

01	A	B	C	D	E
02	A	B	C	D	E
03	A	B	C	D	E
04	A	B	C	D	E
05	A	B	C	D	E
06	A	B	C	D	E
07	A	B	C	D	E
08	A	B	C	D	E
09	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E

14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E
17	A	B	C	D	E
18	A	B	C	D	E
19	A	B	C	D	E
20	A	B	C	D	E
21	A	B	C	D	E
22	A	B	C	D	E
23	A	B	C	D	E
24	A	B	C	D	E
25	A	B	C	D	E
26	A	B	C	D	E

27	A	B	C	D	E
28	A	B	C	D	E
29	A	B	C	D	E
30	A	B	C	D	E
31	A	B	C	D	E
32	A	B	C	D	E
33	A	B	C	D	E
34	A	B	C	D	E
35	A	B	C	D	E
36	A	B	C	D	E
37	A	B	C	D	E
38	A	B	C	D	E
39	A	B	C	D	E

PRACTICE QUESTIONS

OutSystems

PD11E

Professional Traditional Web Developer O11 (EN)

EDITION

Demo

QUESTIONS

9

FORMAT

Limited Edition PDF

Before you begin

What this is

9 practice questions for OutSystems **PD11E**, each with its answer key and an explanation. Answer a question before you read its key — the explanation is worth more once you have committed.

If something looks wrong

Send us three things and we will check it:

- the exam code, **PD11E**
- the question number
- the email address on your order

Write to **support@mometrix.net**. We reply within one business day.

QUESTION 1

SINGLE CHOICE

A developer needs to apply a style to a single Screen and have it override a style defined in the Theme. Which approach is most appropriate in OutSystems Traditional Web?

- ☐ A Add a new style class in the Theme so it is automatically available on every screen.
- ☒ B Use an inline style on the screen's container widget, scoped only to that screen.
- ☐ C Edit the Web Flow's default CSS file and redeploy the module.
- ☐ D Create a new application skin and assign it to the user role.

ANSWER B

WHY Inline styles added on a screen's container widget are scoped to that screen, so they override theme styles for that screen only without affecting other screens. Editing the Theme would change the application-wide styling, modifying the Web Flow CSS affects the whole module, and application skins change styling per role rather than per screen.

QUESTION 2

SINGLE CHOICE

Which widget property should a developer configure to ensure a mandatory input is validated on the server side in a Traditional Web screen?

- ☒ A Set the widget's Mandatory property to Yes and provide a Validation Message.
- ☐ B Set the widget's Required property to Yes in the Extended Properties.
- ☐ C Add the widget inside an If widget bound to ServerValid.
- ☐ D Enable the Form's ValidateOnSubmit property.

ANSWER A

WHY In OutSystems Traditional Web, marking an input widget as Mandatory (Yes) and supplying a Validation Message triggers both client-side indication and server-side validation when the form is submitted. The Required attribute is an HTML attribute, not the OutSystems server-side validation mechanism. ServerValid is not the validation hook for inputs, and there is no single ValidateOnSubmit property that governs per-field validation.

QUESTION 3

SINGLE CHOICE

A developer needs to count the number of active Customers grouped by Country. Which Aggregate configuration is correct?

- ☒ **A** Add a Filter where IsActive = True, group by Country, and use the Count aggregate function on Customer.Id.
- ☐ **B** Add a Sort by Country and use the Count aggregate function on Customer.Name.
- ☐ **C** Use the SQL widget to execute a custom SELECT COUNT(*) GROUP BY statement.
- ☐ **D** Add a Join to the Orders entity and group by Order.Country.

ANSWER A

WHY To group and count in an Aggregate, you filter by the relevant condition (IsActive = True), group by the desired attribute (Country), and apply Count to an identifier (typically the primary key Id). Sorting does not affect grouping, the SQL widget is used only when the requirement cannot be expressed with an Aggregate, and grouping by Order.Country would produce results based on the joined entity rather than the Customer.

QUESTION 4

SINGLE CHOICE

In a Traditional Web screen, which event is triggered only once, when the screen is first rendered, and is the correct place to run initialization logic that should not re-run on AJAX refreshes?

- ☐ **A** Screen Preparation
- ☒ **B** On Initialize
- ☐ **C** On Render
- ☐ **D** On Destroy

ANSWER B

WHY The On Initialize event of a Traditional Web screen runs exactly once when the screen is first requested, making it the appropriate place for one-time initialization. Screen Preparation runs every time the screen is prepared, including on AJAX refreshes; On Render and On Destroy are not standard screen lifecycle events in OutSystems Traditional Web.

QUESTION 5

SINGLE CHOICE

Which statement best describes the purpose of a Web Flow entry point in an OutSystems Traditional Web application?

- ☒ **A** It defines the URL through which users start navigating a Web Flow and the first Screen they see.
- ☐ **B** It exposes a Server Action as a REST endpoint for external consumers.
- ☐ **C** It schedules a Timed Task that runs at application startup.
- ☐ **D** It is the default destination Screen when a destination Screen is not specified in a Link or Button.

ANSWER A

WHY A Web Flow entry point defines how users enter the flow, including the URL and the initial Screen to display. Exposing a REST endpoint is done through a REST service method, scheduled execution is handled by Timers, and there is no generic 'default screen' fallback for Links or Buttons in Traditional Web navigation.

QUESTION 6

SINGLE CHOICE

A developer wants to make sure that only authenticated users can access certain Screens in a Web Flow. What is the recommended approach in Traditional Web?

- ☐ **A** Check the UserId variable in the Screen Preparation and call a Notify action if empty.
- ☒ **B** Add a Check Role or Check Web Function widget at the entry point and define the allowed roles.
- ☐ **C** Place an If widget around each Link to hide navigation links from anonymous users.
- ☐ **D** Disable anonymous access by unchecking the Anonymous property on each Screen.

ANSWER B

WHY The recommended pattern is to secure the Web Flow entry point with a Check Role or Check Web Function widget so that only authorized users can reach the flow's Screens. Manually checking UserId and notifying the user is not a secure access control pattern. Hiding links via an If widget is only cosmetic and does not block access, and there is no Anonymous property per Screen that performs access enforcement in Traditional Web.

QUESTION 7

SINGLE CHOICE

A developer wants to display a user-friendly error message when a Server Action fails during a button click. Which is the most appropriate way in Traditional Web?

- ☒ **A** Wrap the Server Action call inside an Exception Handler widget and define the On Exception branch with a Notification.
- ☐ **B** Add an Error widget at the bottom of the Screen so it shows whenever the screen renders.
- ☐ **C** Call the built-in HandleError() function from the Logic tab and refresh the screen.
- ☐ **D** Set the Screen's OnRender property to return an error message string.

ANSWER A

WHY The Exception Handler widget is the standard mechanism to catch exceptions raised from a block of logic and react to them, typically by showing a Notification with a friendly message. An Error widget shows static error messages for failed validation, not exceptions from server actions. HandleError() is not a standard OutSystems function, and OnRender is not a screen property used for returning error text.

QUESTION 8

SINGLE CHOICE

What is the correct scope of a Site Property in an OutSystems Traditional Web module?

- ☒ **A** It holds a value that is shared across all users of the module and persists across requests.
- ☐ **B** It holds a value specific to the current user's session and is reset when the session ends.
- ☐ **C** It holds a value scoped to a single Screen and is only accessible within that Screen.
- ☐ **D** It holds a value stored in the user's browser cookies and travels with every HTTP request.

ANSWER A

WHY A Site Property in OutSystems holds a value that is shared across all users and sessions for the module, and can be evaluated at runtime. Session-scoped values are stored in session variables, screen-scoped values in Local Variables of the Screen, and values traveling with every request are kept in cookies or request parameters — not Site Properties.

QUESTION 9

MULTIPLE CHOICE

Which of the following are valid ways to model a one-to-many relationship between a Customer entity and an Order entity in OutSystems? (Choose two.)

- ☒ **A** Add a foreign-key Attribute named CustomerId of type Customer Identifier on the Order entity.
- ☒ **B** Create an Attribute of type Order Entity on the Customer entity.
- ☐ **C** Define a many-to-many association between Customer and Order and rely on OutSystems to materialize a join table.
- ☐ **D** Add a foreign-key Attribute of type Customer Identifier on Customer pointing to Order.

ANSWER A - B

WHY A one-to-many relationship in OutSystems is modeled by placing a foreign-key identifier (CustomerId of type Customer Identifier) on the 'many' side (Order), and OutSystems automatically creates a reverse navigation Attribute of type Order Entity on the 'one' side (Customer). A many-to-many association is used for many-to-many relationships, not one-to-many, and placing the foreign key on the parent side would invert the relationship incorrectly.