

Blender - Python API

#1

Serdar ARITAN

Department of Computer Graphics
Hacettepe University, Ankara, Turkey





Blender / Python API

Blender is a free 3D graphics package

- 3D object modelling
- Materials and lighting
- Animation and simulation
- Video editing and compositing
- Integrated physics engine
- Free download, installed in labs

Blender can be scripted with **Python**

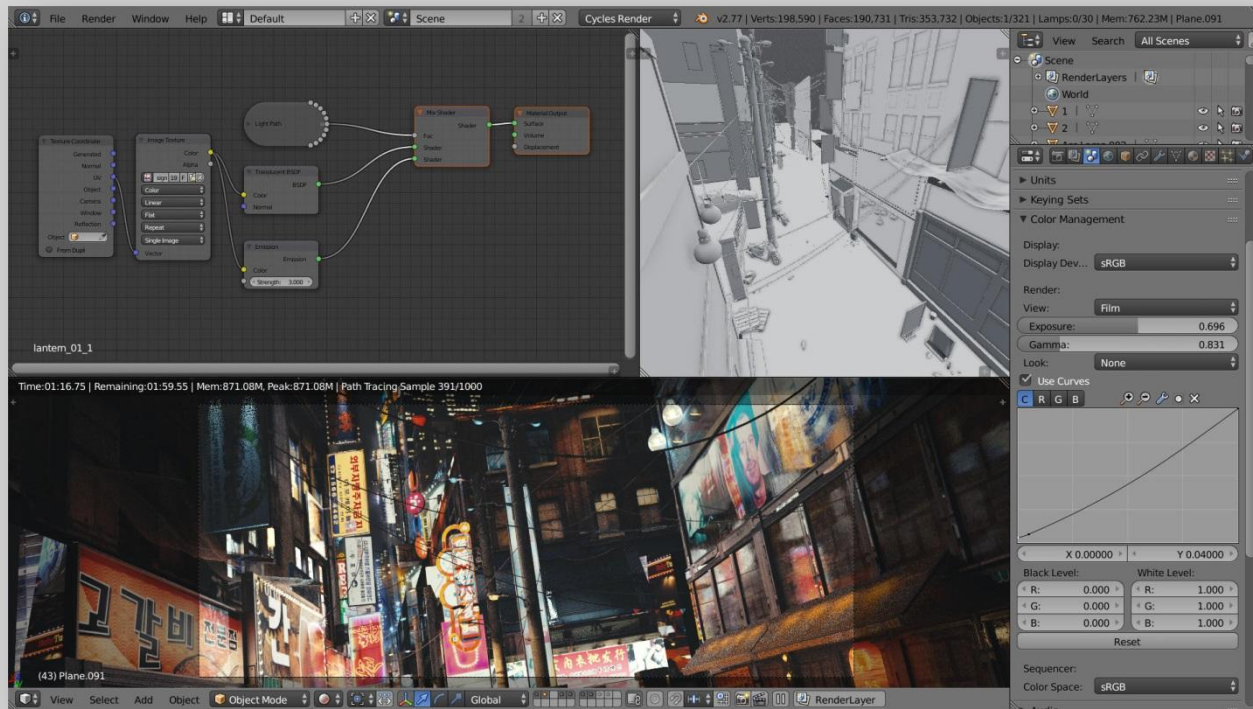
- Uses Python 3
- Just about everything can be scripted
- Creating objects, lights, cameras
- Creating and applying materials
- Rendering and animation
- The Blender interface itself



Blender / Cycles Render Engine

Cycles is Blender's built-in unbiased path-tracer engine that offers stunning ultra-realistic rendering.

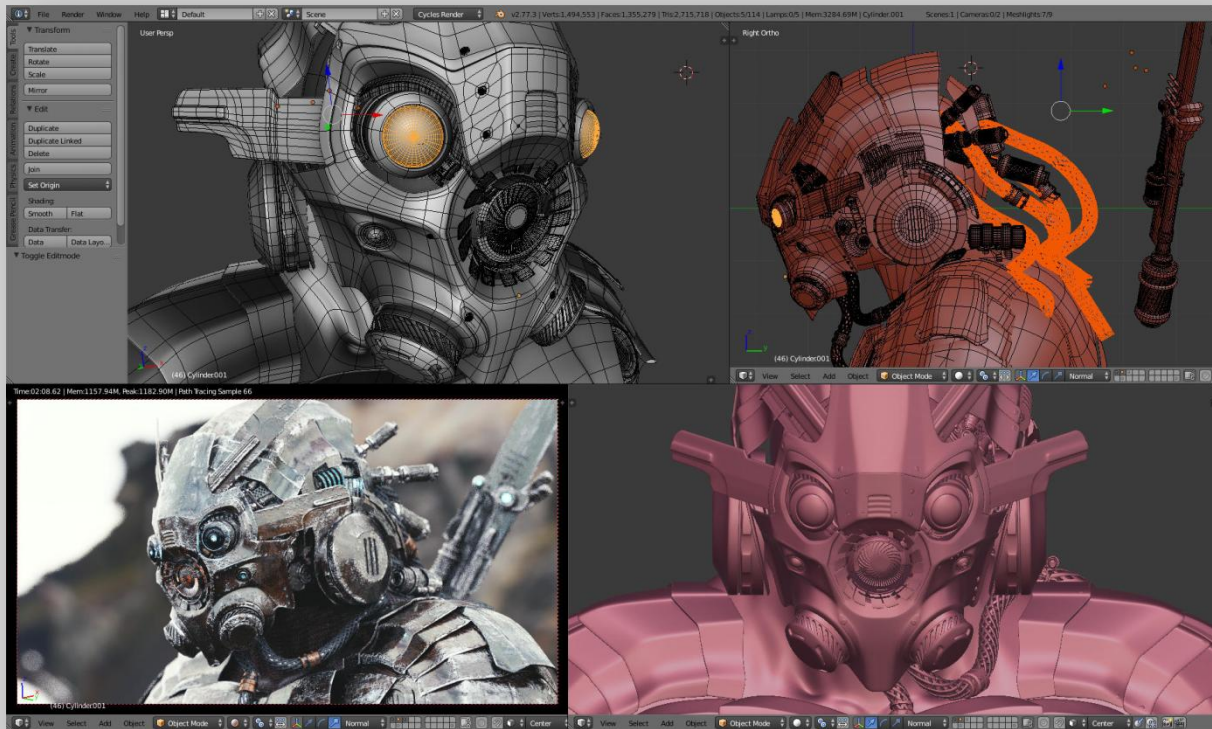
- Real-time viewport preview
- CPU & GPU rendering
- PBR shaders & HDR lighting support
- VR rendering support



Blender / Modeling, Sculpt, UV

Blender's comprehensive array of modeling tools.

- Advanced sculpting tools and brushes
- Multi-resolution and Dynamic subdivision
- 3D painting with textured brushes and masking
- Python scripting for custom tools and add-ons





Blender / Animation & Rigging

- Envelope, skeleton and automatic skinning
- B-spline interpolated bones
- Curve editor and dope sheets
- Custom bone shapes for fast input
- Sound synchronization



Blender / Animation & Rigging

Blender has a flexible Python controlled interface. Layout, colors, size and even fonts can be adjusted.

- Customize the interface layout and colors
- Hi-res/Retina screen support
- Create your own tools and add-ons
- Draw over the OpenGL viewport
- Connect with Blender's Render API





SHOWREEL 2013

Course Outline

- **The Blender Interface**
- **The `bpy` Module**
- **The `bmesh` Module**
- **Topics in Modeling and Rendering**
- **Introduction to Add-On Development**
- **The `bg1` and `b1f` modules**
- **Advanced Add-On Development**
- **Textures and Rendering**



The Blender Interface

The Default Blender Interface

- 3D Viewport
- Header Menu
- Properties Window
- Tool Shelf and Tool Properties
- Timeline

The Scripting Interface

- Text Editor
- Command Log
- Interactive Console

Customizing the Interface

Starting Blender from the Command Line (for Debugging)

Running Our First Python Script

- Finding the Function
- Testing the Function
- Writing the Script



The `bpy` Module

`bpy.ops`

`bpy.context`

`bpy.data`

`bpy.app`

`bpy.types`, `bpy.utils`, and `bpy.props`

`bpy.path`

Selection, Activation, and Specification

- Selecting an Object
- Activating an Object
- Specifying an Object (Accessing by Name)

Pseudo-Circular Referencing and Abstraction

Transformations with `bpy`

Visualizing Multivariate Data with the Minimal Toolkit

- Visualizing Three Dimensions of Data
- Visualizing Four Dimensions of Data
- Visualizing Five Dimensions of Data



The `bmesh` Module

Edit Mode

- Selecting Vertices, Edges, and Planes
- Switching Between Edit and Object Modes Consistently
- Instantiating a `bmesh` Object
- Selecting Parts of a 3D Object

Edit Mode Transformations

- Basic Transformations
- Advanced Transformations

Note on Indexing and Cross-Compatibility

Global and Local Coordinates

Selecting Vertices, Edges, and Faces by Location

Checkpoint and Examples

Topics in Modeling and Rendering

Specifying a 3D Model

- Specifying Meshes and Textures

Common File Formats

- Wavefront (**.obj** and **.mtl**), **STL** (STereoLithography), **PLY** (Polygon File Format)
- Blender (**.blend**) Files and Interchange Formats

Minimal Specification of Basic Objects

- Definition of a Cube
- Naive Specification
- Using Indices to Share Vertices and Normals
- Using Coplanar Vertices to Reduce Face Count
- Using Face Vertices to Simplify Indices
- Representing a Cube as a Primitive

Common Errors in Procedural Generation

- Concentric Normals
- Flipped Normals
- Z-Fighting



Introduction to Add-On Development

A Simple Add-On Template

Components of Blender Add-Ons

- The `b1_info` Dictionary
- Operators and Class Inheritance (`bpy.types.Operator`)
- Panels and Class Inheritance (`bpy.types.Panel`)
- `Register()` and `Unregister()`
- Scene Properties and `bpy.props`

Precision Selection Add-On Example

- Code Overview for Our Add-On
- The `poll()` Classmethod
- `EnumProperty` Variables
- Preparing Our Add-On for Distribution



The **bg1** and **blf** Module

Instantaneous Drawing

Handlers Overview

- Clock Example
- Managing Handlers
- Types of Handlers
- Persistent Handlers

Handlers in **blf** and **bg1**

Example Add-On

- Drawing Lines and Text
- Declaring Button-Activated Drawing Functions
- Declare Main Drawing Function
- Declaring the Operator with Handlers
- Declaring the Panel with Dynamic Drawing

Extending our **bg1** and **blf** Template



Advanced Add-On Development

Developing in Blender's Filesystem

- Creating an Add-on in the Filesystem
- Using **F8** to Reload Add-Ons
- Important Takeaway
- Managing Imports

IDEs for In-Filesystem Development

- Lightweight (Notepad++ and Vim)
- Midweight (Sublime Text, Atom, and Spyder)
- Heavyweight (Eclipse PyDev, PyCharm, and NetBeans)
- Compiling Blender as a Python Module

Best Practices for External Data

- Using File Interchange Formats
- Using Hardcoded Python Variables
- Algorithmic Manipulation of Primitives Summary

Advanced Panel Creation

- Panel Organization and Icons



Textures and Rendering

Vocabulary of Textures

- Types of Influence in Blender
- Types of Textures in Blender

Adding and Configuring Textures

- Loading Textures and Generating UV Mappings
- Textures Versus Materials in Blender
- UV Coordinates and Loops

Removing Unused Textures and Materials

Rendering Using Blender Render

- Adding Lights
- Adding Cameras
- Rendering an Image



The Blender/Python API can do the following:

Edit any data the user interface can (Scenes, Meshes, Particles etc.)

Modify user preferences, keymaps and themes

Run tools with own settings

Create user interface elements such as menus, headers and panels

Create new tools

Create interactive tools

Create new rendering engines that integrate with Blender

Define new settings in existing Blender data

Draw in the 3D view using OpenGL commands from Python



The Blender/Python API can't (yet)...

Create new space types.

Assign custom properties to every type.

Define callbacks or listeners to be notified when data is changed.



Note: The Blender Python API has areas which are still in development.

The following areas are subject to change.

- operator behavior, names and arguments
- mesh creation and editing functions

These parts of the API are relatively stable and are unlikely to change significantly

- data API, access to attributes of blender data such as mesh verts, material color, timeline frames and scene objects
- user interface functions for defining buttons, creation of menus, headers, panels
- render engine integration
- modules: bgl, mathutils & game engine





<https://www.blender.org/download/>

♥ Donate

Download >

LTS

Releases

Previous Versions

Requirements

Daily Builds

Demo Files

Benchmark

Extensions

Releases >

4.5 LTS

4.4

4.3

4.2 LTS

4.1

4.0

3.6 LTS

3.5

3.4

3.3 LTS

3.2

3.1

3.0

2.93 LTS

2.83 LTS

Your Best Companion

With 2 years of updates, full Vulkan support, and quality-of-life improvements, Blender 4.5 LTS is every Blenderhead's best friend.

Download Blender 4.5 LTS

Watch Recap Video

EEVEE & VIEWPORT

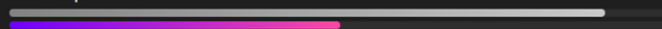
Start Me Up, Fast

Blender 4.5 LTS is noticeably faster thanks to performance improvements such as [faster texture loading](#), quicker [startup times](#), and [multi-threaded shader compilation](#) by default on supported platforms.

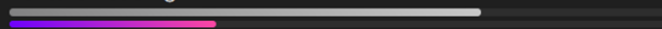
A new [user preference](#) allows choosing between multi-threaded and subprocess compilation.

● Blender 4.5 LTS ● Blender 4.4

Startup Time

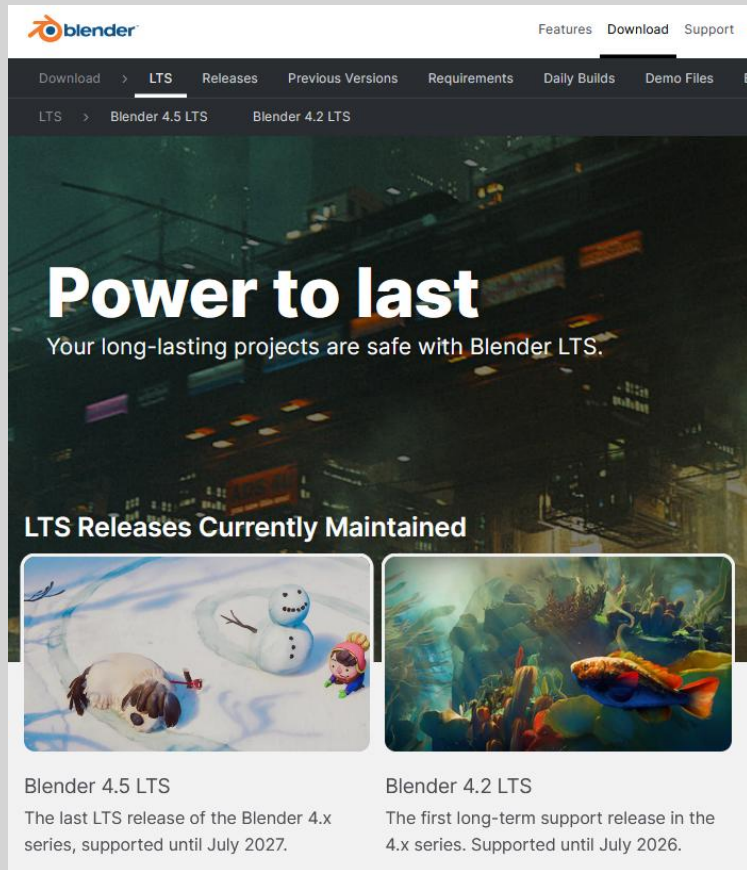


Texture Loading



0 1 2 3 4 5 6 7 8

Lower is better Unit: seconds



The screenshot shows the Blender website's LTS page. At the top, there's a navigation bar with 'Features', 'Download', and 'Support'. Below it, a sub-navigation bar highlights 'LTS' under the 'Download' section, with other options like 'Releases', 'Previous Versions', 'Requirements', 'Daily Builds', and 'Demo Files'. The main heading is 'Power to last' with the subtext 'Your long-lasting projects are safe with Blender LTS.' Below this, a section titled 'LTS Releases Currently Maintained' features two cards. The first card for 'Blender 4.5 LTS' shows a snowman and a penguin in a snowy landscape, with text stating it's the last LTS release of the 4.x series supported until July 2027. The second card for 'Blender 4.2 LTS' shows a colorful koi fish in an aquarium, with text stating it's the first long-term support release in the 4.x series supported until July 2026.

Power to last
Your long-lasting projects are safe with Blender LTS.

LTS Releases Currently Maintained

Blender 4.5 LTS
The last LTS release of the Blender 4.x series, supported until July 2027.

Blender 4.2 LTS
The first long-term support release in the 4.x series. Supported until July 2026.

The Blender LTS program is aimed at ensuring that long-lasting projects can be executed using a stable Blender version, which will provide critical fixes throughout a 2-year time span. The LTS version will not have any new features, API changes or improvements.



SCRIPT LANGUAGES FOR ANIMATION

Blender 3.6.4 LTS – September 26, 2023



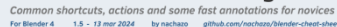
Download

- [Linux](#)
- [macOS – Intel](#)
- [macOS – Apple Silicon](#)
- [Windows – Installer](#)
- [Windows – Portable \(.zip\)](#)





Changelog


- “Jump to keyframe” grayed out. Unable to jump between keyframes. [[#112673](#)]
- Transfer Mode operator doesn’t update outliner selection. [[#101550](#)]
- Python modifiers.execution_time doesn’t work on curve objects. [[#112397](#)]
- Fix [#112399](#): Memory leak with exceptions from scripts in the text editor. [[d9ab0a3ef7f](#)]



Tips & tricks

- 1 Remove duplicated vertices, faces or edges. Use **Merge Vertices** → **By Distance**
Solves many workflow problems with transforms, modifiers...
- 2 Apply scale (**Ctrl + A**) and other transforms (rotation/location) if needed.
Solves many workflow problems with transforms, modifiers...
- 3 Check Normals. Use **Mesh** → **Normals** → **"Recalculate Outside"** (or **Air + N**) for fixing.
Verify it in checkbox: "Viewport Overlays" → Face orientation
- 4 Check Object origin. Correct it in **Object** → **Set Origin** (select geometry in Edit Mode if needed).
Solves many problems with scale, modifiers like Array, Mirror...
- 5 Better use **quads** (4 edges). Avoid **ngons** (+4 edges). Also avoid, if possible, **tris** (3 edges).
- 6 Check **Render** (and other options) icons marked on object/collections in the **Outliner** to your needs.
- 7 Check if "**Snap**"  or "**Proportional editing**"  are activated, causing bad behaviors.
- 8 In Edit Mode, check **Options** → **Transform** → **"Correct Faces Attributes"** if needed.
For correctly transform meshes with applied UV map & textures.

Node Wrangler

-  +  Disconnect/cut nodes
-  +  Create a **Mix** node
Select one or more for auto-mixing
-  +  Auto-connect nodes
 *Connect selected nodes*

More shading nodes

Hue/Saturation/Value & Brightness/Contrast : Change texture properties, like color, contrast...

Blackbody : Node for light temperature color, for example in an Emission surface (1500K to 12000K).


Ambient Occlusion : Add shadow in the object folds.
You can add "dirt" or remark this zones with this setup:



Optimization tips

- 1 File → Clean Up → **Recursive Unused Data-Blocks**
Clean unused materials, textures, images... in the scene.
- 2 Right Click in Collection at Outliner: **Instance to Scene**
Make instances, optimizing the polygon count.
- 3 Use instances (**[Alt] + [D]**) and share materials in objects whenever possible.
- 4 File → External Data → **Automatically Pack Resources**
Activate for pack textures and resources into the .blend file.
- 5 Modifier: **Decimate**
Optimize polygon count in the scene.
- 6 Add-on: Material Utilities → **[Panel] + [Q]** →
Specials → Merge Base Names
Merge duplicated materials used by assets (then, Clean Up).
- 7 Bake textures and reduce model polygon count.
Select blank image and bake type in Render → Bake.
- 8 Use Simplify, "**Persistent Data**" and other options detailed in "**Default file start-up confига**".

Other

-  **W** Walk/ fly navigation shortcut
*Or **o** clement, depending your keyboard region*
-  **in a menu:** Search in opened menu content
-  **Hide**
 **+**
 **Show all hidden**
-  **+**
 **Link/transfer data**
For copy modifiers to multiple selected objects
-  **Free textures & assets:** bit.ly/recursos-3d

Interesting basic free add-ons

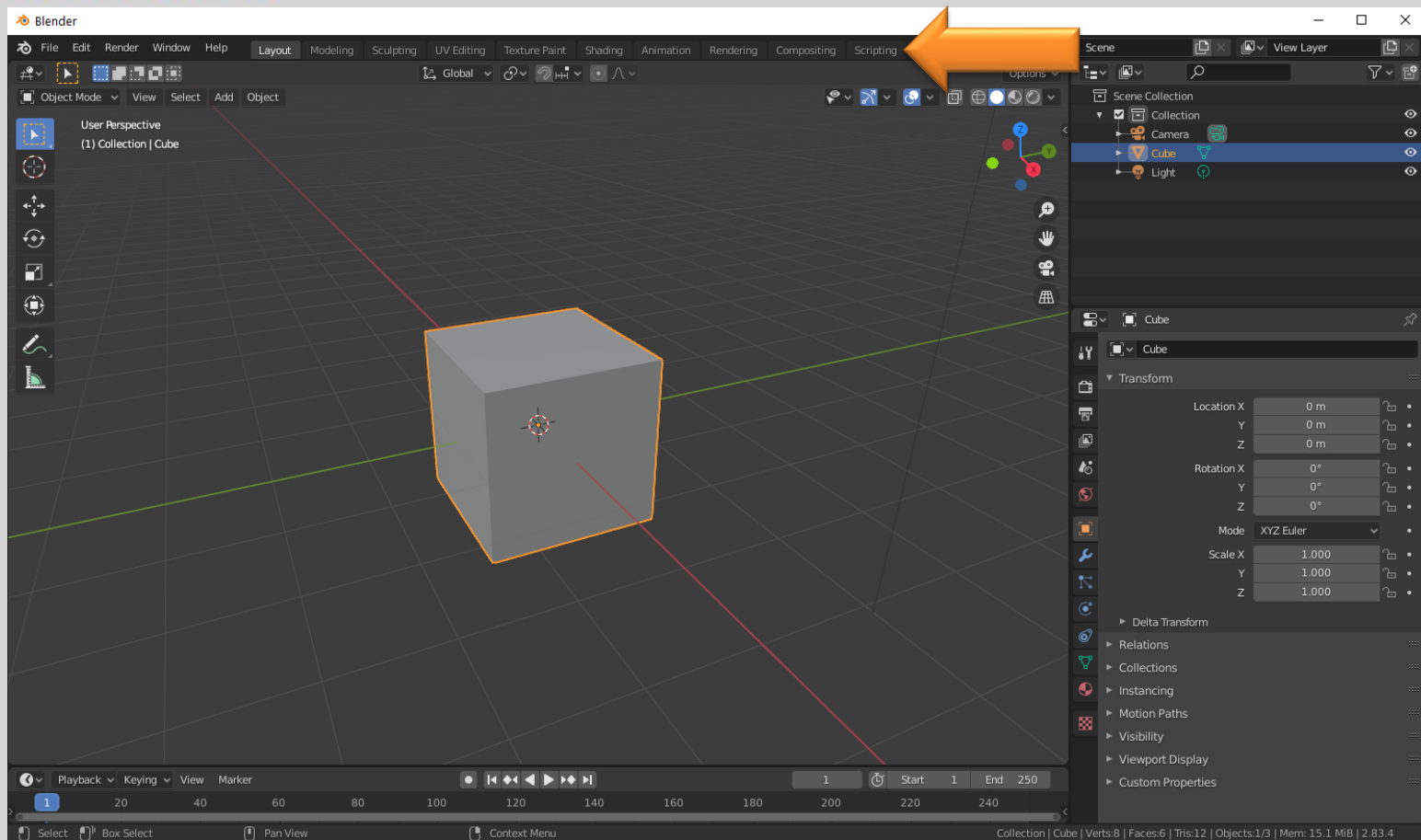
- Node Wrangler Material Utilities Extra Objects Loop Tools
 Bool Tool Images as Planes ANT Landscape Real Snow
 Scatter Objects Copy Attributes Menu **Easy HDR!** *
 UV Squares * BlenderKit * *All in Preferences → Add-ons, except **

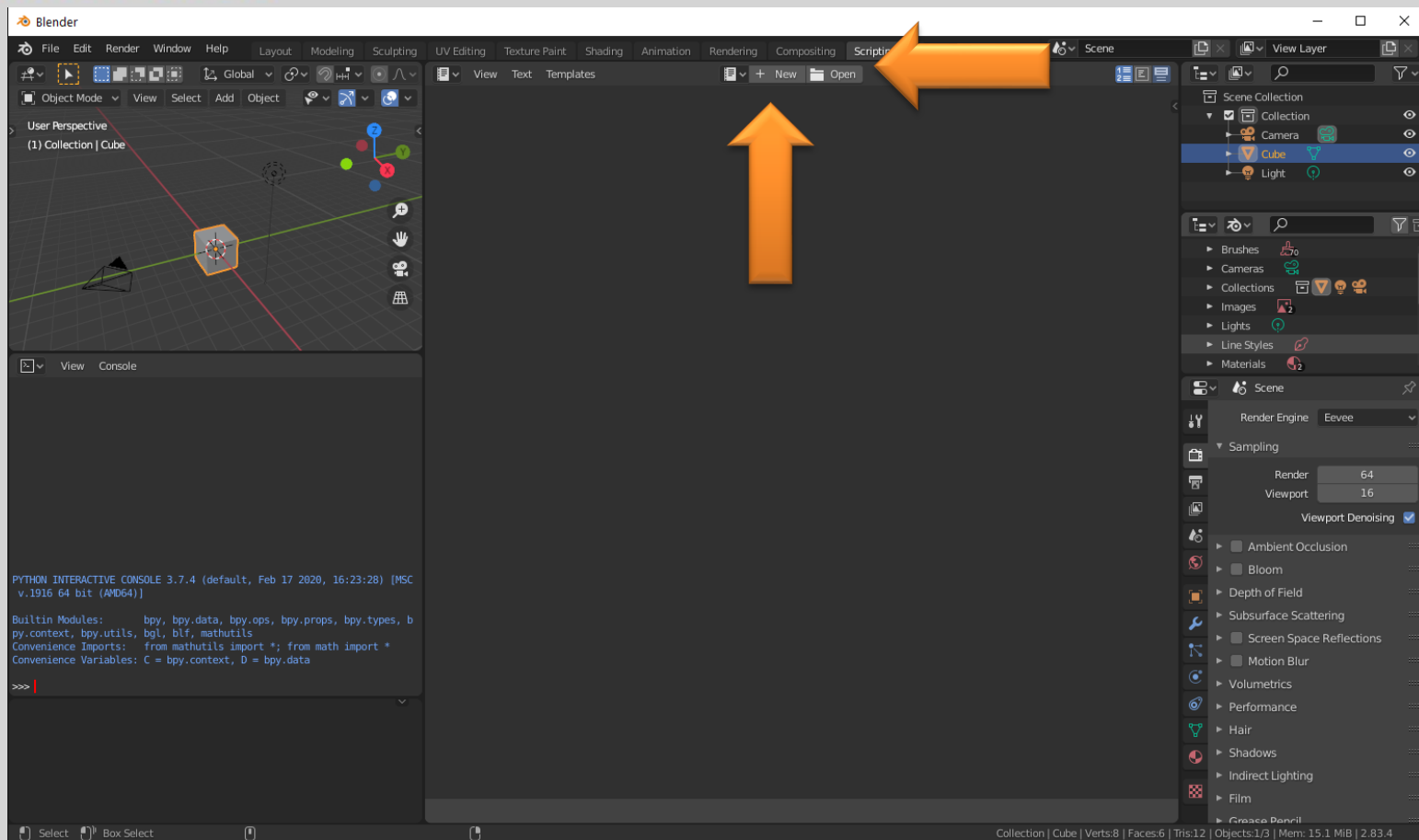
Default file start-up config

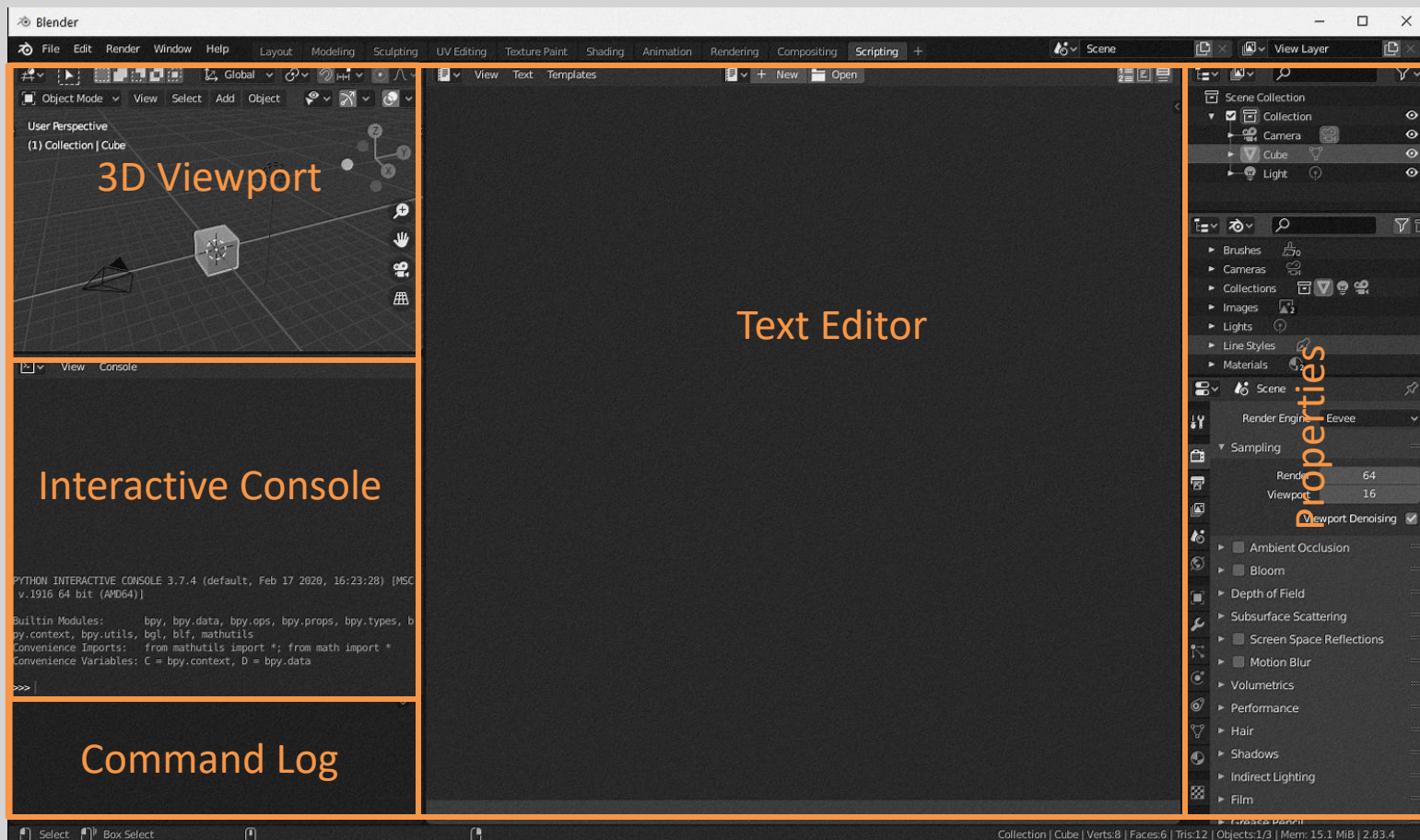
- In **Viewport shading**: *Random & Cavity*.
- In **Output**, check "Render region".
- In **Render**
 - Select **Cycles**. Configure your samples.
 - In **Eevee** config, check "*Ambient Occlusion*".
 - **Simplify**: Configure to your liking/project, usefull.
 - Performance: check "*Persistent Data*".
 - Color Management: Choose to your liking.
- In the **outliner** filters, activate "*Selectable*".
- In **World**: Set "Sky Texture" at 0.130 strength.
- In **Blender preferences**: Up the "Undo steps" number and select **Cycles Render Engine** if possible.

- Geometry Nodes [Distribute Points on Faces](#) [Instance On Points](#) [Join Geometry](#) [Random Value](#) [Object Info](#) [Weight Paint](#)











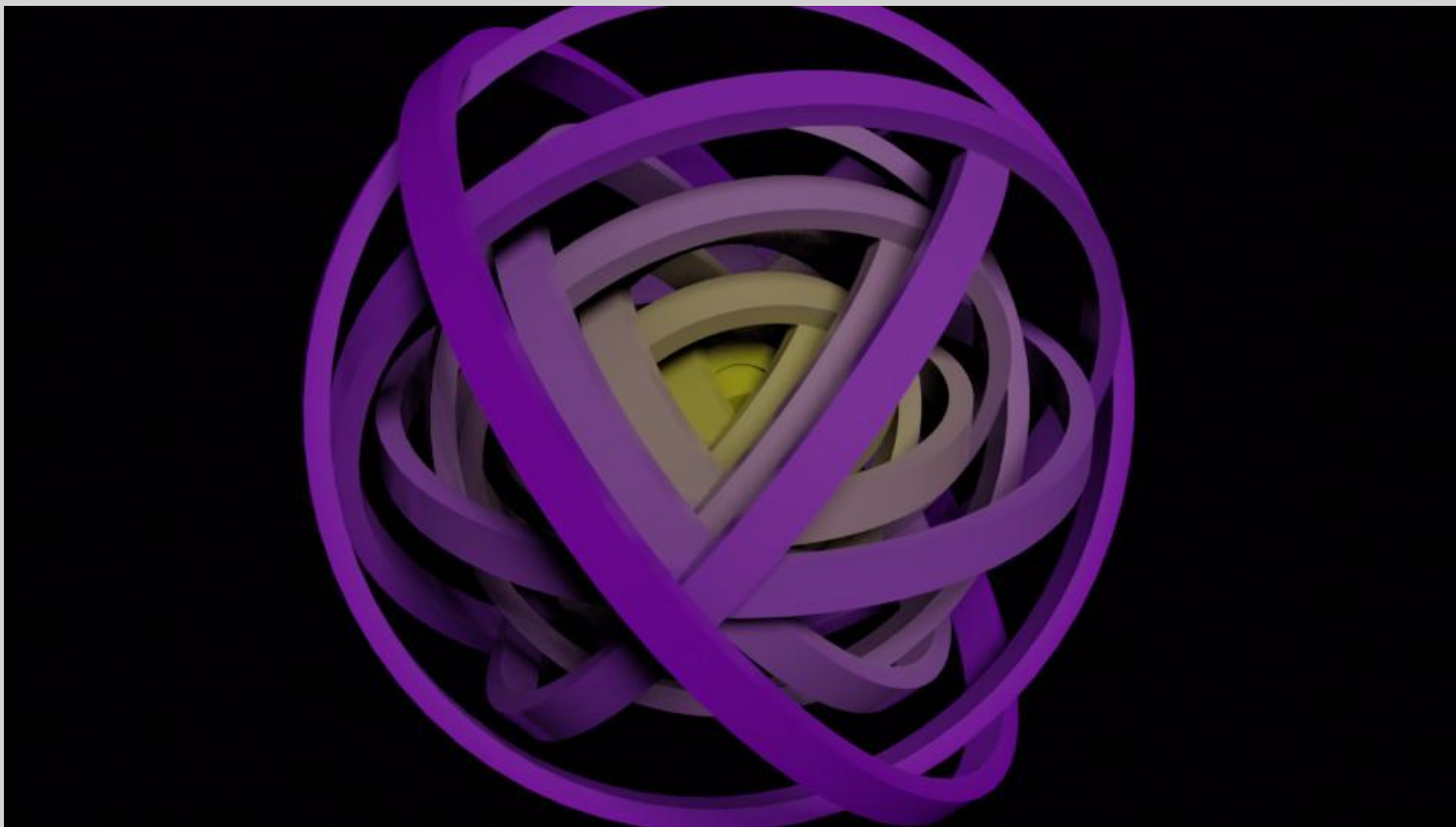
<https://www.blender.org/download/its/>

[**https://docs.blender.org/api/blender_python_api_current/**](https://docs.blender.org/api/blender_python_api_current/)

[**http://blender.stackexchange.com/**](http://blender.stackexchange.com/)

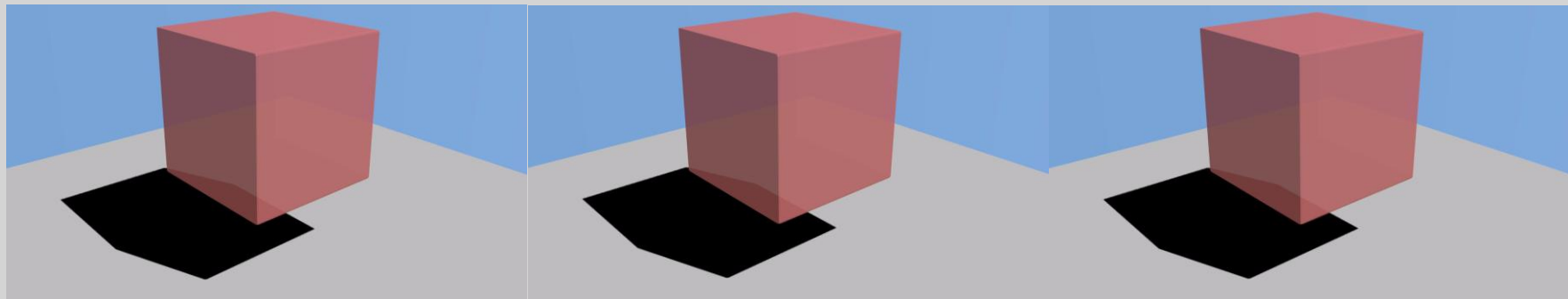


Here are some simple examples we will do



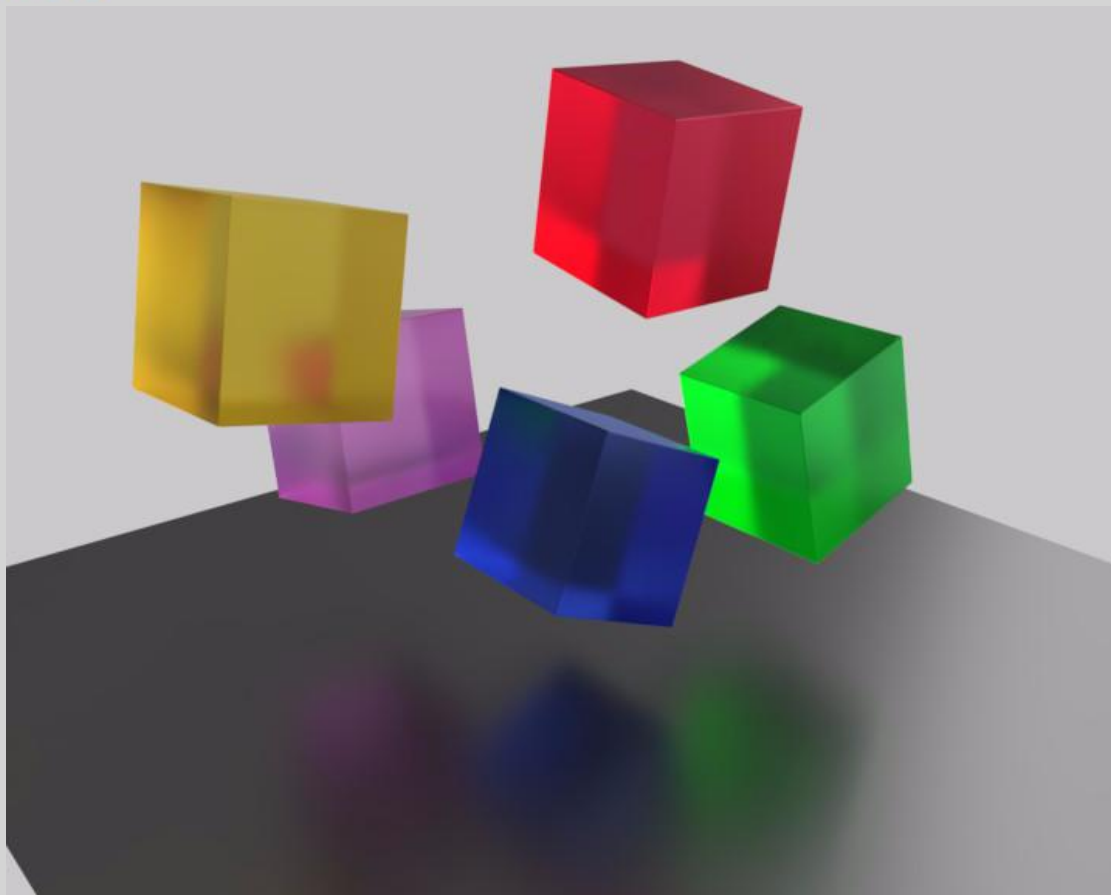


Here are some simple examples we will do



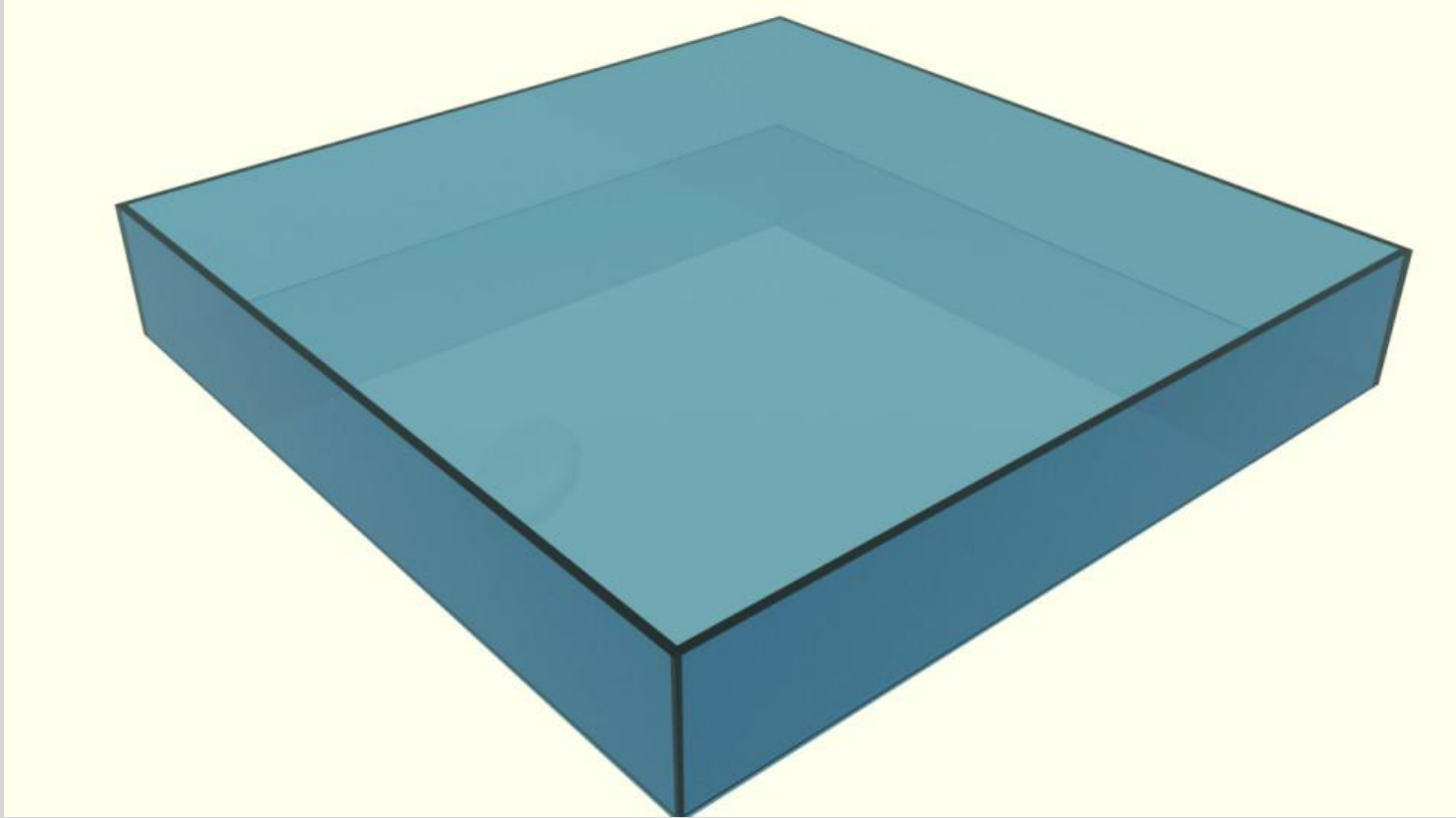


Here are some simple examples we will do



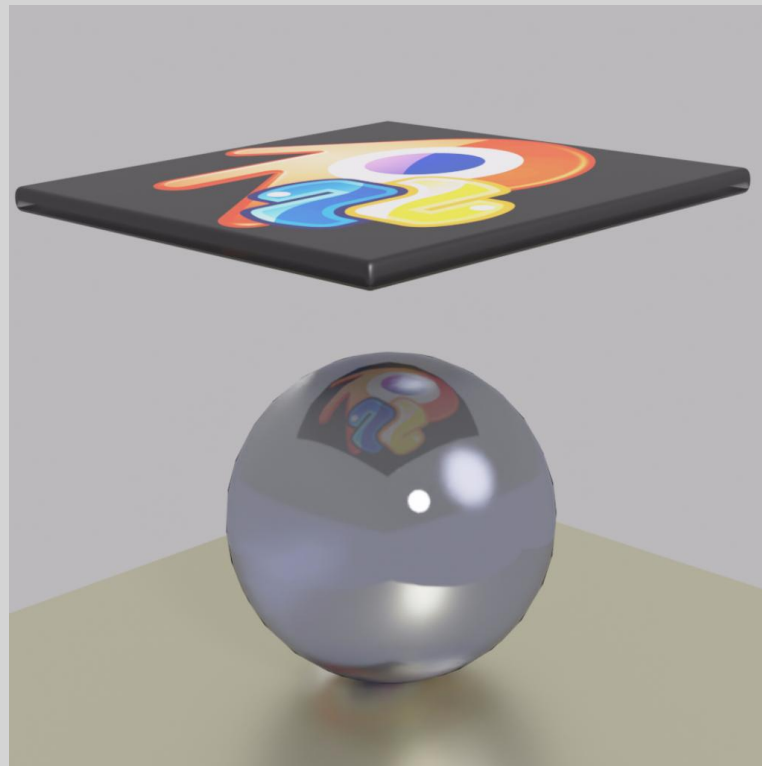
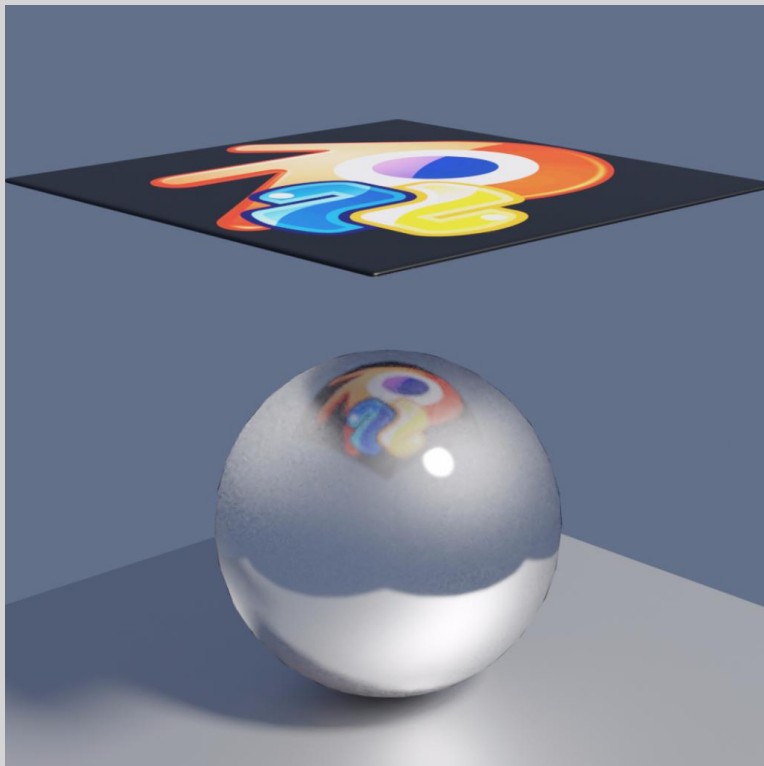


Here are some simple examples we will do



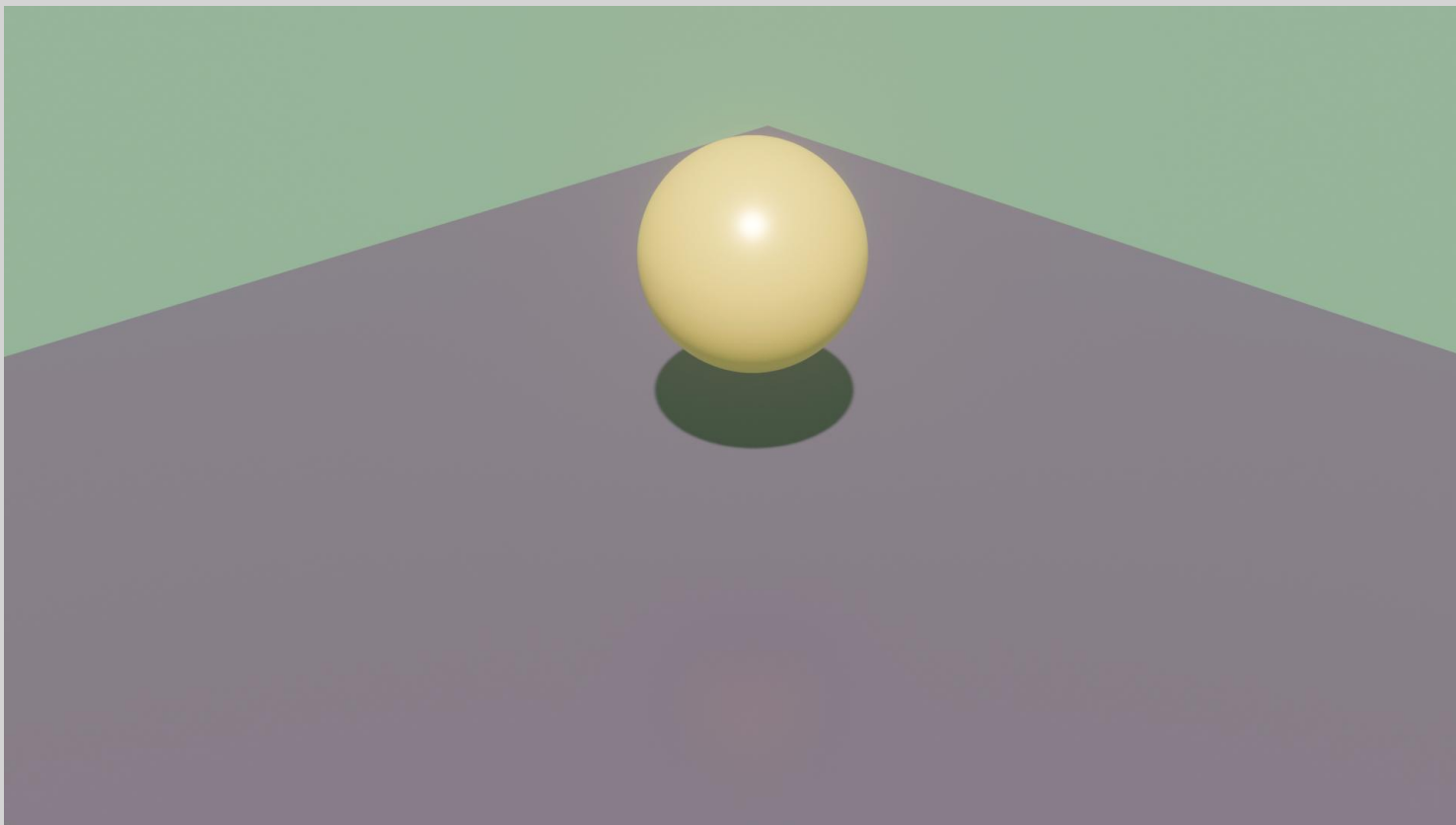


Here are some simple examples we will do





Here are some simple examples we will do





Google search results for "serdar aritan".

Hacettepe Üniversitesi
<https://yunus.hacettepe.edu.tr/~ser...>

You visit often

Serdar Aritan'nın Web Sayfası
Serdar ARITAN. Biyomekanik Laboratuvarı Koordinatörü. 1989 yılında Yıldız Teknik Üniversitesi'n den Endüstri Mühendisi olarak mezun oldum.

Hacettepe Üniversitesi
<https://avesis.hacettepe.edu.tr/~ser...>

Dr. Öğr. Üyesi SERDAR ARITAN | AVESİS
1991 yılında göreve başladığım Hacettepe Üniversitesi'nde, 1999 yılından beri öğretim üyesi olarak çalışmaktayım. Ayrıca Biyomekanik Laboratuvarının araştırma ...

LinkedIn · Serdar ARITAN
590+ followers

Serdar ARITAN - Faculty Member at Hacettepe University
Ankara, Ankara, Türkiye · Faculty Member · Hacettepe University
Faculty Member at Hacettepe University · My research interests include mechanical modelling of soft tissue, human motion analysis and simulation, ...

Google Scholar
<https://scholar.google.com/citations>

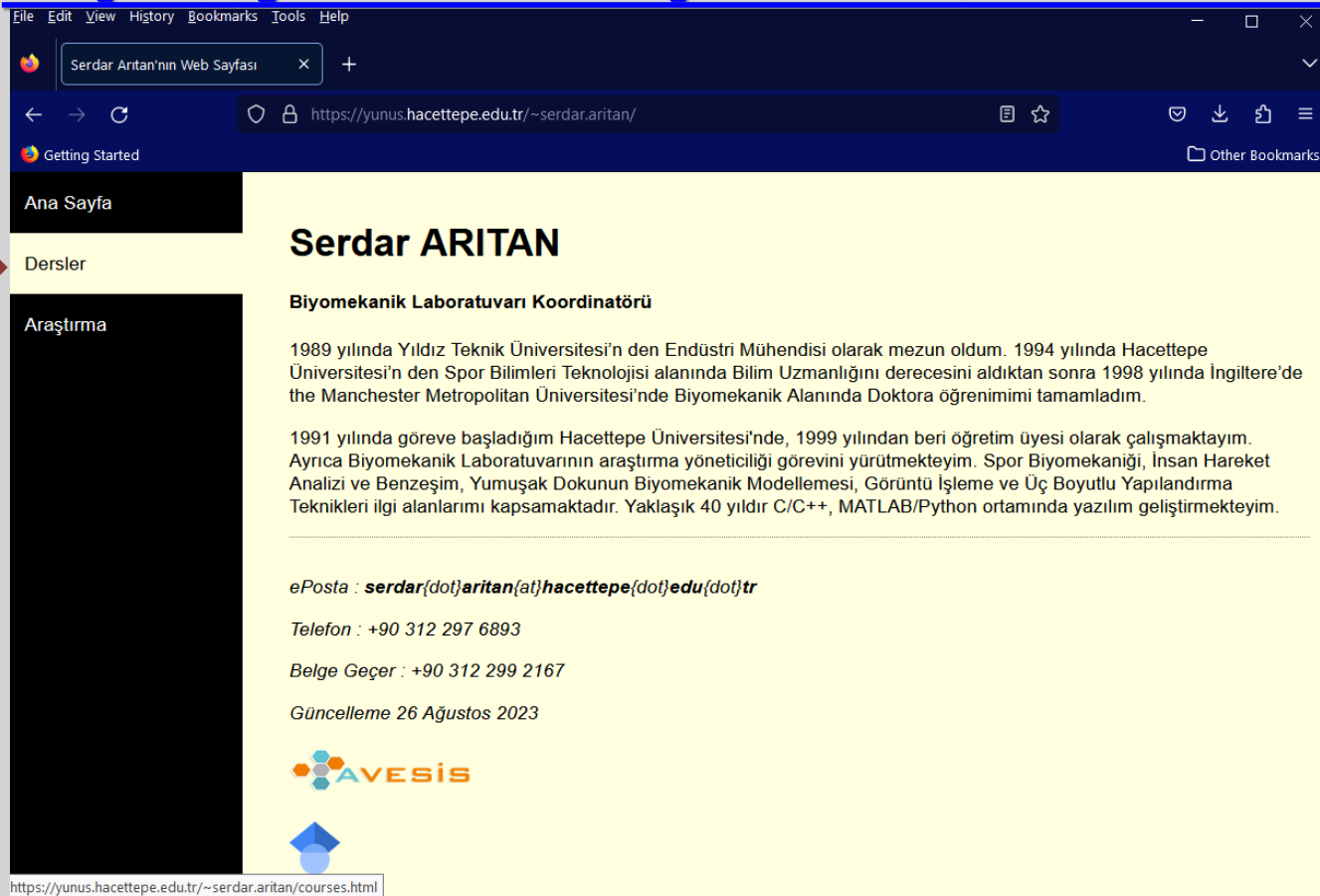
Serdar Aritan
Faculty Member, Hacettepe University · Cited by 765 - Biomechanics - Sports Biomechanics - Simulation - Computer Graphics

ORCID
<https://orcid.org/~serdar-aritan>

Serdar Aritan (0000-0001-6430-3156)
Works (31) · Investigation the effect of rigid taping on knee and hip joint kinematics in chronic stroke patients with knee hyperextension gait. · Comparison of ...



<https://yunus.hacettepe.edu.tr/~serdar.aritan/>



File Edit View History Bookmarks Tools Help

Serdar Arıtan'nın Web Sayfası

← → ↻

Getting Started

Ana Sayfa

Dersler

Araştırma

Serdar ARITAN

Biyomekanik Laboratuvarı Koordinatörü

1989 yılında Yıldız Teknik Üniversitesi'nin Endüstri Mühendisi olarak mezun oldum. 1994 yılında Hacettepe Üniversitesi'nin Spor Bilimleri Teknolojisi alanında Bilim Uzmanlığını derecesini aldıktan sonra 1998 yılında İngiltere'de the Manchester Metropolitan Üniversitesi'nde Biyomekanik Alanında Doktora öğrenimimi tamamladım.


1991 yılında göreve başladığım Hacettepe Üniversitesi'nde, 1999 yılından beri öğretim üyesi olarak çalışmaktayım. Ayrıca Biyomekanik Laboratuvarının araştırma yöneticiliği görevini yürütmekteyim. Spor Biyomekaniği, İnsan Hareket Analizi ve Benzeşim, Yumuşak Dokunun Biyomekanik Modellemesi, Görüntü İşleme ve Üç Boyutlu Yapılandırma Teknikleri ilgi alanlarımı kapsamaktadır. Yaklaşık 40 yıldır C/C++, MATLAB/Python ortamında yazılım geliştirmekteyim.


ePosta : [serdar{dot}aritan{at}hacettepe{dot}edu{dot}tr](mailto:serdar.aritan@hacettepe.edu.tr)

Telefon : +90 312 297 6893

Belge Geçer : +90 312 299 2167

Güncelleme 26 Ağustos 2023

 AVESIS



<https://yunus.hacettepe.edu.tr/~serdar.aritan/courses.html>



Ana Sayfa

Dersler

Araştırma

Serdar ARITAN

2025-2026 Güz Yarıyılı (22 Eylül 2025 / 28 Aralık 2025)

- Bilişim Enstitüsü Lisansüstü Dersler

BCA 603 Hareket Analizi Sistemleri (Tezli Program)

BCO 601 Python Programlama

BCO 602 Animasyon İçin Betik Diller



Sağlık Bilimleri Enstitüsü Lisansüstü Dersler

HAB 619 Spor Bilimlerinde Bilimsel Programlamaya Giriş

HAB 718 Spor Biyomekaniğinde Hareket Analizi

Genel sınavlar (5 Ocak 2026 / 16 Şubat 2026)

- Lisans Dersleri

ANR305-2 Maç ve Yarışma Analizi [Şube 2 Bireysel Sporlar için Hareket Analizi]

ANR 413 Bilgisayar Programlama

Genel sınavlar (5 Ocak 2026 / 18 Ocak 2026)

Bütünleme sınavları (26 Ocak 2025 / 4 Şubat 2025)



<https://yunus.hacettepe.edu.tr/~serdar.aritan/>

Ana Sayfa

Dersler

Araştırma

BCO 602 Animasyon İçin Betik Diller

Tartışılacak işlenecek konular

1. **Hafta** : Ders tanıtımı, Blender yazılımının kurulması ve Blender arayüzü.
2. **Hafta** : Blender Arayüzü, BPY modülü, Etkileşimli kullanım
3. **Hafta** : BPY modülü, dosyadan veri okuma, Visualizing Three Dimensions of Data
4. **Hafta** : Blender Mesh Definition, Blender & NumPy, BMESH modülü
5. **Hafta** : BMESH modülü, Işık Kullanımı
6. **Hafta** : Kamera Ayarları, Render(Görüntü Alma) ve Nesne Takibi
7. **Hafta** : Işık Uygulaması, Gamma düzeltmesi, Handlers, Keyframing, Text Uygulaması
8. **Hafta** : Blender Fizik : Katı cisim uygulamaları
9. **Hafta** : Blender Fizik : Çok gövdeli katı cisim uygulamaları, eklemler. **Ara Sınav I**
10. **Hafta** : Blender Fizik : Şekil değiştirebilir cisim (yay-damper-cisim) uygulaması. Sonlu Elemanlar Analizi nedir.**Ara Sınav II**
11. **Hafta** : Blender Fizik : Parçacık(Particles) ile sıvı, saç benzeşimi, Kuvvet Alanı.
12. **Hafta** : Blender Fizik : Sıvı ve Kumaş modelleme
13. **Hafta** : Texture, Blender Node, Blender için Add-On yazılması

Genel Sınav : Genel Sınav için Proje Çalışması

Python Kaynak