<< Back

Blender

1 unit = 1m
Blender Launcher
Blender Shortcuts

Shortcuts

Manual adjustments Ctrl , preferences

General

Action	Shortcut
Property box	F9
Apply to all selected elements	hold Alt
Frame selected	. (numpad)
Frame camera	0 (numpad)
Set selected item as active viewpoint	Ctrl 0
Set active camera to view	Ctrl Alt 0
Transfrom all selected objects	Alt (hold)
Join objects	Ctrl j
Move to collection	M
Link to collection	Shift M
Radial view selection menu	`
Scale around z axis	Alt s
Cursor/view to center	Shift c
Place curcor	Shift right mouseclick
Drag viewport/window	Shift middle mouse
Zoom viewport/window	Ctrl middle mouse
Hide selected	h
Isolate selected	Shift h
Unhide all	Alt h
Quick favorites	q
Snap to orthographic views	hold Alt while rotating
Toggle quad view (orthographic and perspective)	Ctrl Alt q
Batch rename	Ctrl f2

Rendering

Action	Shortcut
Render frame	F12
Render sequence	Ctrl F12

Action	Shortcut
Render region	Ctrl b
Delete render region	Ctrl Alt b

Modeling

Action	Shortcut
Extrude point, edge or poly	e
Select edge/poly loop	Alt click (the loop will follow the direction relative to where you clicked)
Rotate allong edges	Ctrl Shift Alt s
Even mode when doing loop cut	e
Subdivide (combine with Auto Smooth for best experience)	Ctrl 1,2,3,... or Ctrl 0 to reset to original mesh
Snap to points with ruler	Hold Ctrl
Scale along normals	Alt s
Measure thickness with ruler	Hold Shift
Mesure angle with ruler	Click in the middle
Delete a ruler	x
Inset	i
Toggle constraint to outer edge while insetting	i + b

Sculpting

Action	Shortcut
Perspective/orthographic	numpad 5
Rescale brush	f
Intensity brush	Shift f
Invert brush	hold Ctrl
Smooth	Shift
Stroke method	e
Dyntopo	Ctrl d
Invert Masked	Alt i
Undo mask	Alt m
Hide masked	Ctrl h
Unhide all	h
Lasso mask	Ctrl Shift lmb
box mask	b

Rigging

Action	Shortcut
Set pivot point	.

Animation

Action	Shortcut
Set keyframe	i

Unwrapping

Action	Shortcut
Select UV island	i

Shading

Action	Shortcut
Preview shader	Ctrl Shift click
Principled texture setup	Ctrl Shift t
Cut node wires	Ctrl right-click drag

Tracking

Action	Shortcut
Create new tracker	Ctrl click
Show search box	Alt s
Track manually	Alt arrows
Lock view to tracker	l
Start auto tracker	Ctrl t
Lock tracker	Ctrl l

Boxcutter/Hardops

Add-on Specific

Install Hardops first, then Boxcutter. This way Boxcutter is the first mode you access when pressing Alt w

Action	Shortcut
Hardops/Boxcutter	
Toggle between Hardops & Boxcutter	Alt w
Viewport settings	Alt v
Align boxcutter	Shift v
Hardops menu	q
Radial hardops menu	Shift q
Box cutter menu/hopstools	d
HOps helper	Ctrl `
Hold shift when releasing cutter	Displays and selects your cutter
Mirror	Alt x
Allign edges	Alt a
MaterialIQ	
Toggle asset library	press e over any area

Add-on's

Name	Description
Loop tools	Mesh modelling toolkit. Several tools to aid modelling.
Edit Mesh Tools	Mesh modelling toolkit. Several tools to aid modelling.
Node wrangler	Improve the node experience by a lot
MACHIN3tools	MACHIN3tools
CURVEmachine	Makes POLY Curve editing more flexible
Hard Ops / Boxcutter	Hard Ops / Boxcutter
Edge Flow	Helps adjusting mesh geometry to curved surfaces.
GrabDoc	A trim & tileable baker for Blender.
Quick Rigid	Easy access to the most used rigid body settings.
Poly Haven	Asset library integration of all polyhaven assets
Materialiq	Material library
EZLattice	Lattice companion for Blender.
AddRoutes	Work with MIDI and OSC signals in Blender
PinVerts	Pin unselected vertices for Proportional Editing to aid in modelling
Shot manager	Render manager
fSpy	Open source still image camera matching
KeenTools	GeoTracker
Dream textures	AI-generate textures, ...
Pallaidium	AI-generate video, image, and audio from text prompts
PhyX	Scene layout tool
Engon	Asset browser
Udin Shaders	Car shaders
Poly Quilt	Retopology tool
StoryPencil	Storyboarding
Gaffer	Light & Hdri Manager
Light manager	Light manager
Sketchup Importer	Tested with 3.6
Compify	Easier/better compositing in 3D space
3DGS Render	Gaussian Splatting Editor/Renderer
DreamUV	Tools that allow you to manipulate UVs in the 3D viewport
BoltFactory	Add a bolt or nut.
Sculpt Bridge Tool	Create a bridge or punch a hole in a mesh
Shaders Plus	Caustics, Thin Film, Dispersion For Cycles & Eevee
Gobos Plus	Procedural & 4K Gobos For Cycles + Eevee
Ies Plus	Procedural Ies Light Nodes
Light Wrangler	Modify lights faster with shortcuts etc
Lens Sim	Lens simulation for Bokeh, Chromatic Aberration
nijiGPen	Adds new features to Grease Pencil for creating 2D graphic design and illustrations
Grease pencil tool wheel	Extended pie menu for selecting Grease Pencil tools quickly.
Camera Plane	Import images and stick them to the camera
Timelapse orbit	Create rotating viewport to record

A curated list of Blender Add-on's

Python scripts

Assign shaders to selected meshes with the same name as the shader

```
import bpy
# Get selected objects
selected_objects = bpy.context.selected_objects
# Loop through selected objects
for obj in selected_objects:
    if obj.type == 'MESH':
        # Check if there's a material with the same name as the object
        matching_material = bpy.data.materials.get(obj.name)
        if matching_material:
            # Clear existing materials on the object
            obj.data.materials.clear()
            # Assign the matching material to the object
            obj.data.materials.append(matching_material)
            print(f"Assigned material '{matching_material.name}' to object
'{obj.name}'")
        else:
            print(f"No matching material found for object '{obj.name}'")
print("Material assignment complete for selected objects.")
```

Quickly Import KitBash3D Models into the Blender Asset Browser

```
# Open your KB3D Pack. Make sure the textures are loaded and everything
works as intended. This script assumes the Scene is named
KB3D_"TitleCasePackName"-Native. It should be that by default.
# Create a new folder in your Asset Library Folder for Blender.
(Edit>Preferences>FilePaths>AssetLibraries). I recommend a setup like this
# (D:/BlenderAssetLibraries/KB3D/CyberDistricts/Cyberdistricts.blend)
# Where "CyberDistricts" would be the AssetLibrary.
# Open the script in Blender by going to the Text Editor and clicking "Open"
to locate the file, or paste it from the Gist.
# Run the script by clicking the "Run Script" button in the Text Editor or
pressing Alt+P.
# The script will create new collections for each type of asset in your
scene and move the assets into these collections. It will also generate a
unique ID for the asset pack and each asset in the pack.
# The script will then update the asset catalog file to include the new
asset pack and each asset in the pack.
# Once the script is finished, you can save your Blender file and close it.
# You should now see it in the asset libraries section on Blender.
# Troubleshooting: If something goes wrong or a mistake is maid I have
minimal errorchecking. Just undo before the script was run and delete the
Catalogs or the "blender_assets.cats.txt" file,
# they may trip up the script if it failed partway through.
```

```
# I put a lot of effort into making this, so if you'd like to show your
appreciation-
#   you can donate to me through PayPal at the link below:
https://paypal.me/mintyfresh

# Update: Remade better.
# TODO: Add simple GUI, test all packs, ensure consistent sizing and proper
integration for easy placing of props.

import bpy
import os
from pathlib import Path
import uuid

#filename
file_name = str(bpy.data.scenes.keys()[0].split('_')[1].split('-')[0])
folder = Path(bpy.data.filepath).parent
new_uuid = str(uuid.uuid4())

# Mapping for collections to asset types
collection_map = {
    "Props": "Prop",
    "Buildings": "Bldg",
    "Structures": "Strc",
    "Vehicles": "Vehi",
    "Creatures": "Crea"
}
kit_catalog = {}

flipped_cmap = {v: k for k, v in collection_map.items()}

# Create the primary collections
primary_collections = ["Props", "Buildings", "Structures", "Vehicles",
"Creatures", "Other"]
for collection_name in primary_collections:
    new_collection = bpy.data.collections.new(collection_name)
    bpy.context.scene.collection.children.link(new_collection)

# Loop through all the empties and sort them into the collections for the
library
for obj in bpy.data.objects:
    if obj.type == "EMPTY" and obj.name.startswith("KB3D_"):

        # Sort the new collection into the correct group
        asset_type = obj.name.split("_")[2][:4]

        # Check if the asset_type exists in the flipped_cmap dictionary
        if asset_type in flipped_cmap:
            cm_col_name = flipped_cmap[asset_type]
        else:
```

```
            cm_col_name = "Other"

        # Create a new collection with Empty's name
        new_collection = bpy.data.collections.new(obj.name)
        bpy.data.collections[obj.name].objects.link(obj)

        # Linking Object groups
        cm_col = bpy.data.collections[cm_col_name]
        # Move the object's children to the new collection
bpy.data.collections[cm_col.name].children.link(bpy.data.collections[obj.nam
e])
        kit_catalog[obj.name] = cm_col.name
print(kit_catalog.values())

# Loop over Object and relink to parents.
for obj in bpy.data.objects:
    if obj.type == "MESH" and obj.parent != None:
        bpy.data.collections[obj.parent.name].objects.link(obj)
for empty in bpy.data.objects:
    if empty.type == 'EMPTY' and empty.name.startswith('KB3D_'):
        bpy.context.scene.collection.objects.unlink(empty)
        for child in empty.children:
            bpy.context.scene.collection.objects.unlink(child)

for col in bpy.data.collections:
    if col.name.startswith('KB3D_'):
        col.asset_mark()

# Create a list of all empty objects
empty_objects = [obj for obj in bpy.data.objects if obj.type == "EMPTY" and
obj.name.startswith("KB3D_")]

# Loop through the list of empty objects and clear their locations
for obj in empty_objects:
    obj.location = (0, 0, 0)

asset_catalog_path = folder / "blender_assets.cats.txt"

# Initialize the list of lines and the asset_uuids dictionary
lines = []
asset_uuids = {}

# Read the existing asset catalog file into a list of lines
with asset_catalog_path.open('a+') as f:
    f.seek(0)  # Move the file pointer to the beginning of the file
    lines = f.readlines()

# Check if the blender_assets.cats.txt file is empty and initialize it if
necessary
if os.path.getsize(asset_catalog_path) == 0:
    header_lines = [
```

```
        "# This is an Asset Catalog Definition file for Blender.\n",
        "#\n",
        "# Empty lines and lines starting with `#` will be ignored.\n",
        "# The first non-ignored line should be the version indicator.\n",
        "# Other lines are of the format
\"UUID:catalog/path/for/assets:simple catalog name\"\n",
        "\n",
        "VERSION 1\n\n"
    ]
    lines.extend(header_lines)

# Add new lines to the catalog if necessary
file_contents = ''.join(lines)

# Initialize the asset_uuids dictionary outside the condition
collection_uuids = {collection_name: [] for collection_name in
collection_map.keys()}
collection_uuids["Other"] = []

if file_name not in file_contents:
    if "KB3D" not in file_contents:
        lines.append(f"{str(uuid.uuid4())}:KB3D:KB3D\n")
    lines.append(f"{str(uuid.uuid4())}:KB3D/{file_name}:{file_name}\n")

    for collection_name in collection_map.keys():
        asset_uuid = str(uuid.uuid4())
        asset_uuids[collection_name] = asset_uuid
lines.append(f"{asset_uuid}:KB3D/{file_name}/{collection_name}:{collection_n
ame}\n")

    asset_uuid = str(uuid.uuid4())
    asset_uuids["Other"] = asset_uuid
    lines.append(f"{asset_uuid}:KB3D/{file_name}/Other:Other\n")

# Write the updated list of lines back to the file
with asset_catalog_path.open("w") as f:
    f.writelines(lines)

kit_catalog = {}

# Asset Library Adder
for col in bpy.data.collections:
    if col.name.startswith("KB3D_"):
        # Gets the asset type of an object from the Name
        # Ex: Bldg Prop Strc
        asset_type = col.name.split("_")[2][:4]  # e.g. "Bldg"

        # Check if the asset_type exists in the flipped_cmap dictionary
        if asset_type in flipped_cmap:
            asset_name = flipped_cmap[asset_type]  # e.g. "Buildings"
```

```
            col.asset_data.catalog_id = asset_uuids[asset_name]
        else:
            asset_name = "Other"
            col.asset_data.catalog_id = asset_uuids[asset_name]  # Set the
catalog_id for the "Other" category

        print(col.name, asset_name)
        # Linking Object groups
        cm_col = bpy.data.collections[col.name]  # Buildings [Collection]
        print('adding', col.name, 'asset_data')
```

Demo Scenes

Description	Name
Fake caustics	Refractive Caustics (Blender 3.2+)
Geo nodes presets wit fields (like C4D)	Geo Nodes presets
	Abstract Loops

Links

Beginners tutorial Blender Shortcuts Manual Export Houdini KineFX Rigs & BlendShapes to Blender

Pre-production

Grease Pencil
Storyboarding in 3D Space
Storyboard with Blender's Grease Pencil: tips for your short film
Blender 2D Animation Basics for Beginners - Grease pencil guide
31-Day Grease Pencil Challenge in Blender

-

Production

How Ian Hubert Hacked VFX - Delighting
A Guide on Achieving Flawless Bakes