

Introduction to Rigging

Introduction to Rigging

Key concepts, terminology, and workflow steps for building a functional rig in 3D animation.

1. What Is Rigging?

Aspect	Description
Definition	The process of creating a controllable skeleton (bones, joints, and controllers) that drives a 3D model's deformation and animation.
Goal	Provide animators with intuitive, reliable controls while preserving the model's shape and volume during motion.
Typical Pipeline	Modeling ? Rigging ? Skinning/Weight?Painting ? Animation ? Rendering.

2. Core Components

Component	Purpose	Common Practices
Bones / Joints	Define the hierarchical structure and articulation points.	Use a clean hierarchy (root ? spine ? limbs). Keep joint orientation consistent.
Controllers	UI objects (curves, shapes, nulls) that animators manipulate.	Separate translation, rotation, and scale controls; use visual shapes/color coding.
Constraints	Link one object's transform to another (e.g., Parent , Point , Orient , Aim).	Use constraints sparingly; prefer driven keys or custom attributes for performance.
IK (Inverse Kinematics)	Solve joint positions automatically based on end?effector placement.	Foot/hand rigs often use IK; add pole vectors to control knee/elbow direction.
FK (Forward Kinematics)	Animate joints sequentially (root ? tip).	Ideal for arcs, swinging motions; blend with IK via switch attributes.

Introduction to Rigging

Component	Purpose	Common Practices
Blend/Switch Attributes	Enable smooth transition between IK and FK or between different pose sets.	Use a 0?1 float attribute; connect via set?driven keys or utility nodes.
Deformers	Additional shape control (e.g., Spline , Cluster , BlendShape).	Use for facial rigs, muscle bulges, or secondary motion.
Custom Attributes	Expose parameters (e.g., "Curl", "Spread") for animators.	Add to controller objects; drive nodes via utility networks.

3. Rigging Workflow – Step?by?Step

1. Preparation

- Verify clean topology (even edge flow, no non?manifold geometry).
- Freeze transformations and delete history on the mesh.
- Align the model's pivot to the world origin for easier joint placement.

2. Skeleton Creation

- Place joints at anatomical landmarks (e.g., pelvis, knee, ankle).
- Keep joint orientation consistent (X?axis points down the bone, Y?axis points up).
- Build a logical hierarchy (root ? spine ? neck ? head; arms/legs as children).

3. Controller Setup

- Create NURBS curves or shapes for each major joint.
- Color?code (e.g., red = translation, green = rotation, blue = scale).
- Group controllers to keep transforms clean (offset group ? controller ? shape).

4. IK/FK Systems

- Duplicate joint chain: one for FK, one for IK.
- Add **IK handles** (e.g., rotate?plane for limbs, spline IK for spines).
- Insert **pole vector** controls for elbows/knees.
- Blend between IK/FK using a switch attribute on the master controller.

Introduction to Rigging

5. Constraints & Connections

- Use **parent constraints** to bind controllers to joints.
- Apply **orient constraints** for rotational alignment.
- Connect custom attributes to utility nodes (e.g., **multiplyDivide**, **condition**) for advanced behavior (foot