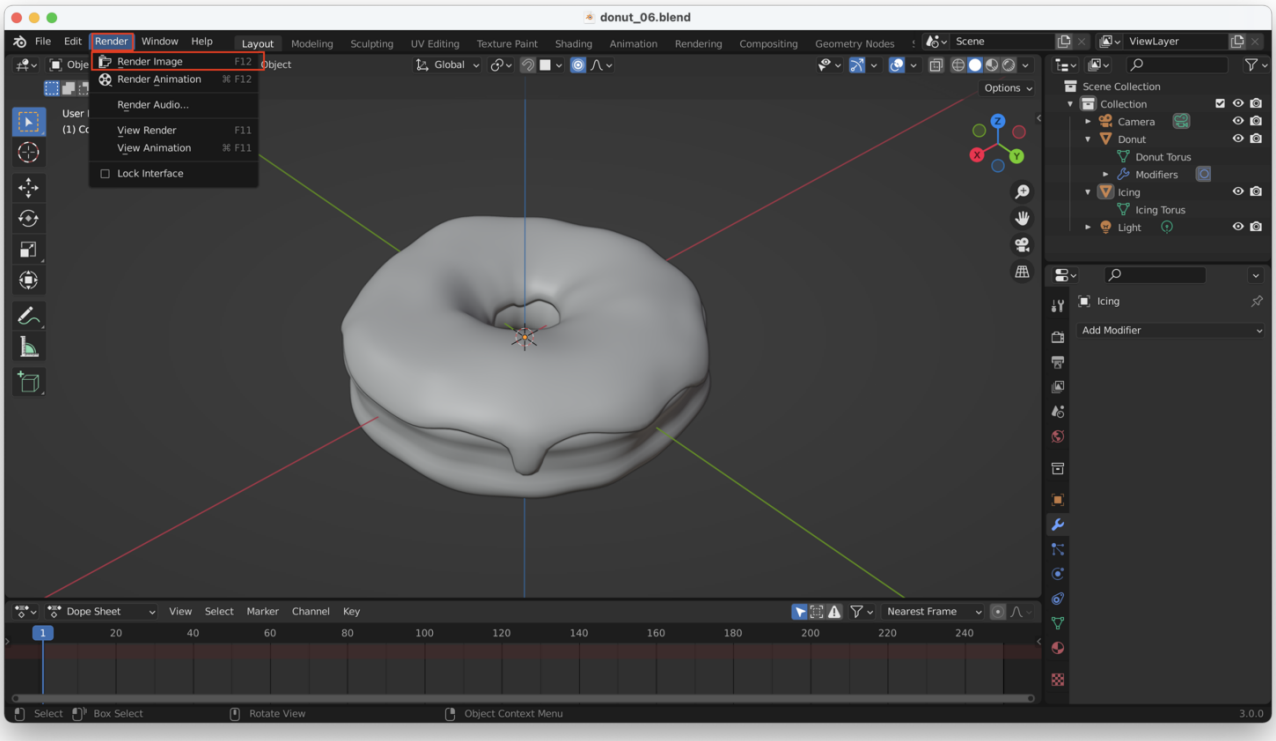
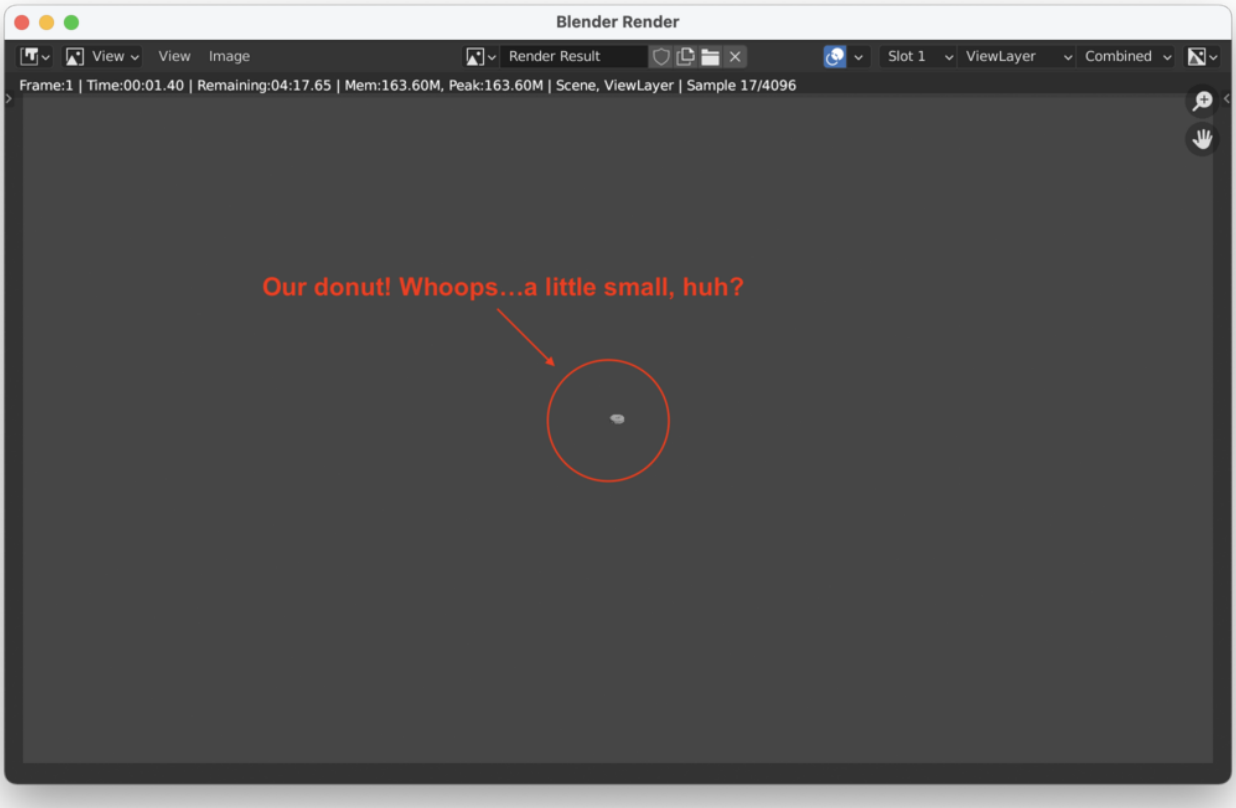
Rendering

Choosing "Render → Render Image" from the top menu bar in Blender will produce a "snapshot" of our 3D scene, from the camera's perspective:

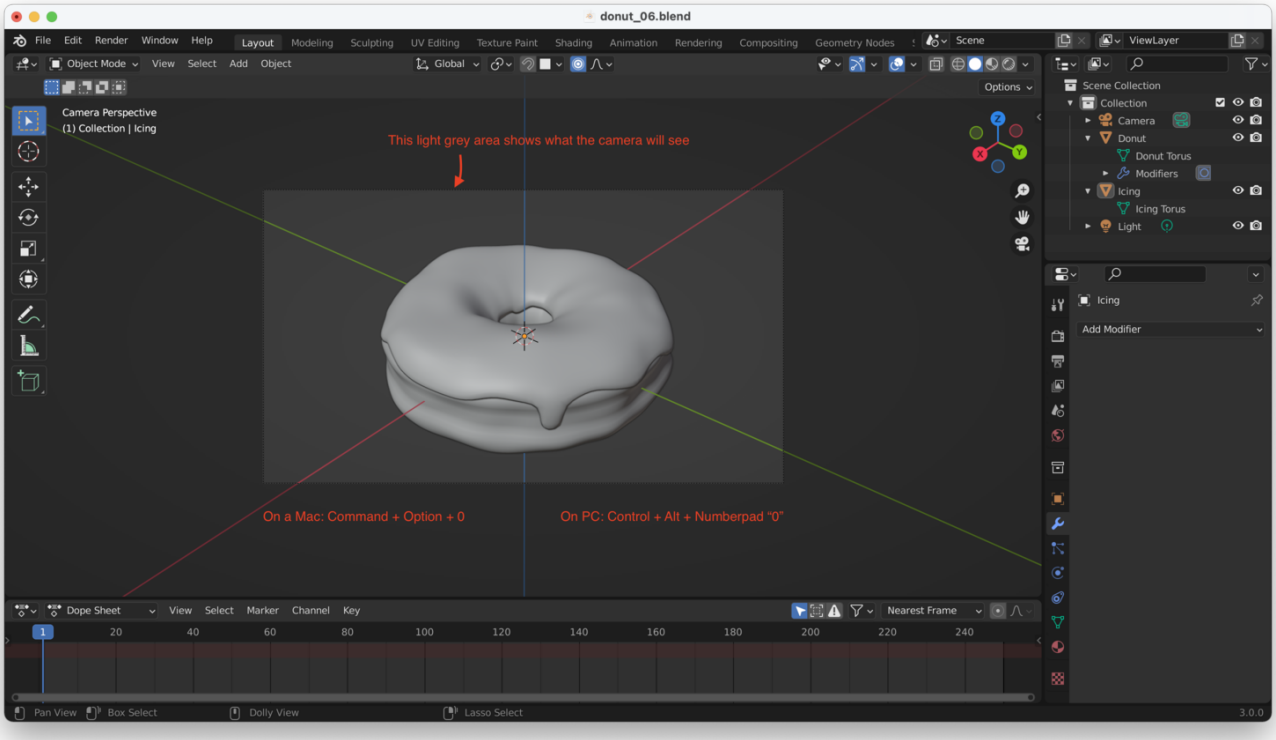




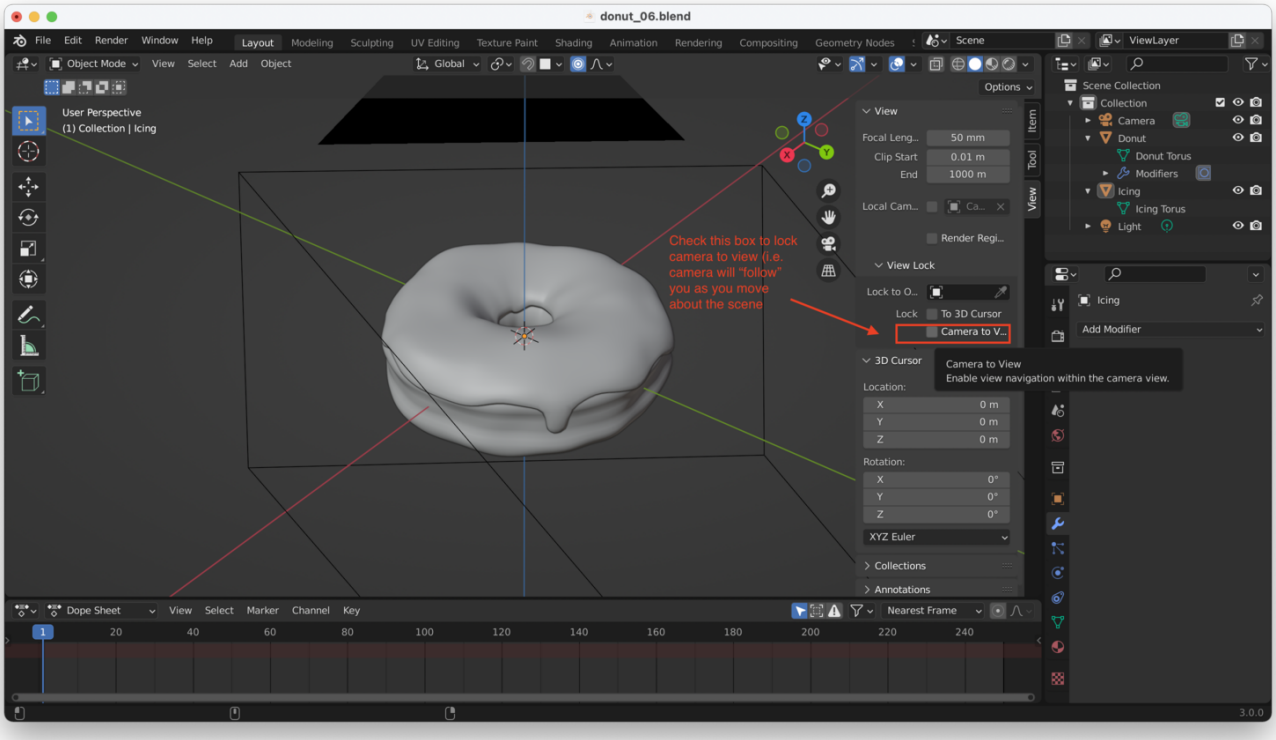
> **Tip:** You can also render the scene with "F12". On newer Macs, you can hold down the "Fn" key then tap on "F12" on the smart bar.

Moving The Camera

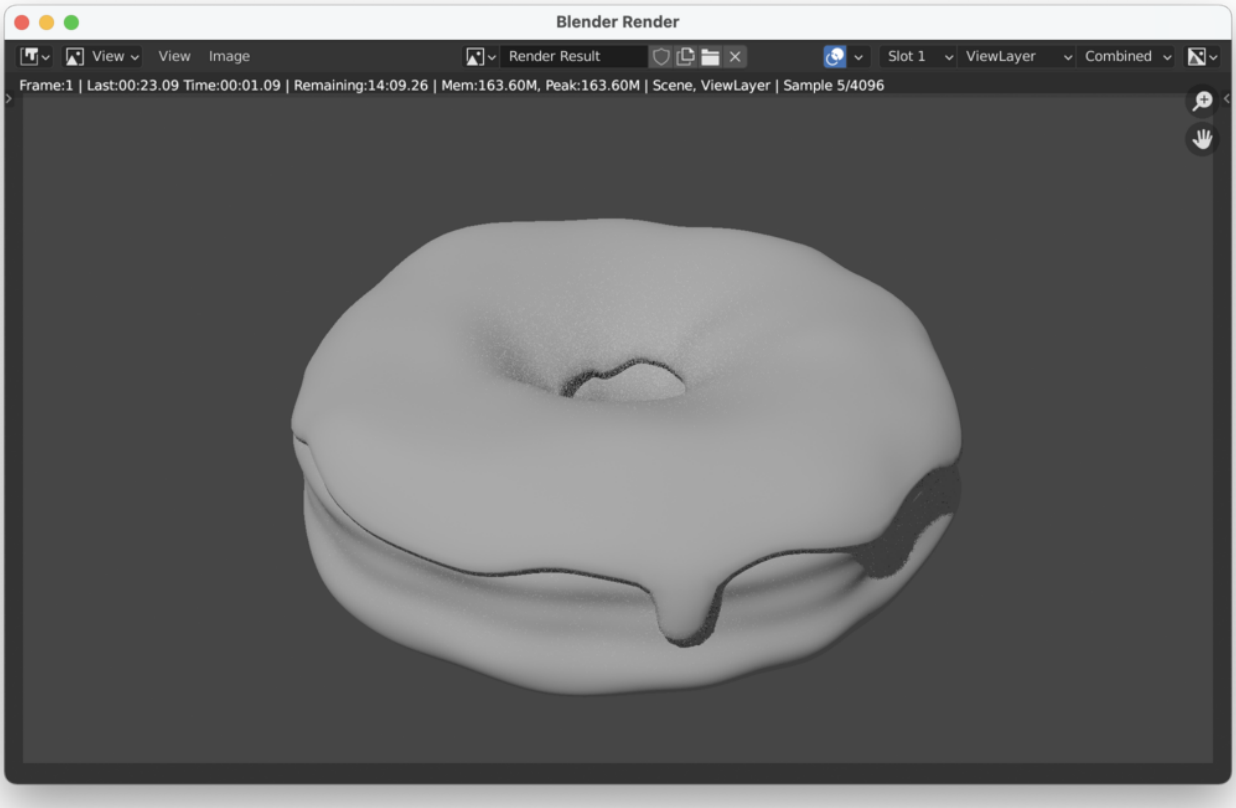
There are a few ways to move the camera. One of the easiest ways is to adjust your viewing angle and position from the Layout window, then hit **"Control + Alt + Number pad 0"**.



An alternative to that is to hit "N" to bring up properties, then under "View" choose "Camera to View", which will lock your camera to your current view (i.e. the camera moves with you):



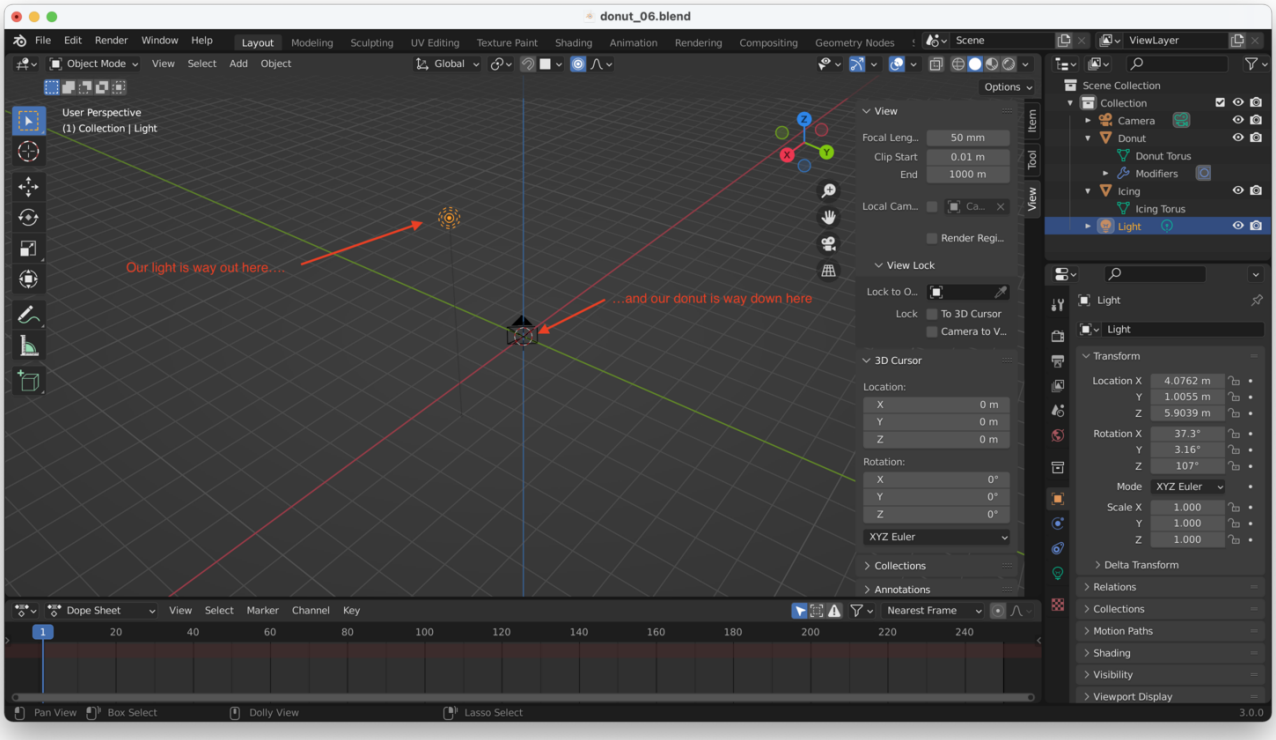
If we now render again, we should see something like this (assuming we moved closer to the donut before repositioning the camera with one of the two techniques above):

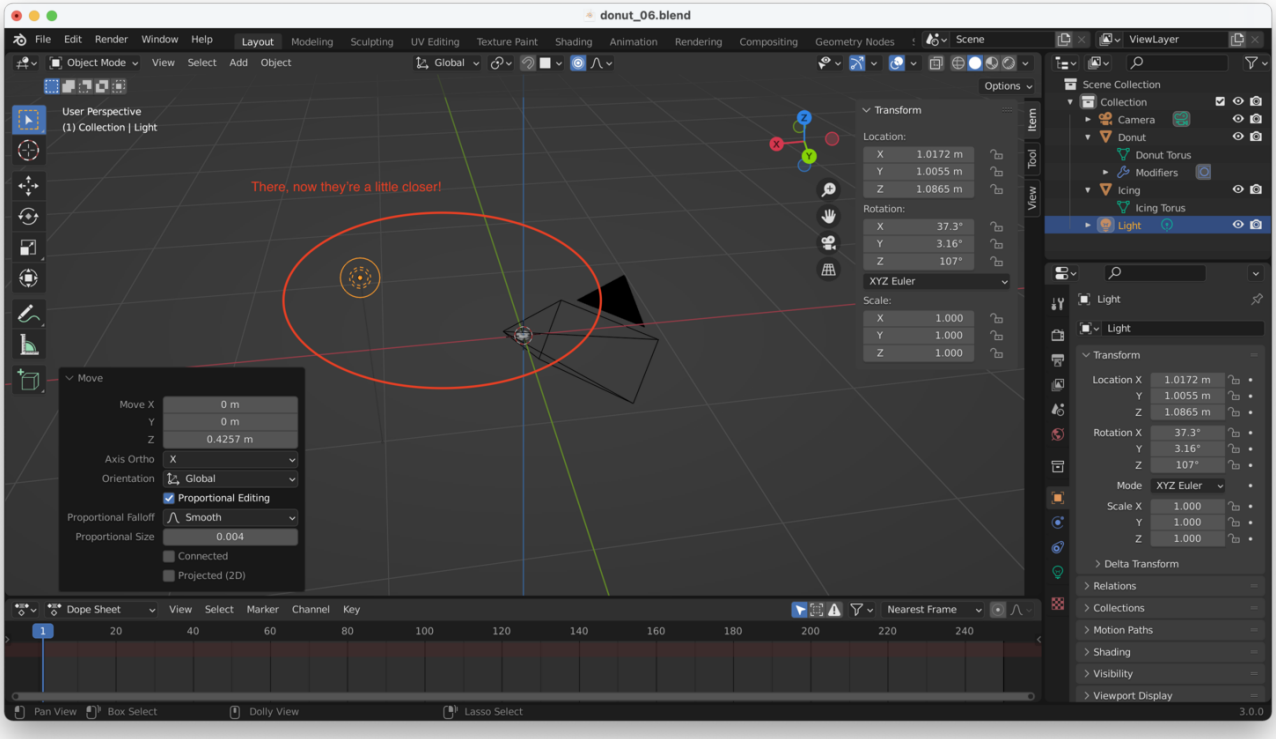


> **Tip**: You can grab the camera and light(s) just like any other object, with "G", then move them around. So that's another way to reposition the camera!

Moving The Light

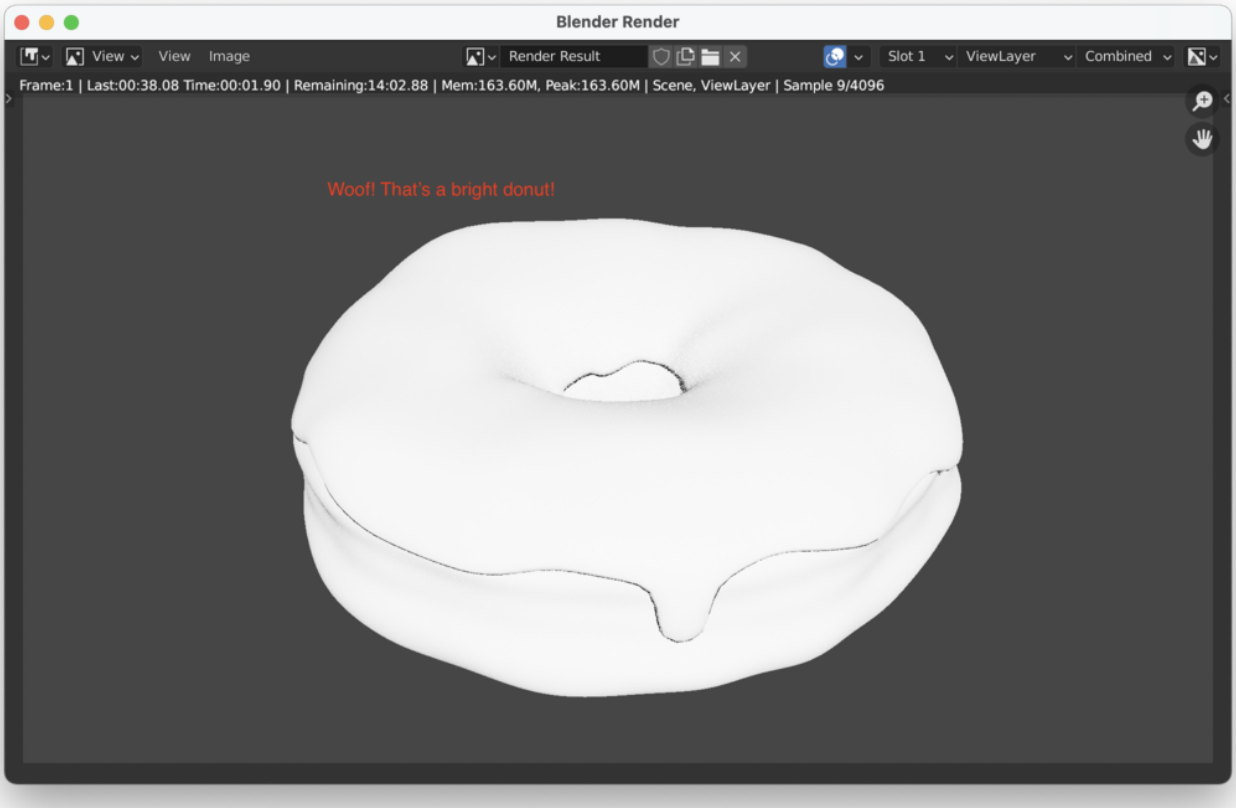
We haven't moved our light yet, either. We should move it in a little, since the default position assumes a **much** bigger scene than the one we are creating:



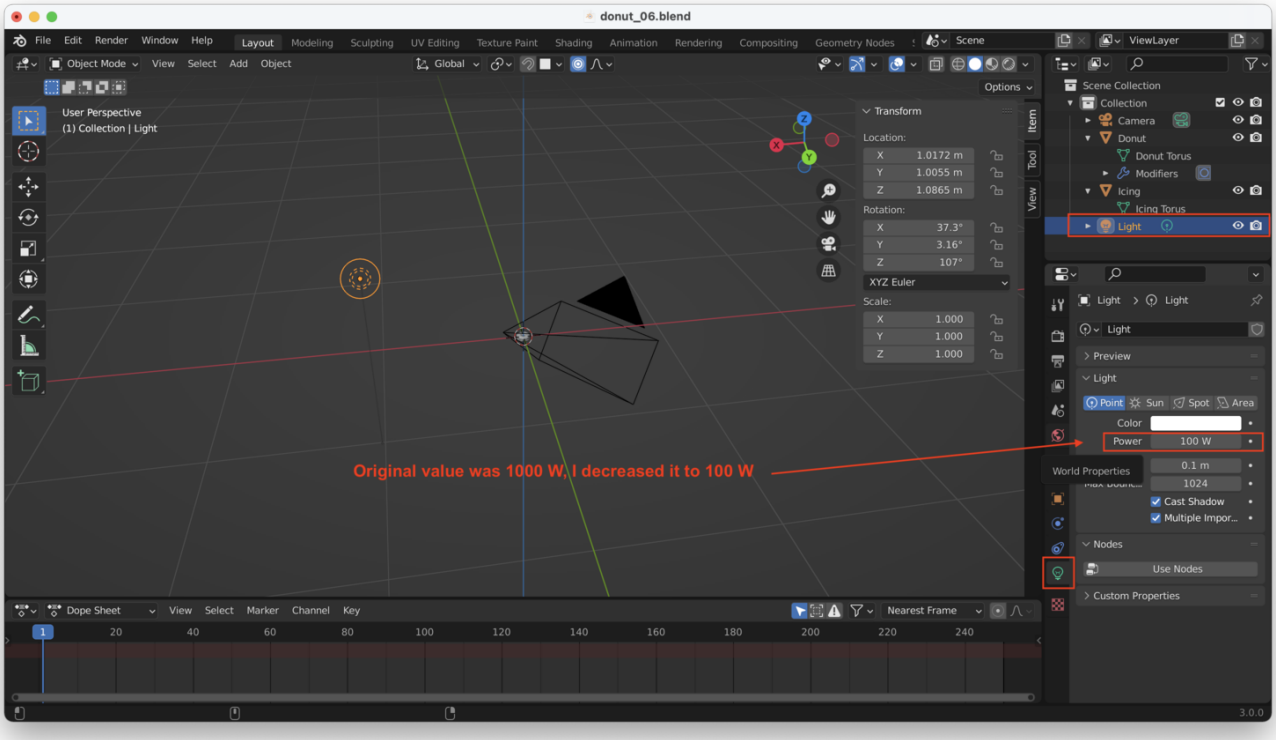


Tip: Don't forget, you can hit "X", "Y", or "Z" after hitting "G", if you want to move your light along a specific axis.

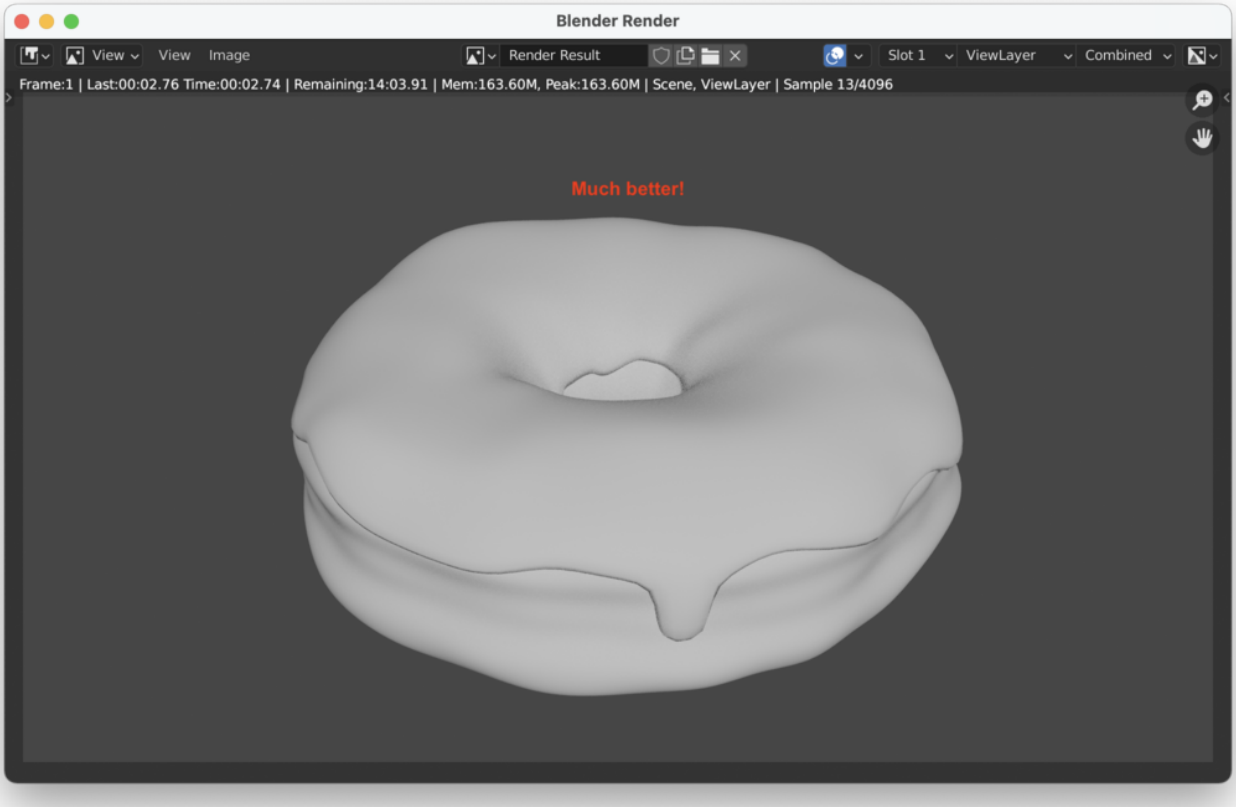
Of course, we haven't adjusted the intensity of the light, so if we render now, we'll get something like this:



We can adjust the "Power" of the light down, to get a something more reasonable:

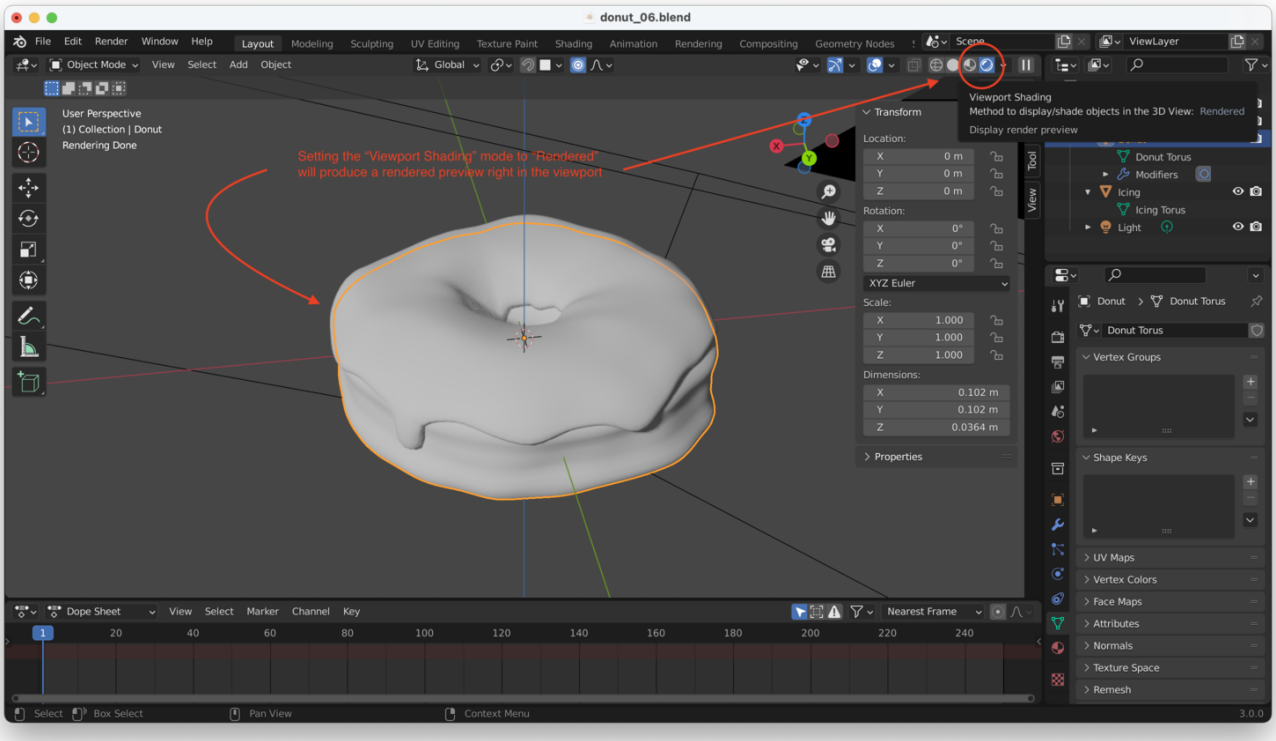


Our render now looks a lot more reasonable:



Rendering In The Viewport

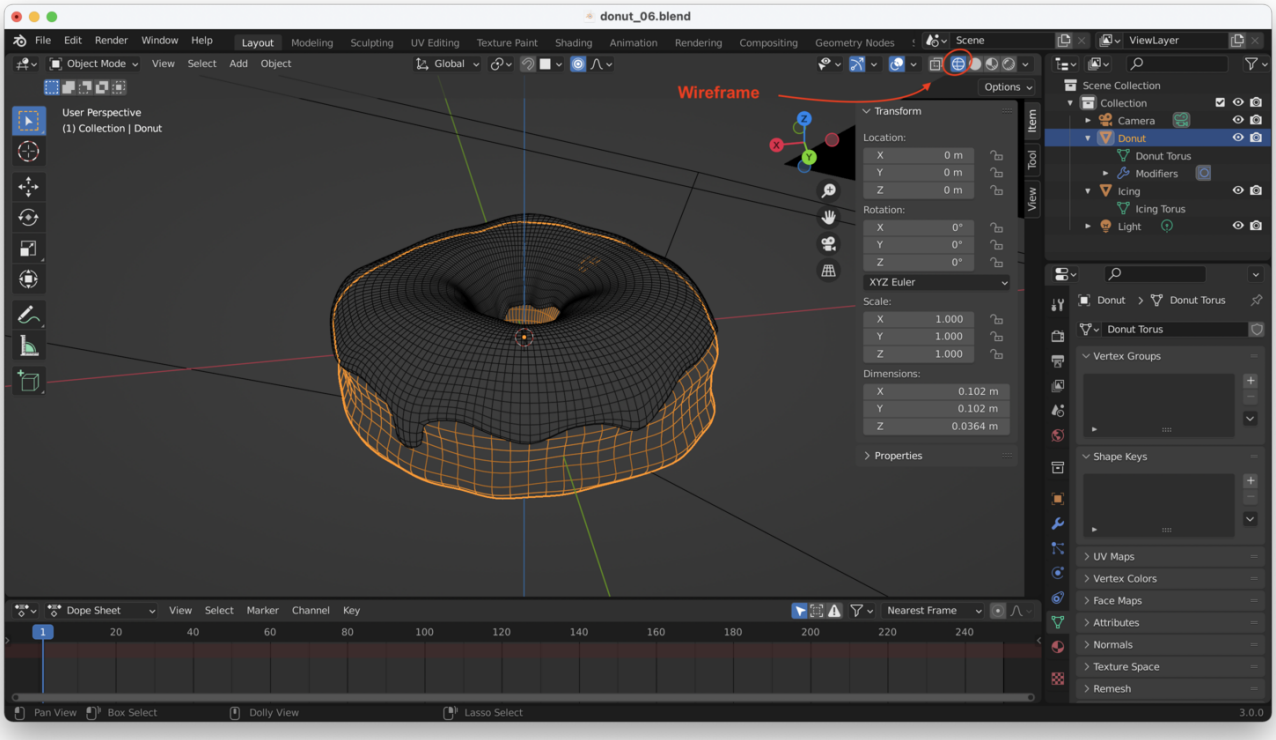
You can switch your "Viewport Shading" mode to "Rendered", which will automatically render your scene in the viewport, re-rendering whenever you change your location in the viewport:



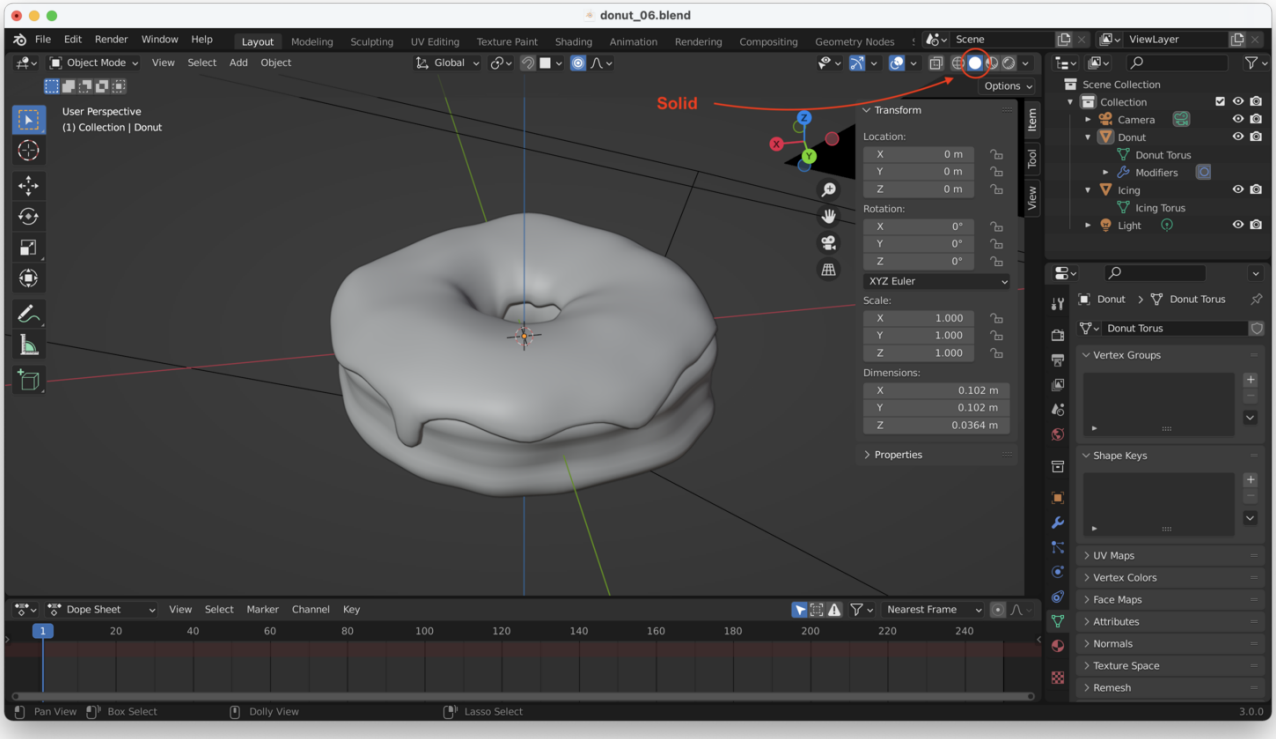
> **Tip**: Play around with the different "Viewport Shading" modes.

Other Viewport Modes

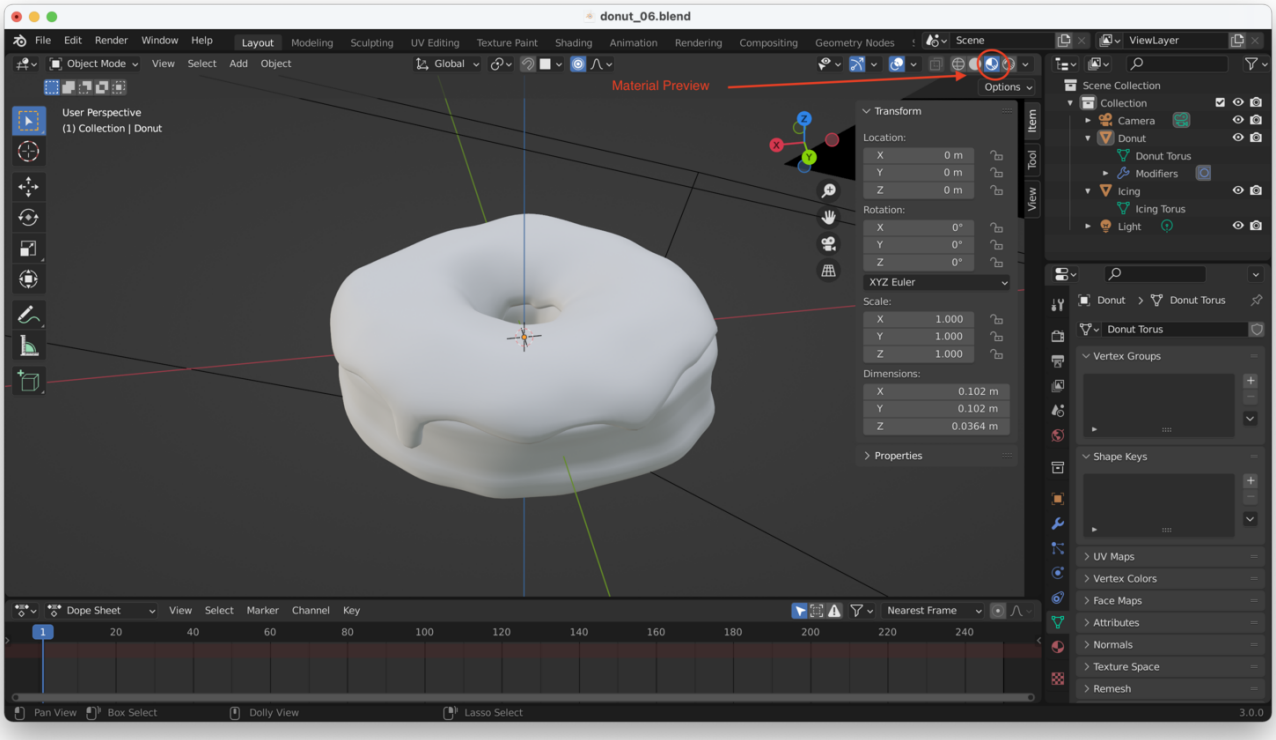
Wireframe (makes it easy to see your model's mesh):



Solid (the default - good for shaping your objects, and for sculpting):



Material Preview (good for checking how reflections will look, what materials will look like with more natural lighting):



How Is Stuff Rendered In The Viewport?

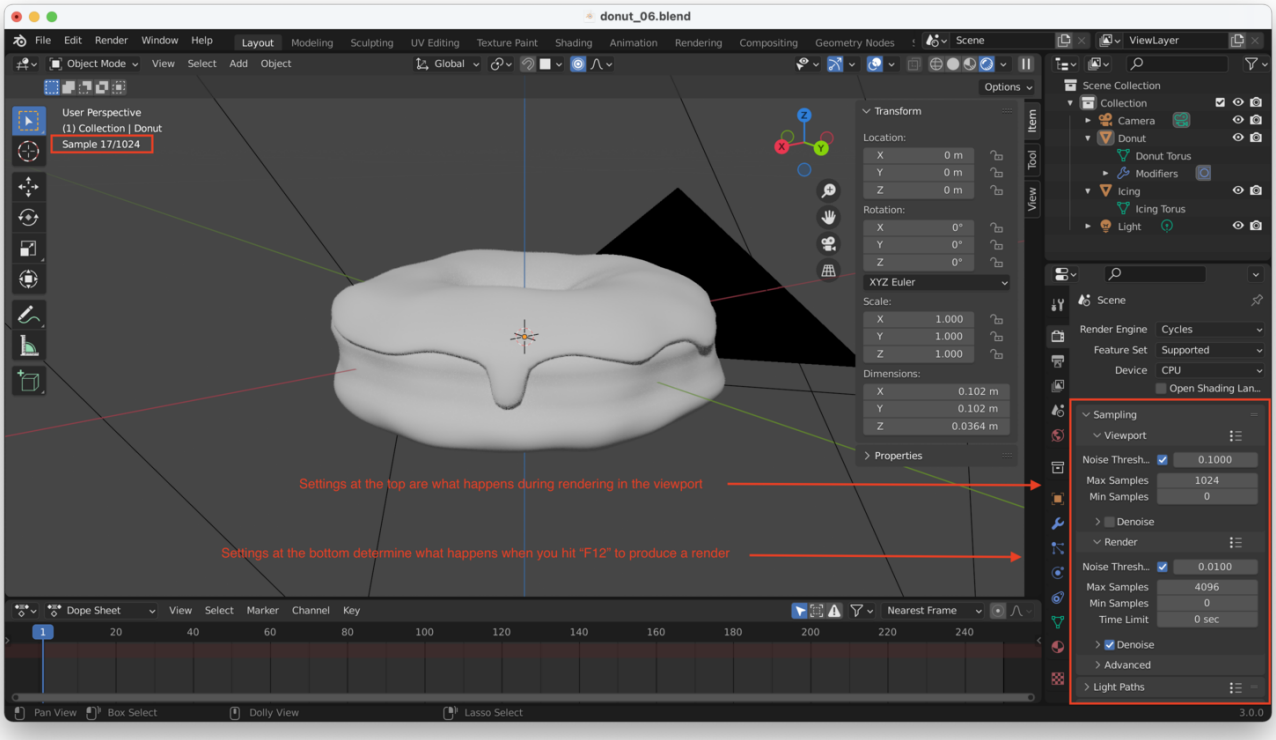
A special tool called a "Rendering Engine" is used. There are many open source and commercial rendering engines: they take your model and apply a set of rules to calculate what your final animation or image should look like, based on lighting and material properties set for the objects in your scene.

There are many commercial and open source rendering engines, but the ones Blender comes with are "Eevee" and "Cycles". By default, rendering in the viewport is done with "Eevee".

"Eevee" is a *realtime* rendering engine, something like what would be used in a video game, to produce instantaneous results.

"Cycles" is NOT real time, rather it is a "path tracer" or "ray tracer". This means cycles actually calculates how light "bounces" around your scene. This means Cycles is not as fast as Eevee, but produces more accurate, realistic renders.

You can turn on Cycles from the rendering menu:

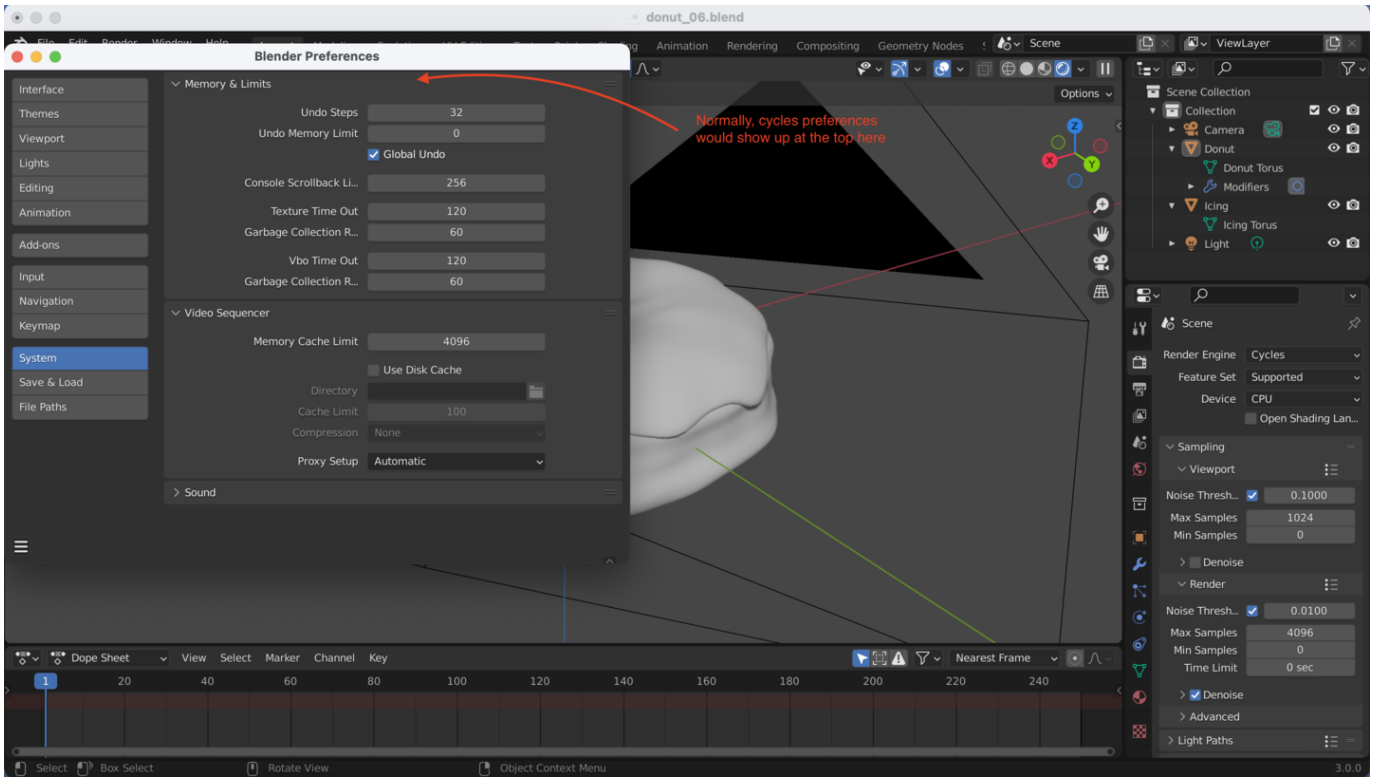


Speeding Up Renders

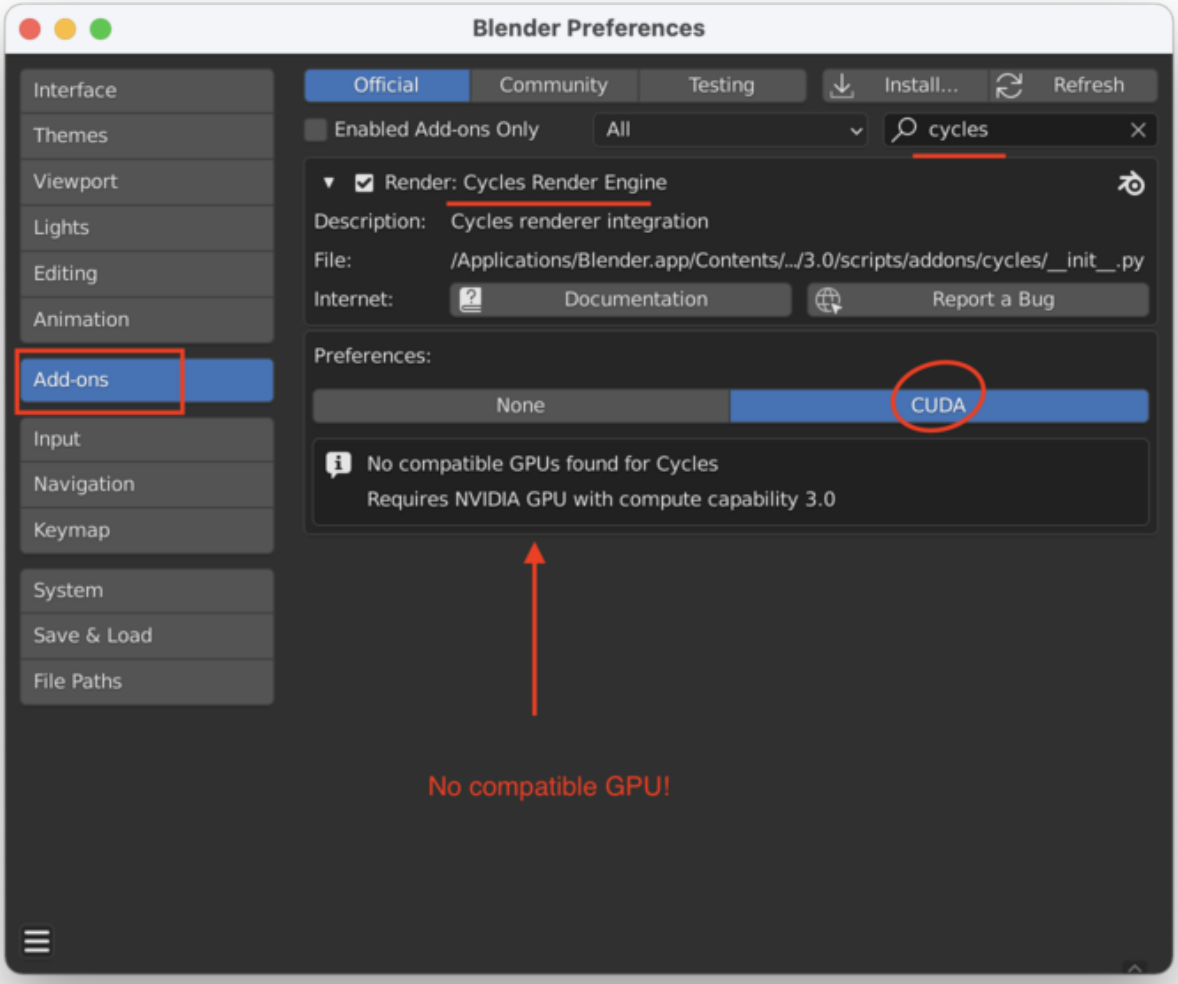
Rendering using a GPU will pretty much always be faster than rendering with a CPU. You can make sure Cycles uses your GPU by going to Edit → Preferences → System, and changing your "Cycles Render Devices" to your GPU(s).

Normally, you'll see options for CUDA, OptiX, and HIP. If you've got a NVIDIA GPU, you'll use CUDA or OptiX (use OptiX if possible). If you have an AMD GPU, you'll use HIP.

Note: If you are on a newer MacBook Pro (like me: I have an M1), you won't see any cycles preferences at all, like this:



You can confirm that the issue is a missing graphics card by clicking on "Graphics", then searching for "Cycles". As you can see here, when I click on "CUDA" the system warns me that I don't have a compatible GPU:



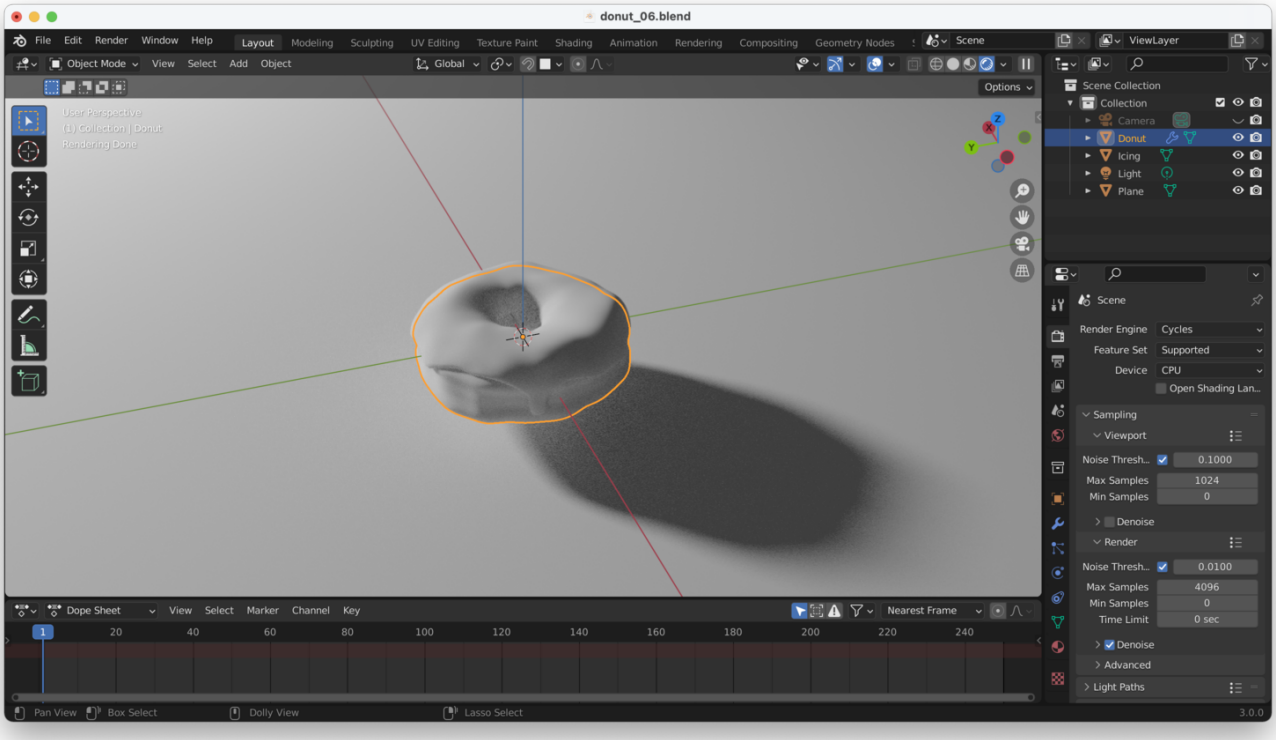
If you don't have a compatible GPU don't worry, Cycles will work fine, it'll just render using the CPU which will be significantly slower.

What To Do If You Don't Have A GPU

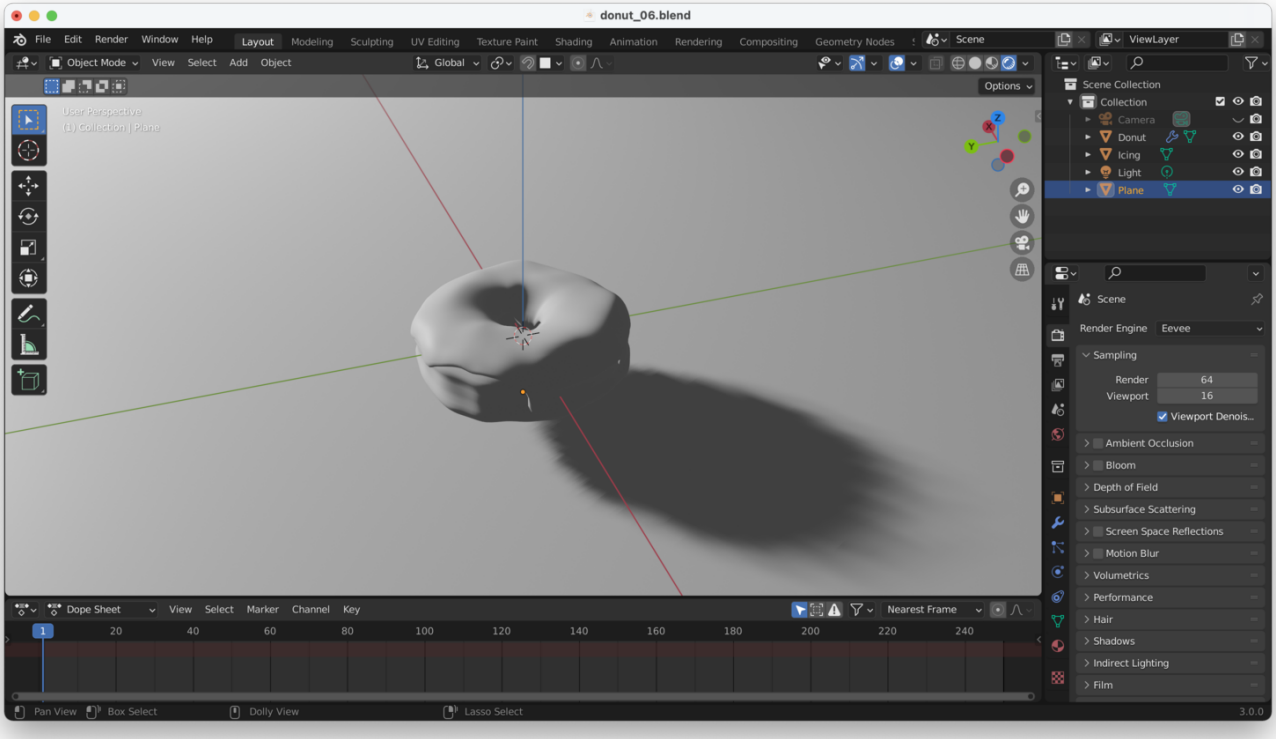
You can use Eevee instead of Cycles (it'll be much faster), but you'll need to make adjustments to get Eevee to do a good job, especially with **shadows**.

Note: I have added a plane underneath the donut to make shadows more obvious.

Here's **Cycles** trying to render a shadow behind the donut:



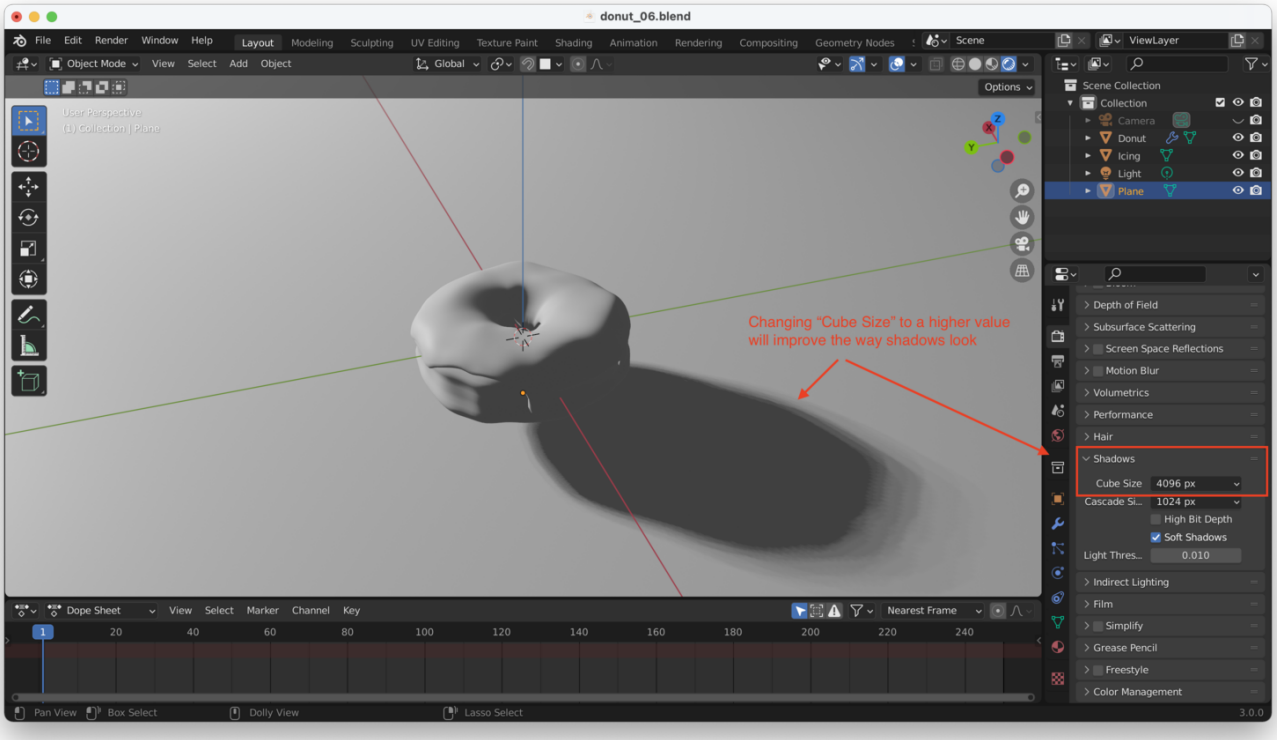
Here's that same render with **Eevee**:



Big difference, yes? If we want to get a more realistic render with Eevee, we need to make some adjustments to Eevee's render settings.

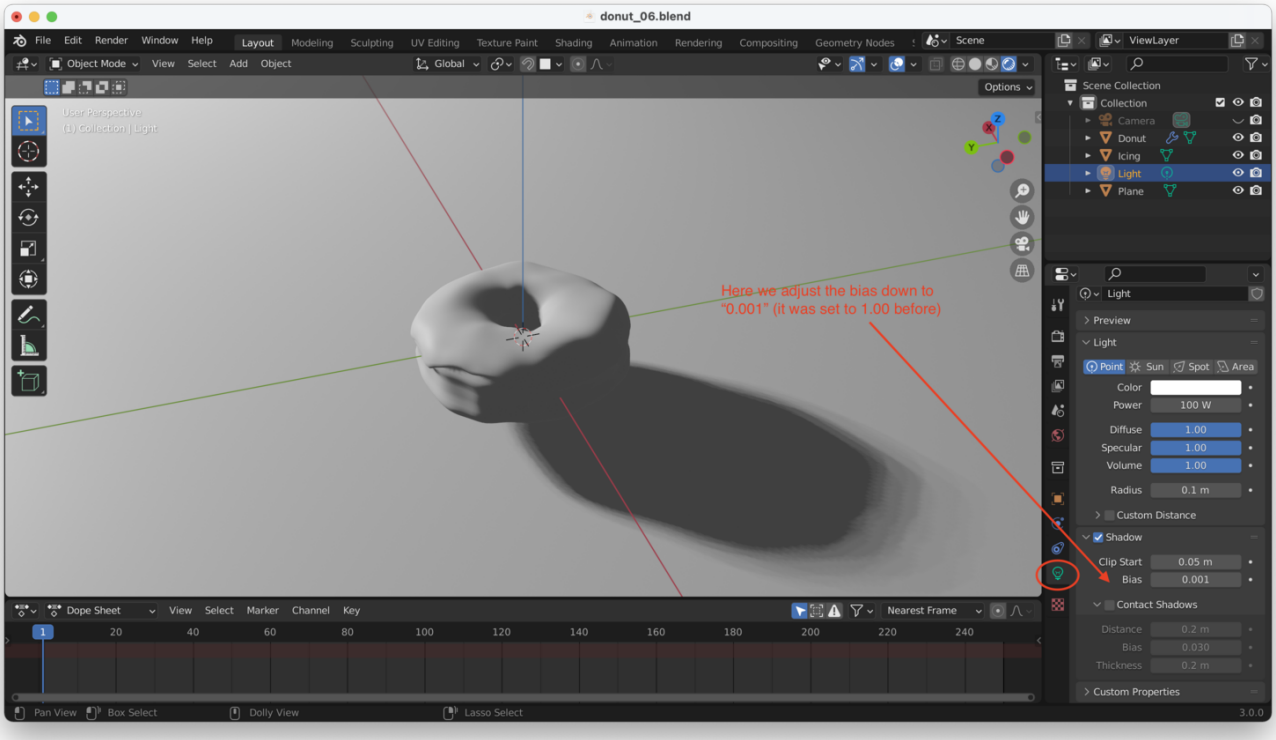
Adjusting "Cube Size"

Changing the "Cube Size" in the "Shadows" settings will make shadows appear much less "jagged":



Adjusting Lamp Settings

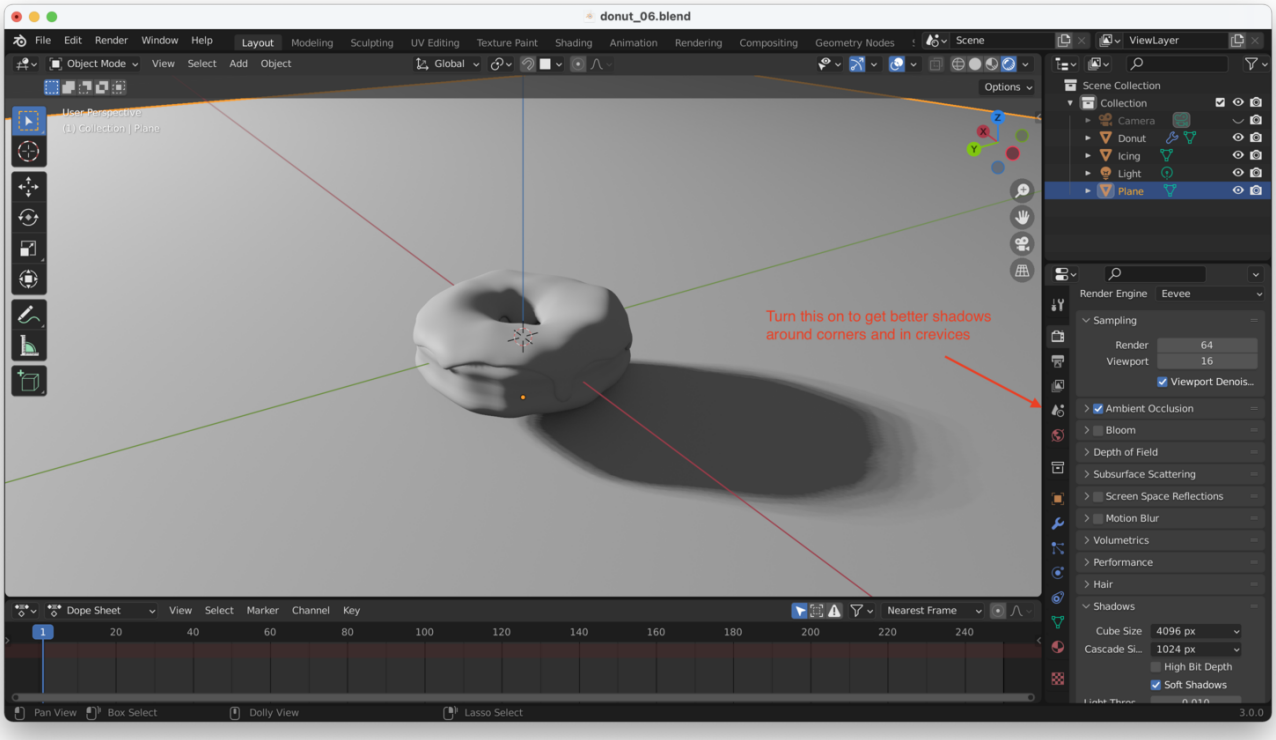
We can make the shadow appear still *more* realistic by changing our Lamp's *bias settings*. This will change the way Eevee treats shadows around smaller parts of our scene (like the "hole" in our donut). Here it is turned down to zero:



Playing with "contact shadows" can also help, but we'll ignore that for now!

Adjusting "Ambient Occlusion"

We can also get better shadowing in crevices and "occluded" (covered or obstructed) spaces by turning on Eevee's "Ambient Occlusion" feature:

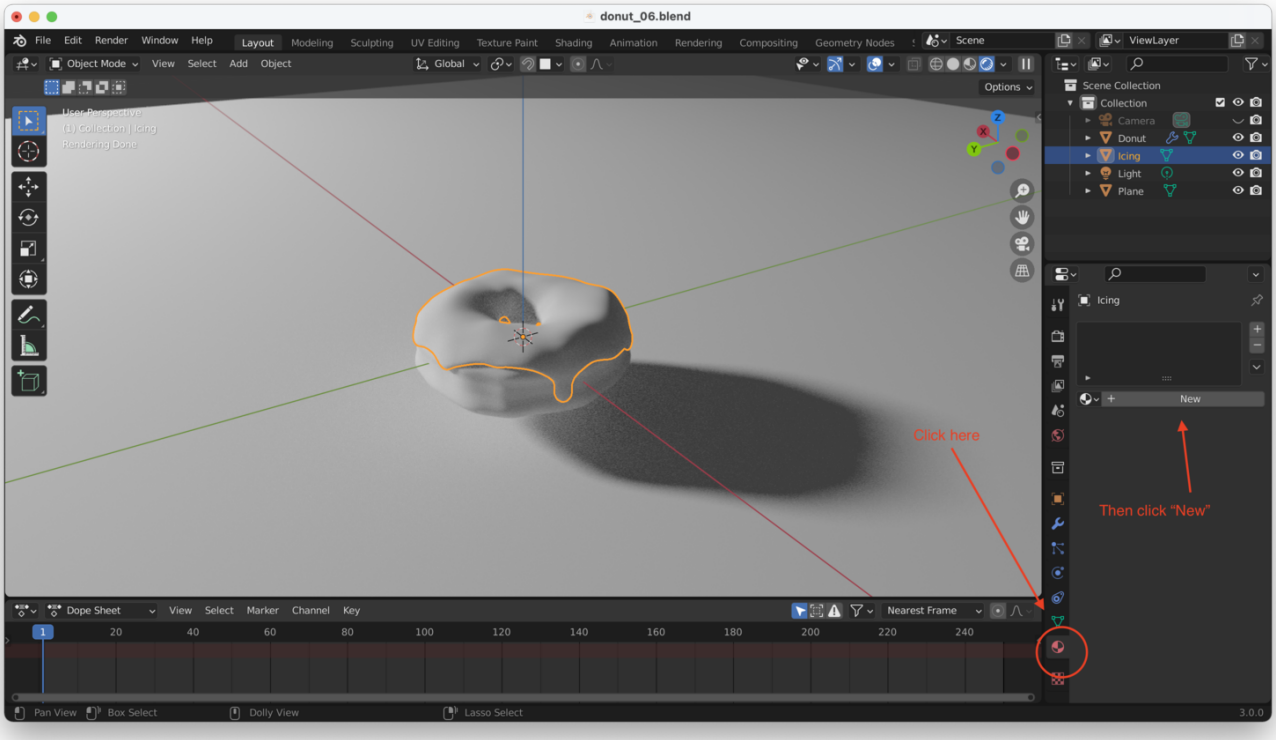


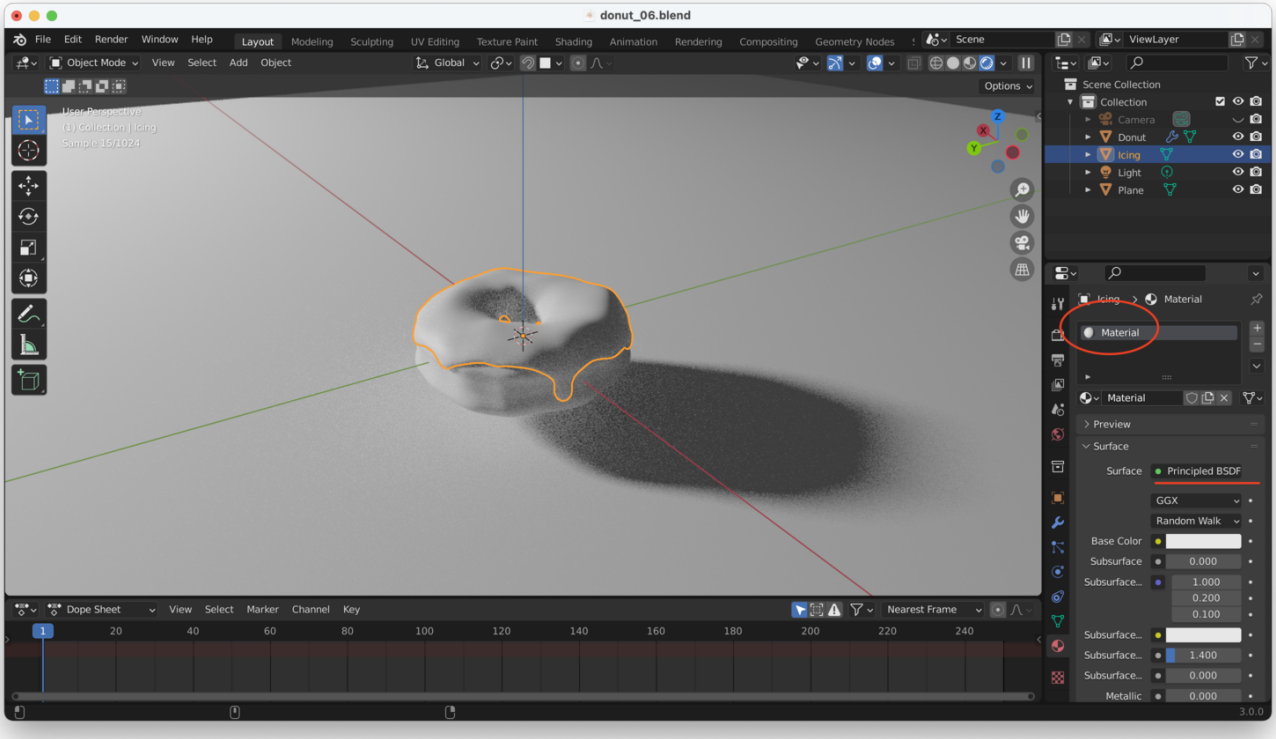
However, **the bottom line is that Cycles is always going to generate more realistic results!**.

Materials

We can associate one or more "materials" with each of our objects. This is really what gives objects a realistic look when they are rendered (fur, glass, hair, etc...).

Let's add a new material to the icing:

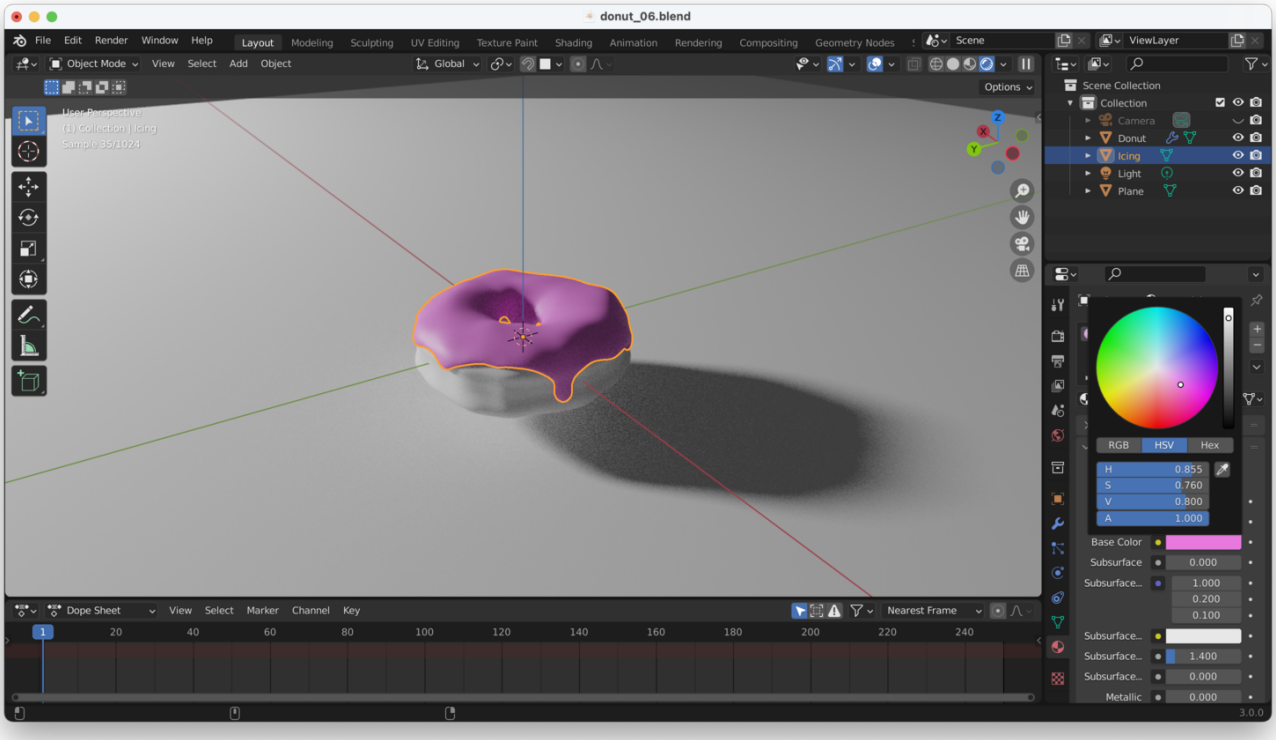




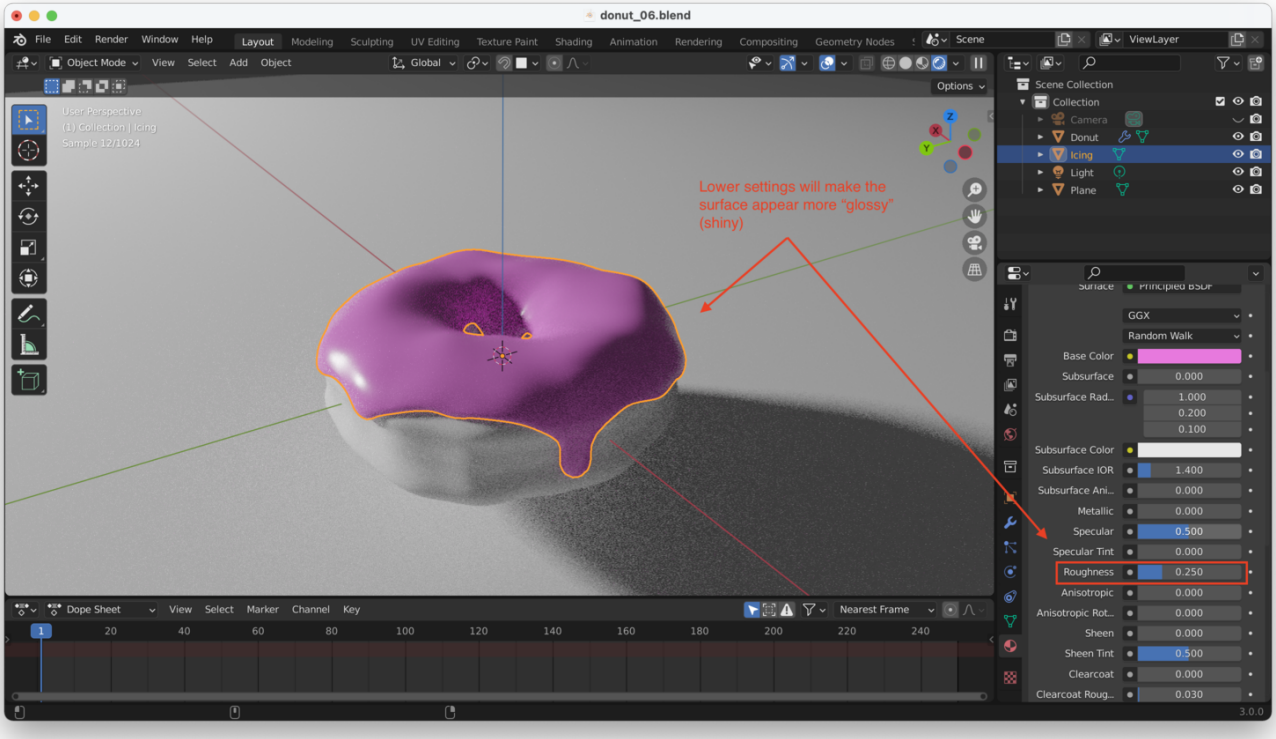
The new material is a "Principled BSDF". This lets us set things like roughness, specularity, etc...

This actually comes from Disney! You can read more in [the blender documentation]

The first and most obvious thing to change is the "base color". Let's make it pink:

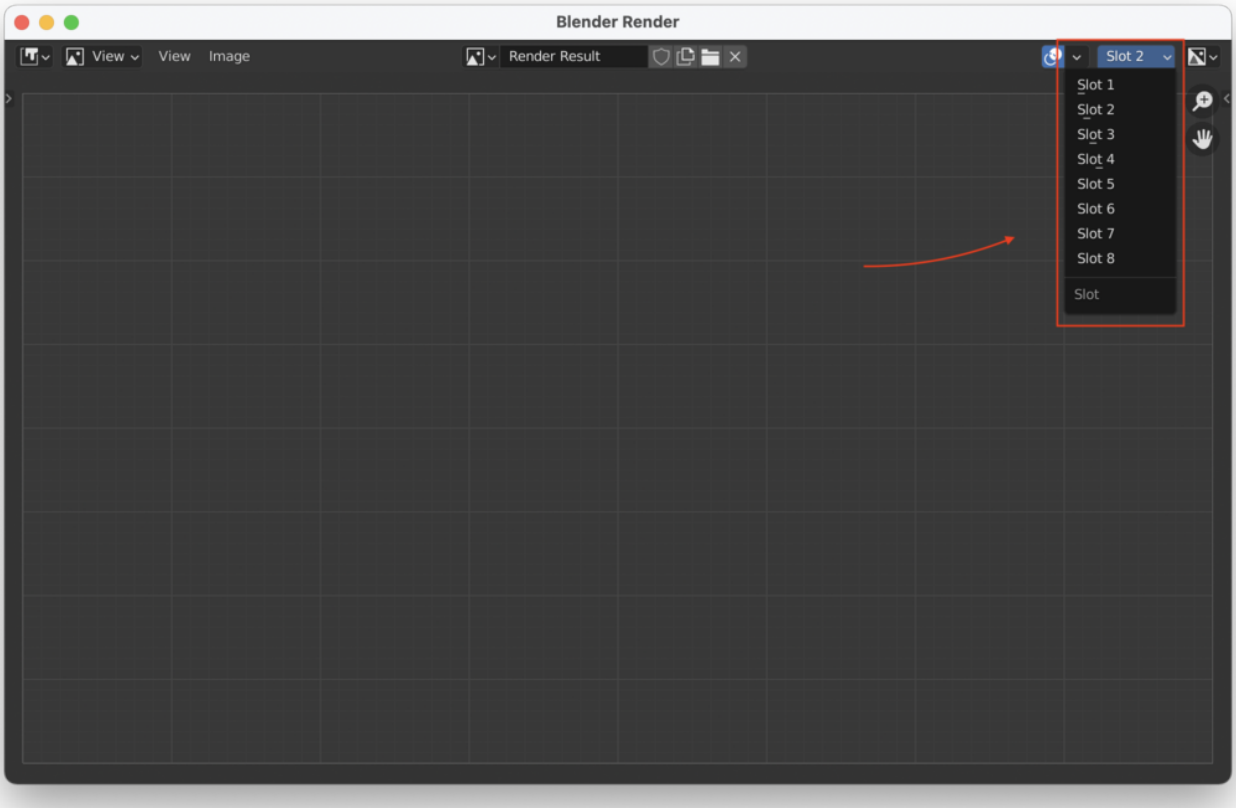


We can also make our icing appear more "moist" by adjusting the "roughness":

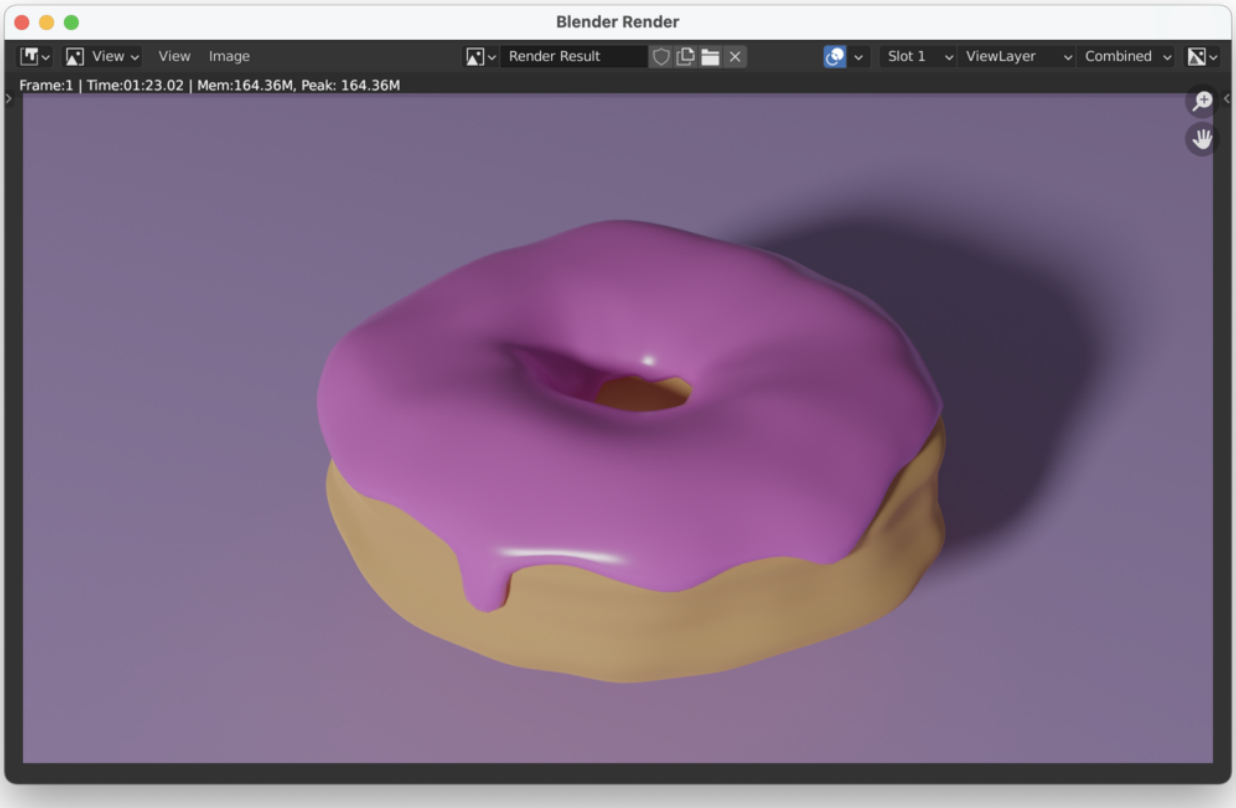


Doing Multiple Renders

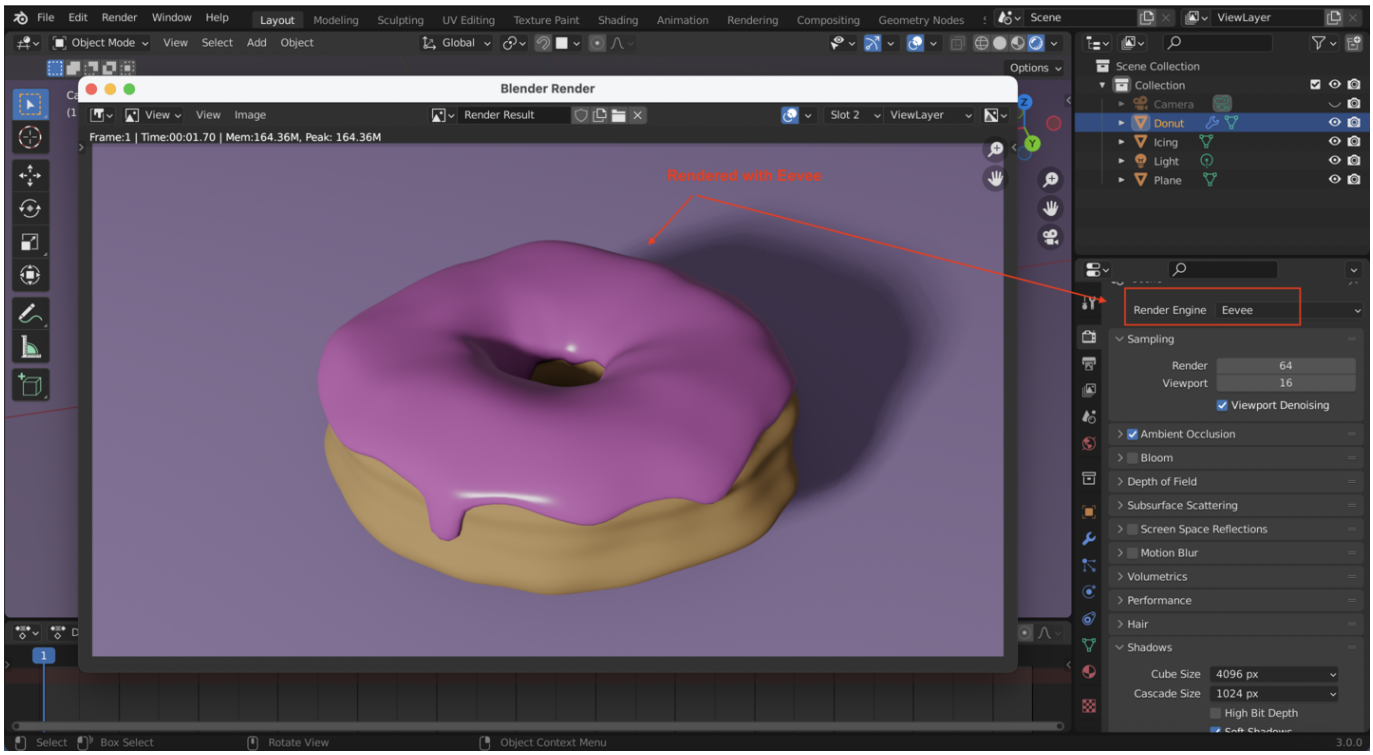
After hitting "F12" or "Fn + F12" you'll notice that you have more than one "Slot" available from the rendering window:



You can use these "slots" to perform multiple renders and compare your results. For instance, here I use Slot 1 to perform a render with "Cycles":

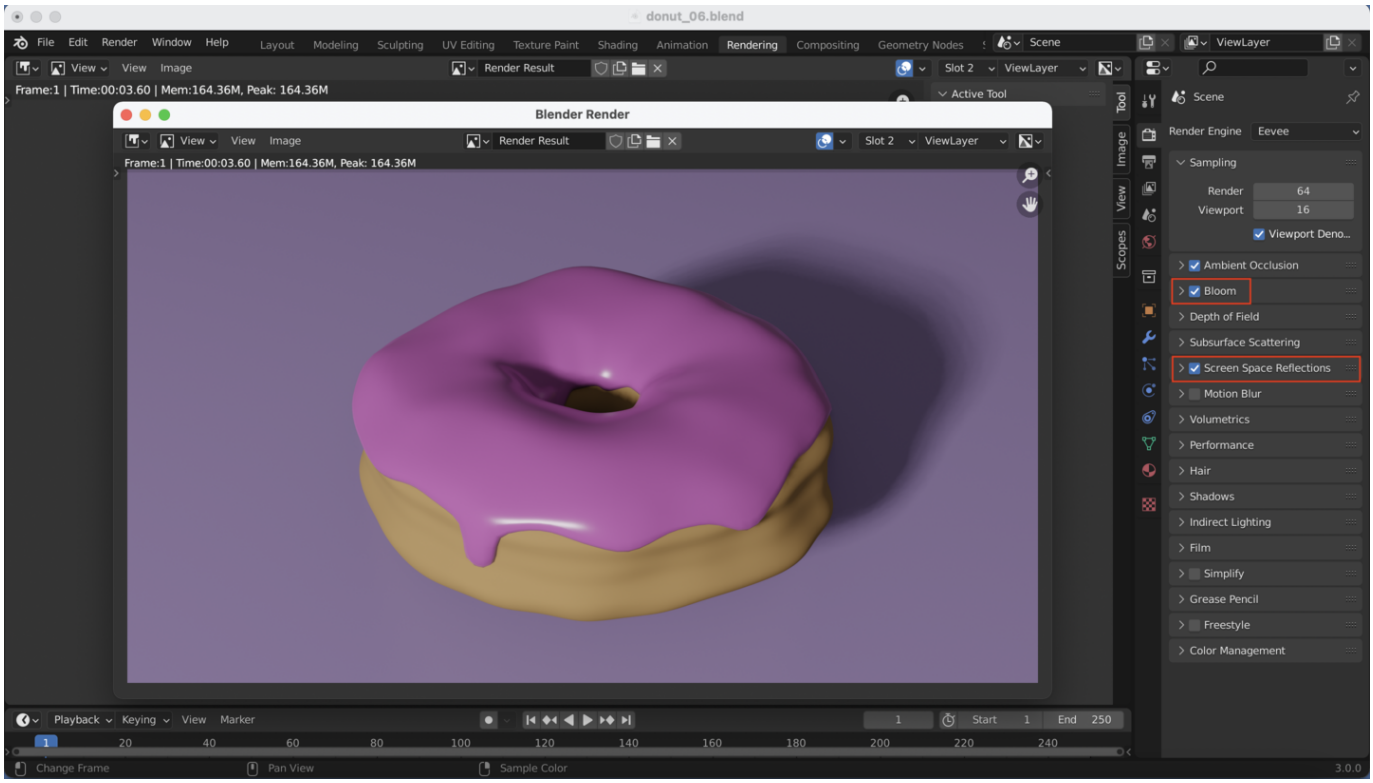


And here's a render with Eevee, in Slot 2:



The reflections here don't look quite as good. We can improve things by turning on "Screen Space Reflections" and "Bloom", which will (sort of) change the amount of "glare"

in the render:



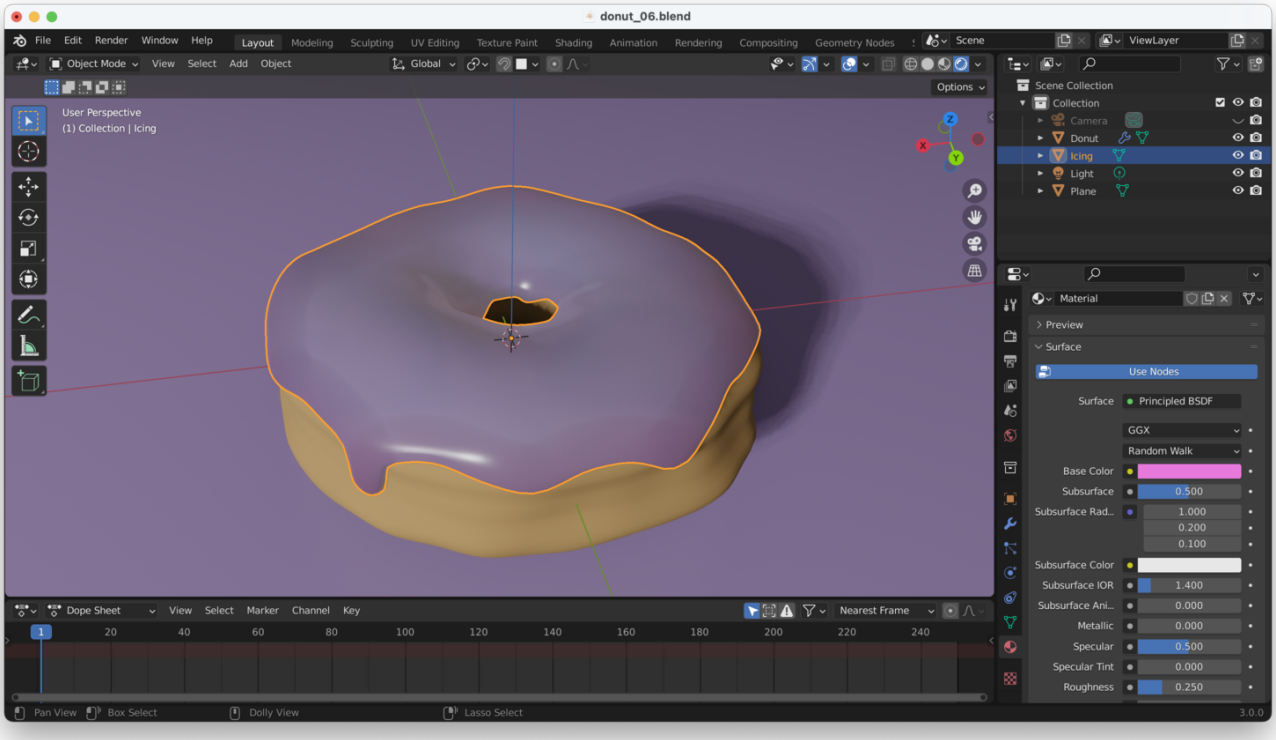
Feel free to open up the "Bloom" and "Screen Space Reflections" menus and play around a bit to see what kind of results you can get!

Improving The Icing Further

Many real-world objects exhibit "scattering" (i.e. light bounces around *inside* the object, or passes through it).

Without scattering, our icing will look "hard", almost like a shiny metal surface. We need scattering to give the light bouncing off the donut a softer, more diffuse feel (and a sense of depth).

From the icing's material properties, we need to adjust "Subsurface Scattering" and "Subsurface Color" to get this effect:

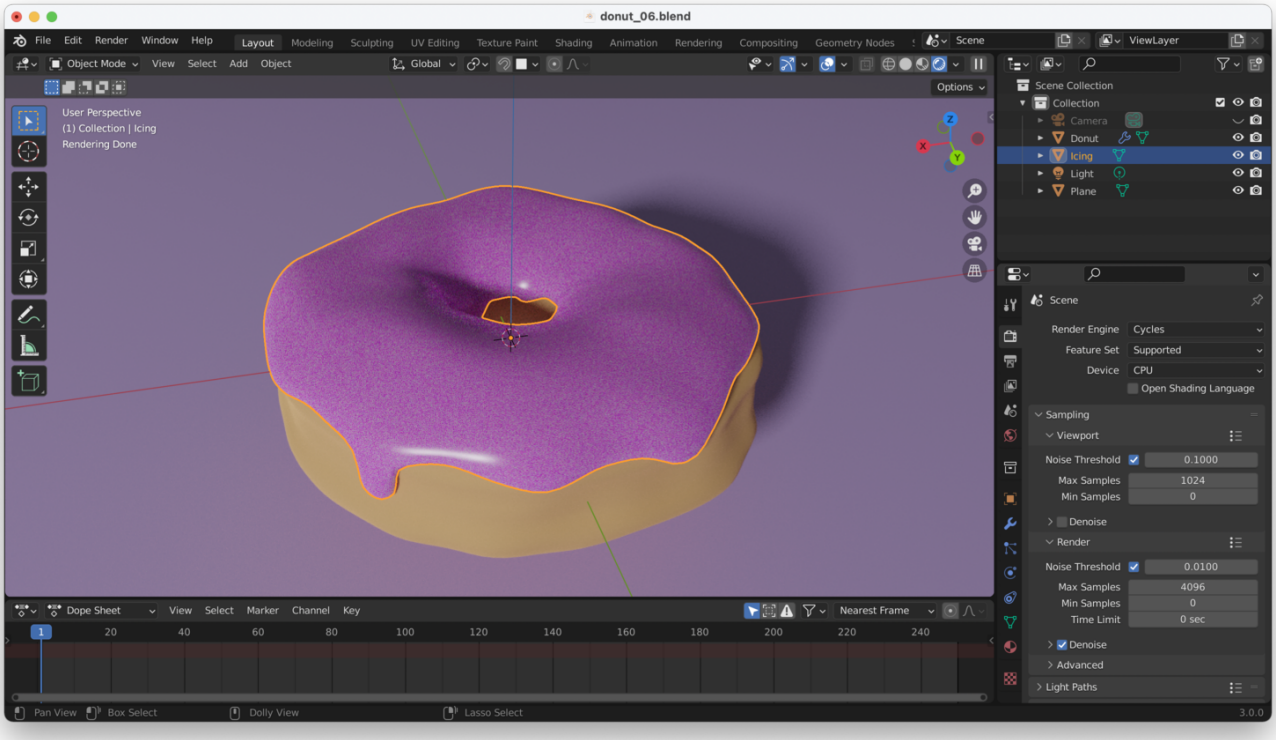


The weird "plastic" look we are now getting is caused by the fact that the "Subsurface Radius" values are too high. We should probably also adjust the Subsurface Color (now a light gray) to something that matches our icing better.

Basically, we need to do a few different things:

1. Adjust the subsurface radius values to something smaller, like "0.002"
2. Adjust the subsurface color to something closer to the color of our icing (so a pink or purple color, in this case)
3. Play with the "Subsurface" effect strength

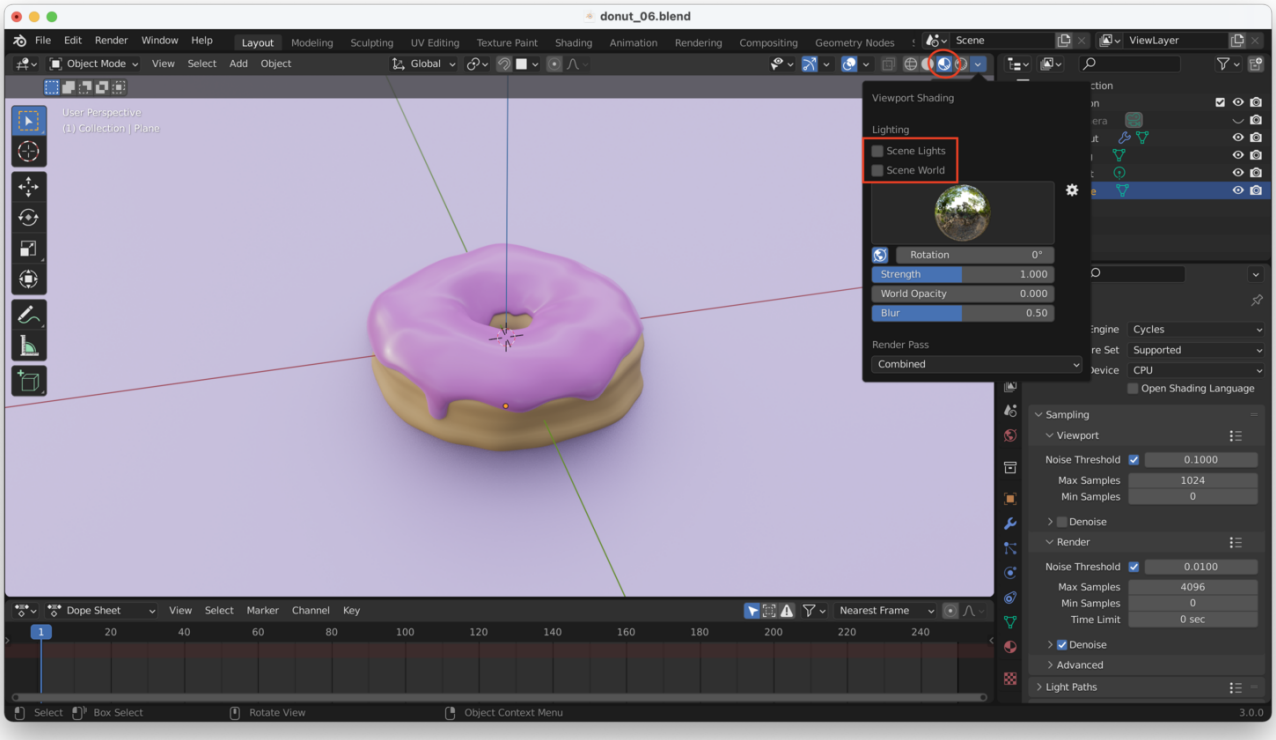
After twiddling those values and adjusting my colors a couple of times, I wound up with this:



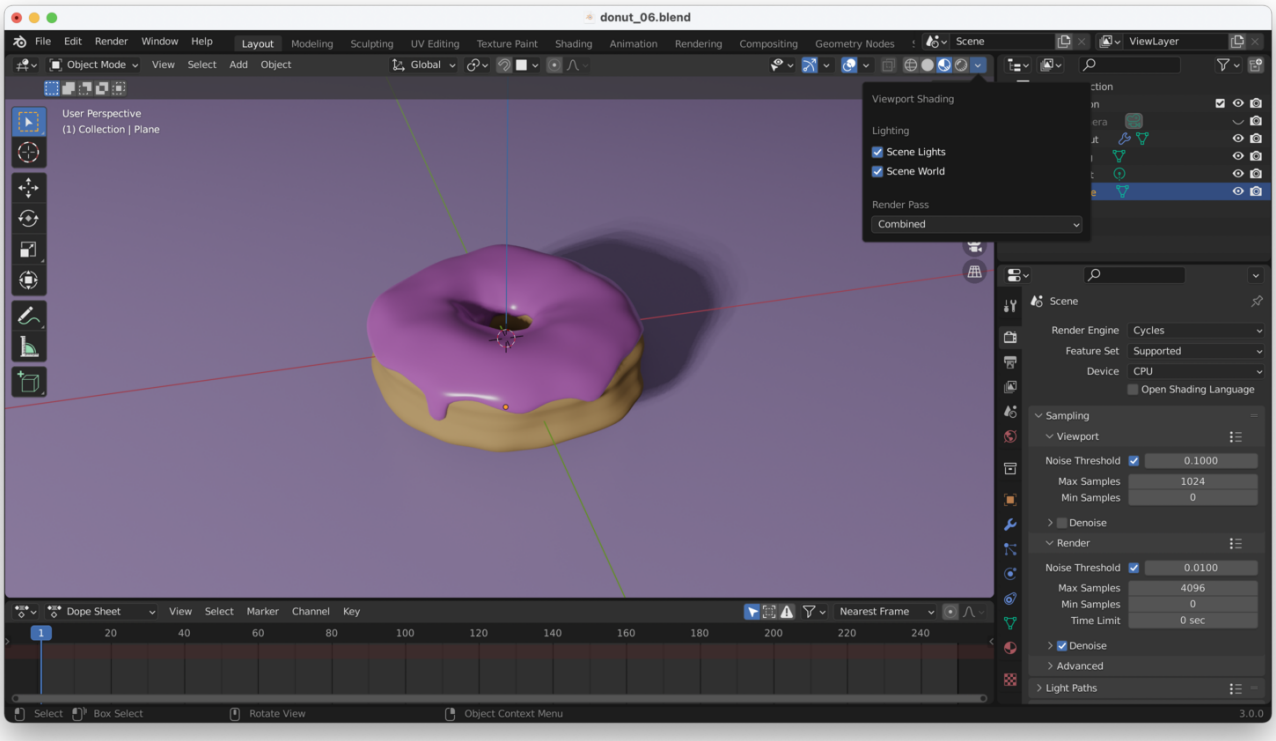
> **Tip:** Some of these finer adjustments (like subsurface scattering) are going to be tough to see when using Eevee. Cycles will pretty much always give more satisfactory results.

Using Material Preview

We can use Material Preview to get a quick idea of what our materials are going to look like, without doing a render. It will use default lighting and background settings, as long as "Scene Lights" and "Scene World" are **unchecked**:



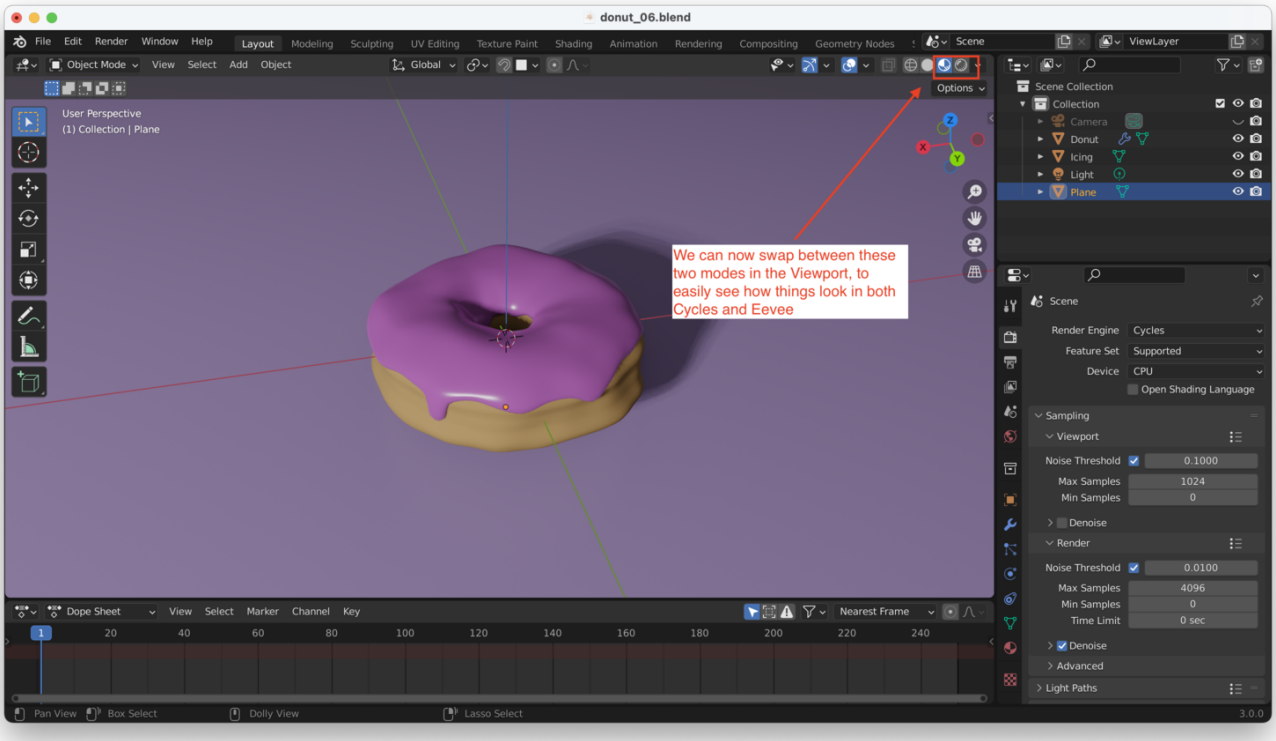
If we check those boxes, we'll see a preview that uses *our* world and light settings (good in our case, since we'll be rendering using our own background and lighting):



> **Note**: Actually, Material Preview uses Eevee to render, so with "Scene Lights" and "Scene World" checked, what we're seeing in the Materail

> Preview is the *same* as what we would get by rendering the scene with Eevee.

So what's the point? We can now swap between "Material Preview" (to see a fast render in Eevee) and "Rendered", to see results from Cycles:



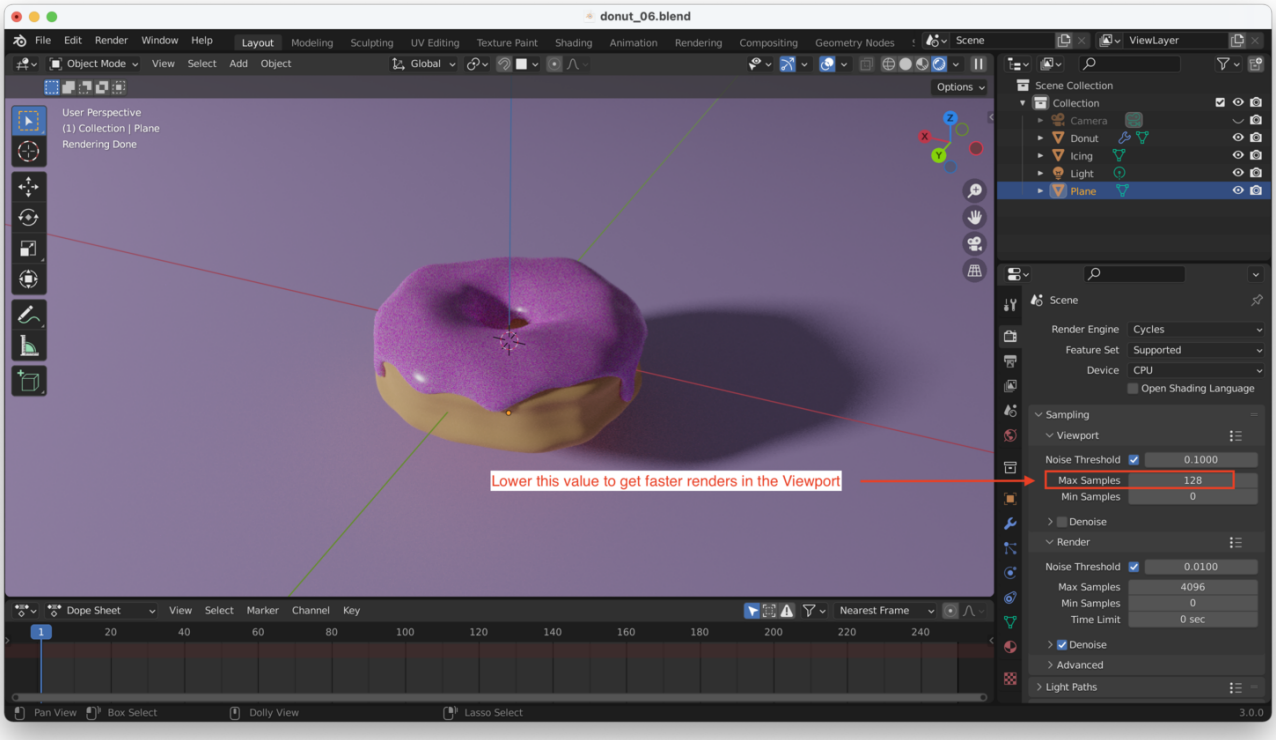
Why Does Cycles Make "Grainy" Looking Renders?

Cycles is doing "ray tracing", meaning it actually traces the path each ray of light would take through the scene before reaching the camera (in reality it is actually "working backwards" from the camera to the light, but the effect is the same).

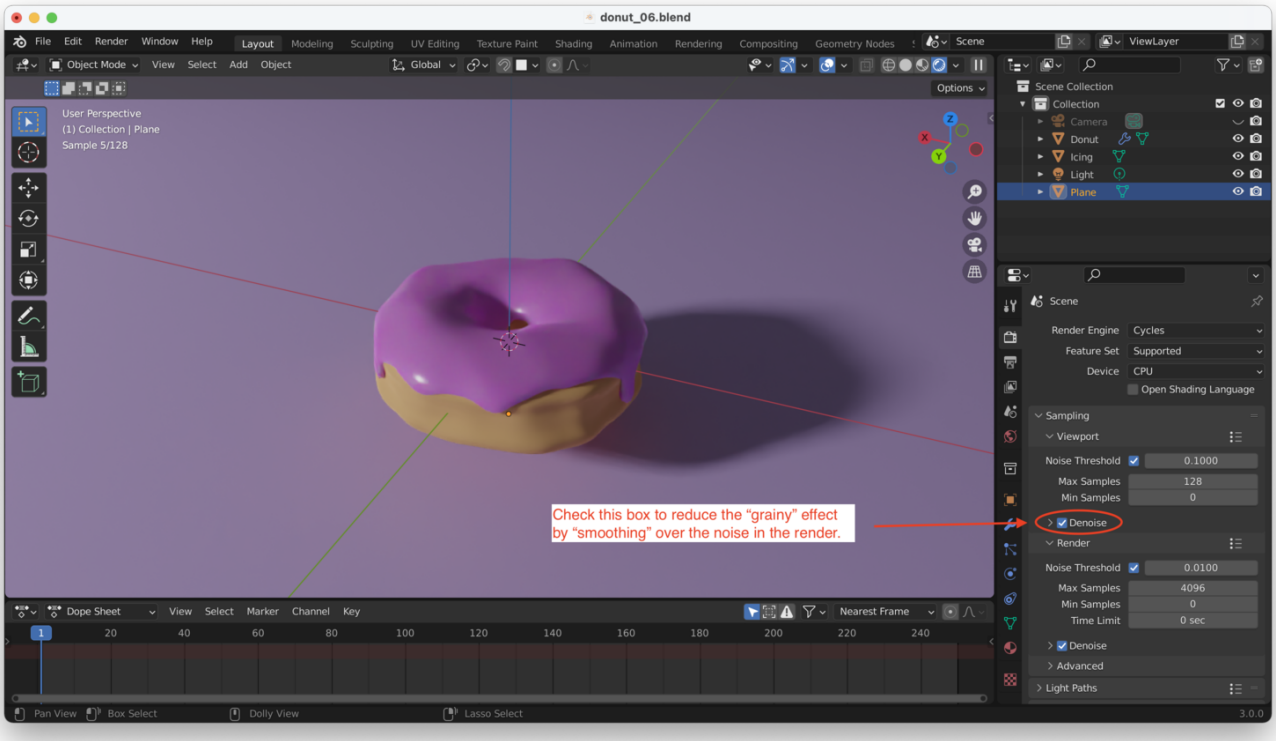
This is an incredibly computationally intensive process, so Cycles has to choose a random "subset" of rays to trace. This is what causes the "grainy" effect. Tracing more rays (letting Cycles run for longer) and allowing more "bounces" off of objects in the scene will *both* improve the quality of the render, at the expense of taking more time.

How Can We Improve Things?

We can **improve render speed** by lowering the "Max Samples" used when Cycles renders things in the Viewport. This will speed up Viewport rendering:



We can **improve render quality** by turning on "Denoising", which will "smooth out" some of the noise:



> **Note**: If you're using your CPU for rendering, turning on denoising will slow things down significantly. Unfortunately hardware accelerated denoising

> only works when you've got a GPU!

Note: In Blender 3.0 there is a "time limit" feature for rendering, so you can force rendering to complete within a given time, regardless of quality: