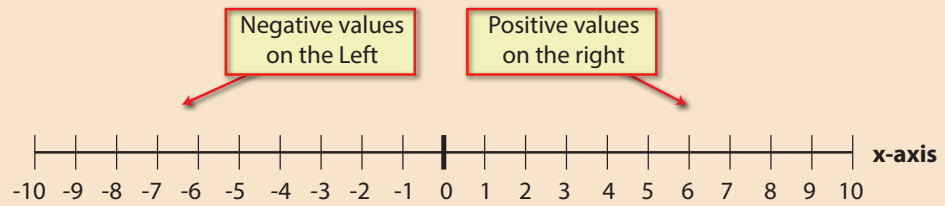


# Axes and Coordinates

When creating a 3D scene, we need to specify the position of any new object.

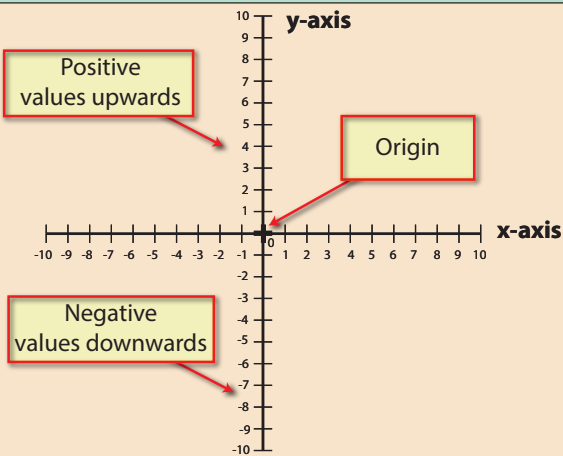
To do this we employ axes which act as a type of ruler allowing us to measure distances in one, two or three dimensions.

If we were measuring distances in a one dimensional space (say, along a telephone wire or a spider thread), we could use a single, imaginary axis (known as the **x-axis**). The centre of the axis is labelled zero with positive increments to the right and negative increments to the left.

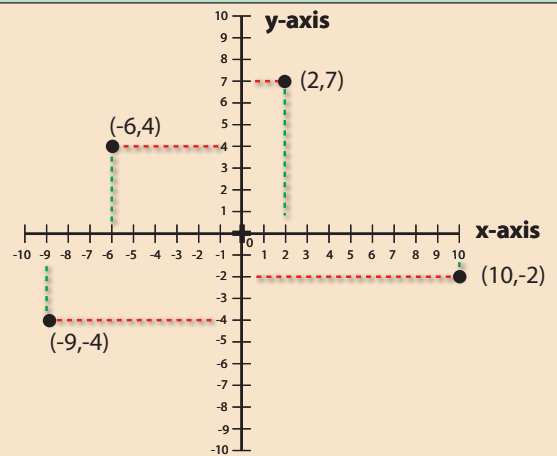


In theory, the axis expands indefinitely in both directions

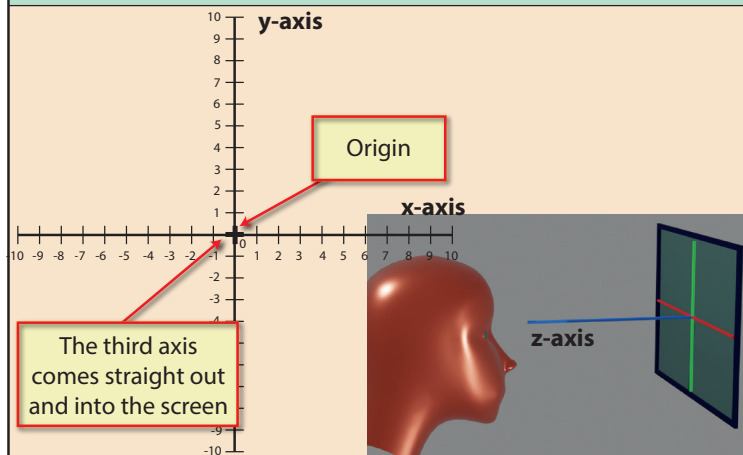
In two dimensional space – for example, a sheet of paper – we use two axes. These are perpendicular to each other and cross at their zero points. This meeting point is known as the **origin**. The new axis is known as the **y-axis**.



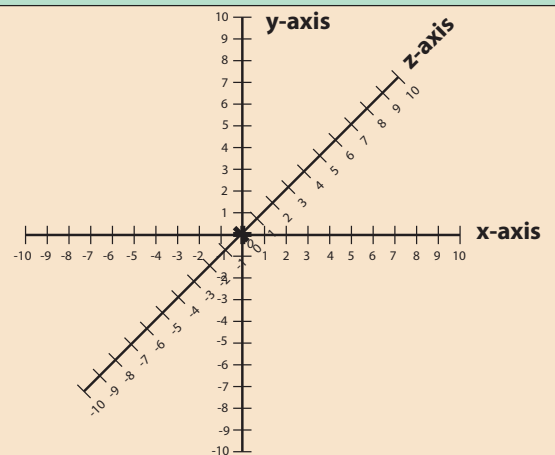
With the two axes, we can specify the position of any point in space by giving its distance along the **x-axis** and then along the **y-axis**. By convention these are enclosed in parentheses and comma-separated.



For three-dimensional space (usually written as 3D space), we need three axes. The third axis (**z-axis**) is perpendicular to the other two. Of course, on paper or a 2D screen we have to settle for a mental picture of the final axis coming straight out of, and into, the paper/screen.



Often, in order give a visible presence to the **z-axis**, it is drawn at 45° to the other two axes. This can lead to some confusion, initially.

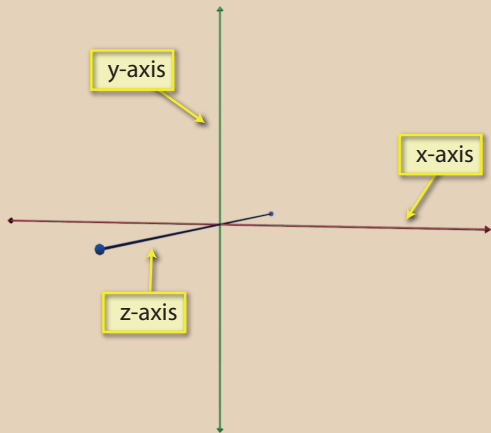




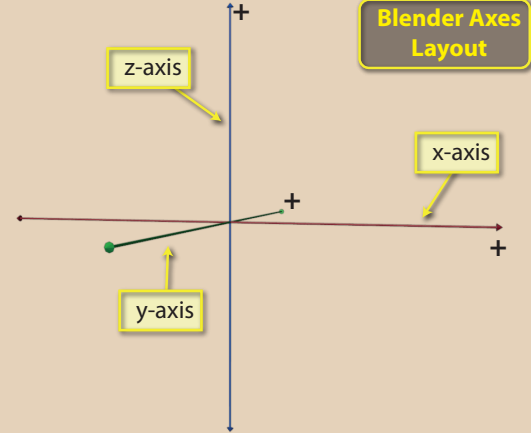
### NOTE

In the last frame we saw that the z-axis had its positive direction flowing “inwards” with the negative direction flowing “outwards”. In fact, there is no universally agreed convention on this and we may find that different software uses the opposite direction of flow for the z-axis.

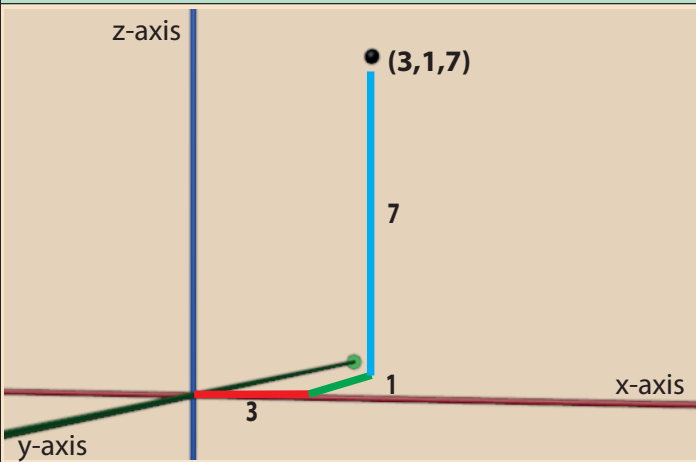
If we move our viewpoint in 3D space slightly, we can see all three axes.



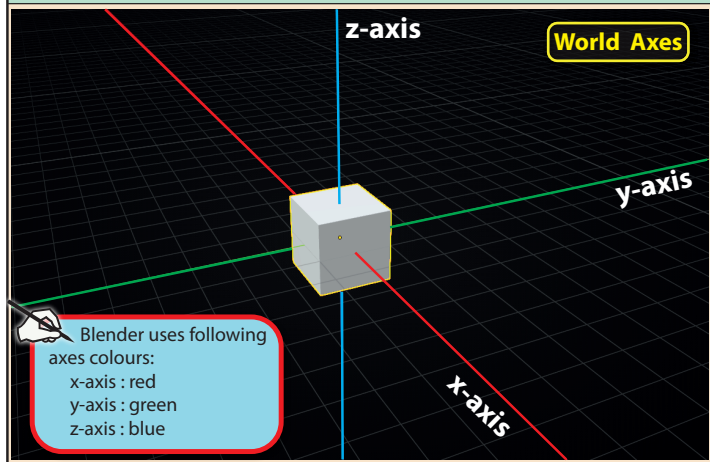
Although the previous frame gives the traditional layout of the axes, Blender rotates things by 90°, making the z-axis vertical. The positive end of the y-axis is now farthest away from our viewpoint.



To specify the coordinates of a point in three-dimensional space we must give its distance along each of the three axes in the order x, y then z.

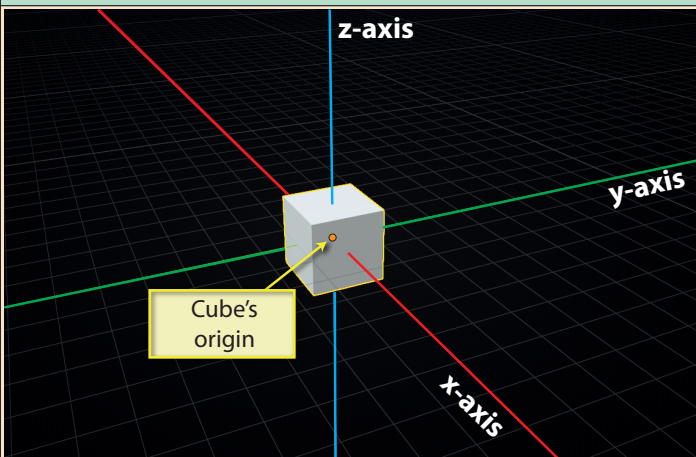


Blender uses several sets of 3D axes. The most important of these is the **World Axes**. It is this set of axes that is used to specify the position of every object we add to a Blender scene. Only the x and y axes are visible initially (the z-axis has been added below).

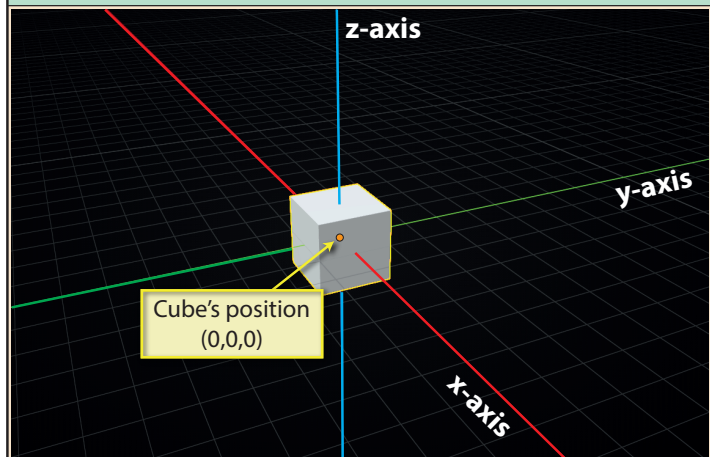


Blender uses following axes colours:  
 x-axis : red  
 y-axis : green  
 z-axis : blue

In a Blender scene, an object's centre (known as the **object's origin**) is used when specifying the position of the complete object. This position is represented in Blender by a **small orange circle**.



Since the *Cube's origin* is at the *World Axes's origin*, the Cube's coordinates are given as  $(0,0,0)$ .



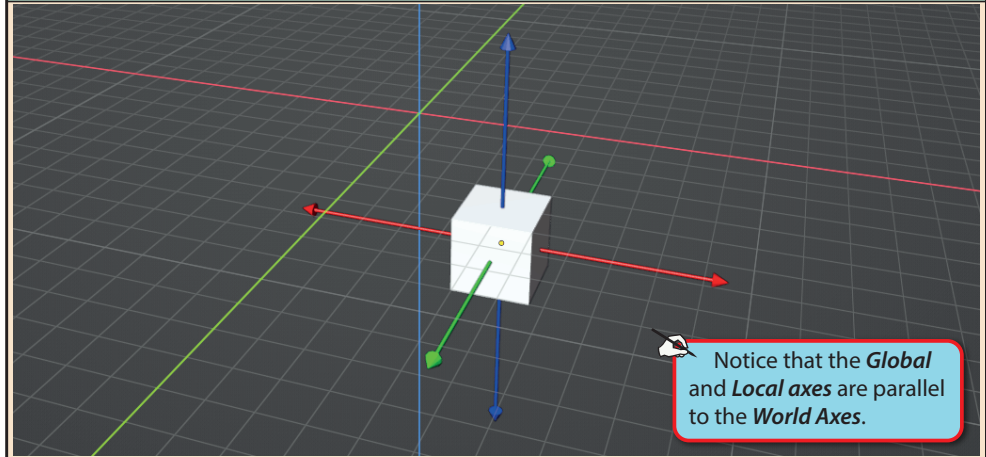
Every object in Blender maintains several different sets of axes, although, unlike the *World Axes*, the other axes cannot be made visible.

Two important sets of axes are the **Global Axes** and **Local Axes**. Every object in a Blender project has its own set of *Global* and *Local* axes.

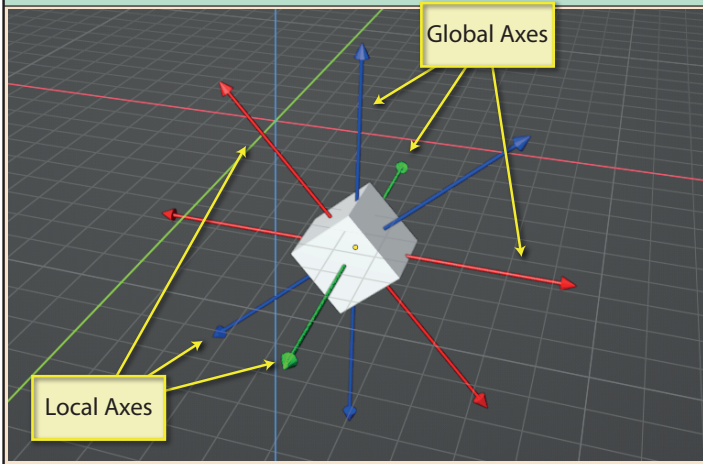
A less important set of axes is the **View Axes** whose position is determined by the location from which we are viewing the scene.

These three axes types are explained briefly here, but we will encounter other axes later in the text.

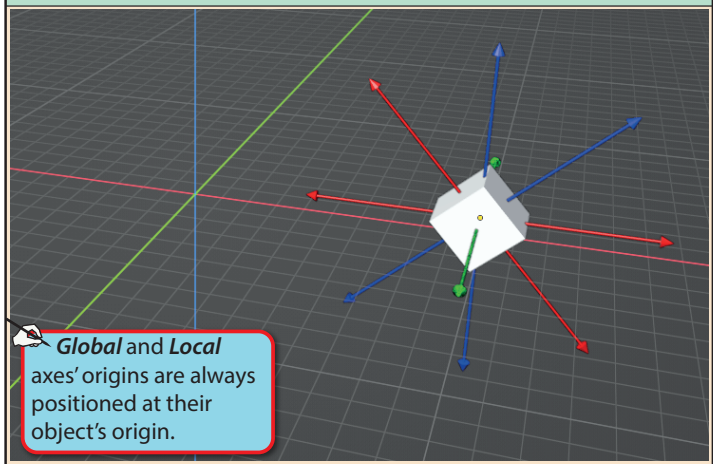
In the scene below we have a *Cube* object which has been moved away from its original position at the *World Axes origin*. Drawn onto the *Cube* are its normally invisible **Global** and **Local** axes. These two sets of axes initially occupy the same position and so only one set of axes are shown below centred on the *Cube's* origin.



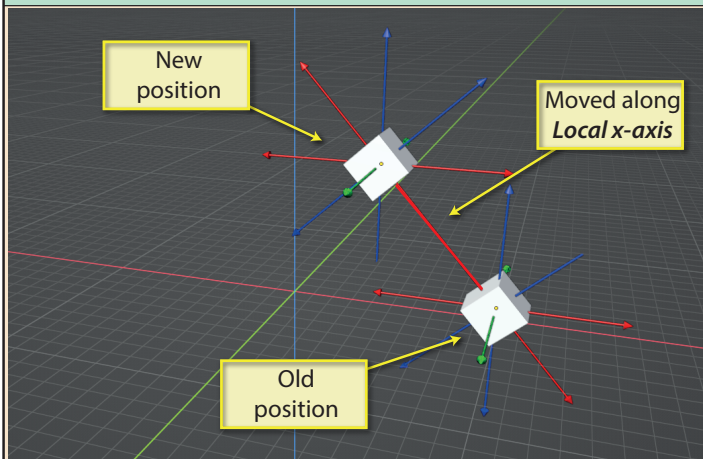
If we rotate the *Cube*, the *Global Axes* remain parallel to the *World Axes* while the *Local Axes* rotate with the *Cube*. Notice that the *Global* and *Local* y-axis (in green) still lie along the same path since it is that common axis that the *Cube* has been rotated



If we move the *Cube*, both the *Global* and *Local* axes will move with it, maintaining their origins at the centre of the *Cube*.



As we'll see later in the text, we can use any of these axes - **Global** or **Local** (or any other axes set) - when moving, rotating or resizing objects. Below, we can see the result of moving our previously rotated *Cube* along its *Local x-axis*.



An object's **View axes** have their orientation based on the screen's surface. The x and y axes are orientated in the traditional maths layout, x being horizontal and y vertical with the positive end of the z-axis coming out of the screen.

