Pure Data Workshop Basics

What is Pure Data?
Pure Data (Pd) is an open-source visual programming language designed for creating interactive multimedia projects, such as music, sound design, visuals, and more. It uses "patches" made of objects connected by virtual cables to process data in real time.

Core Concepts

1. **Patch**: The canvas where you create your program. It's a visual representation of your project.
2. **Object**: The building blocks of Pd. Each object performs a specific function.
    - Example: `[osc~]` generates a sine wave.
3. **Messages**: Small bits of data sent between objects.
    - Example: Sending `440` to `[osc~]` sets the frequency to 440 Hz.
4. **Connections**: Wires between objects that determine the flow of data.
    - **Audio (signal)** connections are thicker and represent sound.
    - **Control (message)** connections are thinner and represent instructions.

Basic Objects and Their Functions

- `[osc~]`: Generates a sine wave.
- `[dac~]`: Outputs sound to your speakers.
- `[metro]`: Creates a repeating trigger (useful for rhythms).
- `[bang]`: A trigger that sends a single message.
- `[number box]`: Displays and manipulates numerical values.
- `[+], [-], [*], [/]`: Perform basic math operations.
- `[toggle]`: Switches between on/off states.

Essential Techniques

1. **Connecting Objects**: Drag from an outlet (bottom of an object) to an inlet (top of another object).
2. **Editing vs. Running Mode**:
    - **Edit Mode** (Ctrl/Cmd + E): Move and modify objects.
    - **Running Mode**: Interact with the patch (click, input data, etc.).

3. **Saving Your Work**: Always save patches with descriptive names. Pd files use the `.pd` extension.

Basic Exercises

1. **Simple Sine Wave Generator**
   - Use `[osc~]` connected to `[dac~]`. Control the frequency with a number box.
2. **Basic Metronome**
   - Use `[metro]`, `[bang]`, and `[toggle]` to create a rhythmic pulse.
3. **Interactive Volume Control**
   - Add `[*~]` between `[osc~]` and `[dac~]` to multiply the signal, and control it with a slider.

Workshop Goals

- Understand the building blocks of Pure Data.
- Create simple sound-generating patches.
- Explore basic interactivity and real-time control.
- Foster creativity and experimentation.

Basic Pure Data Workshop Outline

Here's a logical order for covering the basics in a Pure Data (Pd) workshop:

---

1. Introduction to Pure Data

- What is Pure Data? (visual programming for audio, visuals, interactivity)
- Overview of the interface (menus, patch canvas, toolbar)

---

2. Core Concepts

- Patches: What they are and how they work
- Objects: The building blocks of a patch
- Messages and connections: Control flow and audio flow
- Modes: Edit mode vs. Running mode

---

3. Creating Your First Patch

- Add and connect objects (`Ctrl/Cmd + 1` to create objects)
- Basic sound example:
    - `[osc~]` → `[dac~]`
    - Use a number box to control frequency

---

4. Common Objects and Their Uses

- Audio signal objects: `[osc~]`, `[phasor~]`, `[noise~]`
- Control objects: `[metro]`, `[bang]`, `[toggle]`, `[message]`
- Math objects: `[+], [*~], [/], [-]`

---

5. Practical Examples

- **Exercise 1**: Build a sine wave generator with frequency control
- **Exercise 2**: Create a simple metronome using `[metro]` and `[bang]`
- **Exercise 3**: Add a volume control using `[*~]` and a slider

---

6. Workshop Techniques

- Debugging: Finding and fixing patch errors (check connections, use `[print]`)
- Saving patches: Organize your files and always test them

---

7. Experimentation and Q&A

- Encourage participants to modify their patches (e.g., add multiple oscillators or change tempos).
- Provide time for questions and open exploration.