancel all tasks  
 d) Dispatch a reducer  
 Correct answer is Spawn a non-blocking task – Option b.

Q96. What does i18n stand for in software development?  
 a) Integration  
 b) Internationalization  
 c) Interaction  
 d) Initialization  
 Correct answer is Internationalization – Option b.

Q97. How many letters are between the first and last letters in “Internationalization” that form i18n?  
 a) 15  
 b) 17  
 c) 18  
 d) 19  
 Correct answer is 18 – Option c.

Q98. Which React library is most commonly used for i18n support?  
 a) react-intl  
 b) react-localize-redux  
 c) react-i18next  
 d) next-translate  
 Correct answer is react-i18next – Option c.

Q99. In react-i18next, what is the hook used to access translation functions?  
 a) useLocale  
 b) useLang  
 c) useI18n  
 d) useTranslation  
 Correct answer is useTranslation – Option d.

Q100. Which JSON structure is typically used for storing translations?  
 a) Array of strings  
 b) Key-value pairs  
 c) CSV format  
 d) XML format  
 Correct answer is Key-value pairs – Option b.

Q101. In i18n, what is “l10n”?  
 a) Localization  
 b) Lexicalization  
 c) Legalization  
 d) Language-switching  
 Correct answer is Localization – Option a.

Q102. Which of the following is NOT an i18n challenge?  
 a) Different date formats  
 b) Currency conversion  
 c) String interpolation  
 d) Component state management  
 Correct answer is Component state management – Option d.

Q103. In i18next, what option allows you to fallback to a default language?  
 a) defaultLang  
 b) fallbackLng  
 c) baseLocale  
 d) backupLang  
 Correct answer is fallbackLng – Option b.

Q104. Which of these locales is valid for US English?  
 a) en-uk  
 b) en-us  
 c) eng-us  
 d) en-english  
 Correct answer is en-us – Option b.

Q105. What is the purpose of ICU message formatting in i18n?  
 a) To format database queries  
 b) To support pluralization and gender rules  
 c) To compress translation files  
 d) To sort translation keys  
 Correct answer is To support pluralization and gender rules – Option b.

Q106. What does the Trans component in react-i18next help with?  
 a) Language detection  
 b) Rendering translated strings with React elements inside  
 c) Switching between locales  
 d) Storing translations in cookies  
 Correct answer is Rendering translated strings with React elements inside – Option b.

Q107. Which of the following is NOT a feature of react-i18next?  
 a) Lazy loading translations  
 b) Context-based translations  
 c) Automatic pluralization  
 d) Automatic code-splitting  
 Correct answer is Automatic code-splitting – Option d.

Q108. If a translation key is missing, what will i18next usually display?  
 a) Error message  
 b) Empty string  
 c) The key itself  
 d) Default English  
 Correct answer is The key itself – Option c.

Q109. Which React feature can help with dynamic text changes in multiple languages?  
 a) Context API  
 b) useMemo  
 c) useEffect  
 d) Suspense  
 Correct answer is Context API – Option a.

Q110. Which statement is TRUE about localization?  
 a) It always uses English as fallback  
 b) It adapts content to specific regions/cultures  
 c) It only changes the text language  
 d) It ignores formatting of dates and numbers  
 Correct answer is It adapts content to specific regions/cultures – Option b.