 🎨 **What is a Component?**  
Components are **reusable design elements** in Figma. They consist of a **Master (parent)** and multiple **Instances (children)**, which can have unique attributes while staying connected to the Master.

 🛠️ **Creating Components**  
To make a Component, select a design element (e.g., a button) and click "Create Component." The element transforms into a Master Component, **visible in the Assets panel** for easy reuse.

 🔄 **Instances and Overrides**  
Instances inherit properties from the Master but allow for **custom changes** like color, text, or rounded corners. These overrides let designers customize elements while maintaining consistency.

 📋 **Benefits of Components**

* **Scalability:** Design updates in the Master automatically reflect across all Instances.
* **Efficiency:** Simplifies reuse, reduces redundancy, and ensures design consistency.
* **Control:** Overrides can be reset to revert Instances back to the Master’s settings.

 🖼️ **Practical Examples**  
Buttons, logos, and other reusable elements can be turned into Components. These Components can be managed via the Assets panel and updated globally for seamless workflows

* 🎛️ **Push and Reset Overrides**
  + **Push Overrides:** Changes made to an Instance can be pushed back to the Master, ensuring consistency across all Instances.
  + **Reset Overrides:** Instances can be reverted to match the Master Component by resetting all modifications.
* 🔄 **Switching Instances and Versions**
  + **Switch Components:** Replace one Instance with another using drag-and-drop or menu options, useful for icons or design variations.
  + **Light & Dark Versions:** Detach an Instance to create entirely new Components for different themes, such as light and dark modes.
* 💡 **Multiple Components and Efficiency**
  + **Batch Creation:** Select multiple elements and create Components simultaneously with "Make Multiple Components," saving time.
  + **Asset Management:** Components are stored in the Assets panel, making it easy to locate and reuse design elements.
* 🧑‍🎨 **Advanced Features**
  + **Descriptions and Documentation Links:** Add descriptions and URLs to Components for detailed design specifications or system documentation.
  + **Limitations of Masters:** Switching only works with Instances, not with Master Components, which have unique properties and additional controls.

**Insights Based on Numbers**

* **2 Versions (Light & Dark):** Demonstrates the flexibility of design by maintaining theme-specific variations using Components.
* **6 Arrow Variants:** Highlighted during batch creation, showing how multiple design elements can be converted into Components at once.
* 🛡️ **Main Components Are Protected**
  + **Undeletable Nature:** Main Components can’t be permanently deleted as long as at least one Instance exists. Deleting the Main removes it from the file but allows recovery through Instances.
  + **Restoration Process:** Right-clicking an Instance offers the option to “Restore Main Component,” bringing it back into the file.
* 🌍 **Cross-File Components**
  + **External Components:** Instances from other files can be pasted into your design. They reference the original Main Component from the source file.
  + **Reconnection:** You can right-click an Instance and navigate back to its Main Component, even if it resides in another file.
* ✂️ **Detaching and Modifying Instances**
  + **Detach Instance:** Breaks the link between an Instance and its Main Component, making it a standalone element.
  + **Creating New Components:** Detached Instances can be converted into new Components for customization.
* 👁️ **Hidden Elements in Instances**
  + **Hiding vs. Deleting:** Elements in an Instance are hidden (using the “eyeball” visibility toggle) rather than permanently deleted, ensuring consistency with the Main Component.
* 🔗 **Team Libraries and Updates**
  + **Team Libraries:** A more advanced system for managing shared Components across files, enabling synchronized updates (covered later in the course).

**Insights Based on Numbers**

* **1 Master, Multiple Instances:** Shows how a single Master Component can support a design with multiple Instances.
* **Cross-File Efficiency:** Demonstrates the potential to share Components across projects, enhancing collaboration.

**Best Practices for Organizing Main Components in Figma**

** 📁 Centralize Main Components**

* **Components Page: Create a dedicated page for all Main Components to keep them organized and accessible, especially in large projects.**
* **Separate Masters and Instances: Keep the Masters on the Components page and use Instances across other pages to maintain cleanliness and usability.**

** 🧹 Keep Files Tidy**

* **Move Masters: Right-click on a Component and move it to the Components page for better organization.**
* **Search Functionality: Use the search bar in the Assets panel to quickly find Components by name.**

** 🔄 Practical Tips for Maintenance**

* **Naming Conventions: Use clear and consistent names for Components to make them easy to locate in the Assets panel.**
* **Grouping Similar Components: Organize Components by type (e.g., buttons, modals, cards, logos) within the Components page for better structure.**
* **Design Spec Pages: Eventually create a design system or spec page to further structure Components visually.**

** 🔍 Using Search Effectively**

* **List Mode: Switch to list mode in the Assets panel for a detailed view of Components.**
* **Search with a Space Bar: Typing a space in the search bar displays all Components, serving as a workaround for visualizing the full list.**

** 📋 What Are Naming Conventions?**

* **Using forward slashes ("/") in names organizes Components into nested groups within the Assets panel.**
* **This structure makes Components easier to find, especially in large projects or shared files.**

** 🏗️ How to Implement Naming Conventions**

* **Basic Example: Use names like Arrow/Right and Arrow/Left to group related Components under a single category (Arrow).**
* **Deeper Nesting: Add more detail, such as Button/Large/Light Mode or Button/Small/Dark Mode, to create hierarchical groupings.**
* **Consistency: Ensure naming conventions follow a logical pattern to maintain clarity across files.**

** 🛠️ Practical Applications**

* **Buttons: Organize variations by size (Large, Small) and style (Light Mode, Dark Mode).**
* **Icons: Group icons by function, such as Filled/Actions or Outline/Hardware, as demonstrated with Material Design examples.**
* **Design Systems: Use naming conventions for design system Components, like snack bars, to streamline workflows.**

** 🚀 Benefits for Collaboration and Scaling**

* **Teamwork: Ensures designers and developers can quickly locate and utilize Components.**
* **Variables and Future Enhancements: Paves the way for efficient use of variables, which rely on consistent naming.**