**What is Context in Odoo?**

In Odoo, **context** is a dictionary (dict) that carries additional data while executing an operation. It helps pass information between different parts of the system, such as actions, views, fields, and methods. Context can be used to modify behavior dynamically based on user actions.

**Step 1: Where is Context Used?**

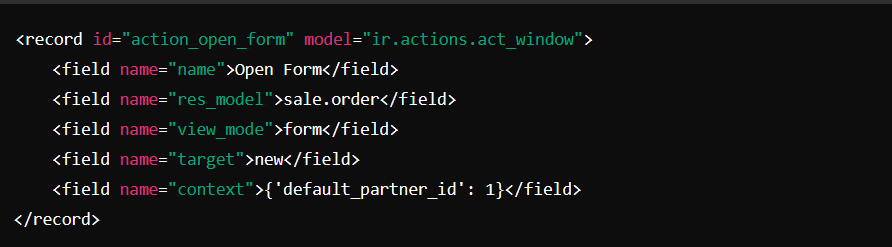
Context is commonly used in:

1. **Actions** (e.g., passing default values to forms)
2. **Views** (e.g., making fields readonly based on conditions)
3. **Methods** (e.g., filtering data dynamically)
4. **Reports** (e.g., generating reports with specific filters)

**Step 2: Passing Context in Actions**

When defining an action in Odoo, you can pass context in the context key.

**Example: Setting a Default Value in a Form**

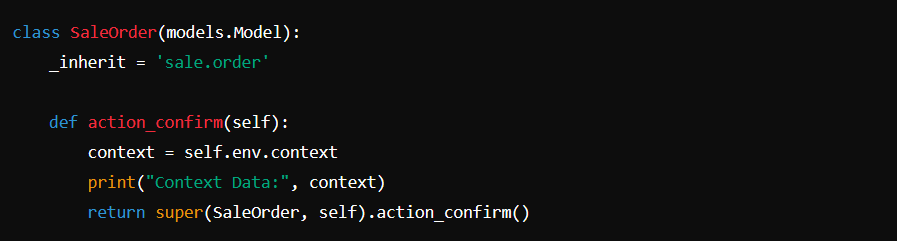
**Explanation:**

* This action will open the **Sale Order** form with partner\_id pre-filled as 1.

**Step 3: Passing Context in Python Methods**

In Python, the context is usually passed as self.env.context.

**Example: Get Context Data in Python**



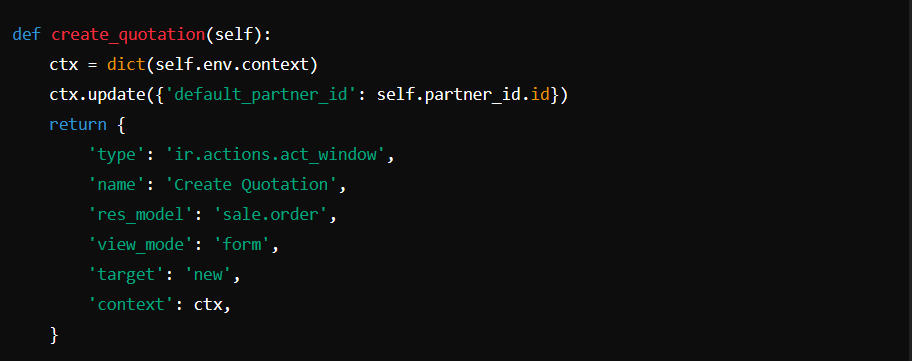
**Explanation:**

* self.env.context retrieves the current context.
* You can print or use the context values in your logic.

**Step 4: Modifying Context in Python**

You can modify context while calling a method or creating a record.

**Example: Updating Context**



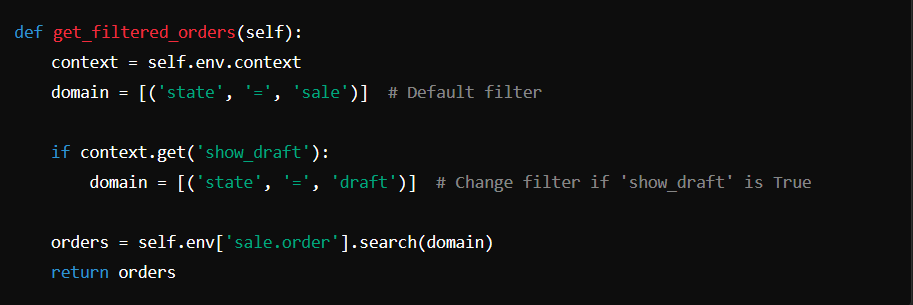
**Explanation:**

* A new quotation form will open with the **default partner pre-filled**.

**Step 5: Using Context in Search Operations**

Context can be used in search() to filter records dynamically.

**Example: Filtering Data Based on Context**



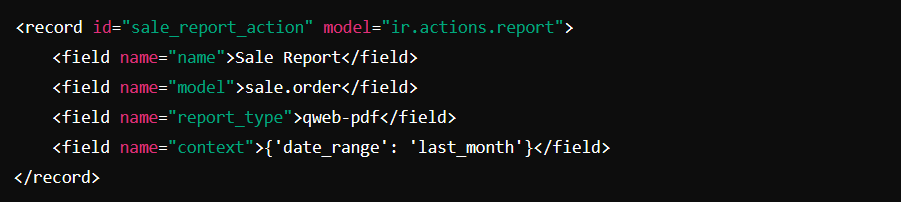
**Explanation:**

* If show\_draft is passed in context, only draft orders will be fetched.

**Step 6: Using Context in Reports**

Context is also useful in reports to generate filtered data.

**Example: Passing Context in Report Action**



**Explanation:**

* The report will be generated only for sales orders from **last month**.

**Conclusion**

Context is a powerful feature in Odoo that helps pass information between views, actions, methods, and reports. It makes applications more **dynamic and customizable**.