

*Objects in JavaScript, just as in many other programming languages, can be compared to objects in real life. The concept of objects in JavaScript can be understood with real life, tangible objects. In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for.*

Find the object to add to the destination project by performing a search using the Search tab in the information window. When the search completes, on the search form, select the object to be added to the destination project. Verify that the destination project is highlighted in the project window. If it is not highlighted, click it. With the object to be added highlighted, click the Add Object or User to Project button in the center column. Find the objects to add to the destination project by performing a search using the Search tab in the information window. The system adds all of the objects that fit the search criteria to the project that you selected in step 1. Both projects and the object must be visible in your project window. This task can be used to move users from one project to another or to move a project to another project. This section discusses how to: Drag the object over the target project and release the mouse button. The system removes the object from the source project and adds it to the target project. From the Row menu, select Advanced, and then select Move Objects. In the detail area, click the objects that you want to move. The system moves the objects from the source project to the target project. This process might take several minutes, depending on the number of objects that you selected. This task also can be used to remove users from a project. Click the Remove Object or User from Project button in the center column. Alternatively, you can select the object, click the Row exit, and then click Remove Object. You can also mark an object for deletion from its transfer locations upon project advancement or from its current save location the location where the system saves the object when you click the Save button in the center column of JD Edwards EnterpriseOne OMW. You can also use this task to remove the specifications for Object Librarian object from your workstation. When you select Delete Object from Server, the system deletes the object from locations that are defined in the transfer activity rules when you click OK. If you select Mark Object to be Deleted from Transfer Locations, the system deletes the object from any other configured locations when the project advances. For an Object Librarian object, you can delete the local and save specifications. If the Object Librarian object is checked in, you can delete the checked-in version of this object by choosing Delete Object from Server. If you select Mark Object to be Deleted from Transfer Locations, the Object Librarian object is deleted from its transfer locations, which are defined in the transfer activity rules when the Project Status is advanced. A Delete form appears. Your available options vary depending on the object type and whether the object has been checked in. Select one or more of the following options, and then click OK: Delete Object from Server Click View Locations for a list of locations from which the object is deleted when you select this option. This action occurs as soon as you click OK. They are deleted from the transfer locations when the project status is advanced. Use this function when someone else has been working on the object and you want to see the changes, or when you have made changes to the object but want to abandon them in favor of another version of the object. The Get button enables you to get the specifications for objects that reside in your path code only. However, you can download the specifications of an object that resides in other areas of the system. For example, you might want to get the specifications for an object as it existed in a previous software release. Use the advanced get feature to specify the location of the object that you want to download. If you want to review the object and not save any changes, use the Get button to copy the latest specifications to your local workstation instead of checking out the object and then erasing the checkout. Click the Save button in the center column. The jdeomw dialog box displays. OMW provides two methods with which to save an object. The first method is to save the object to a zip file. The second method is to save the object to a shared database location. The shared location is configured by your OMW administrator, and is the most common method with which you will save objects or projects. Click Yes to save the object to a zip file, or click No to save the object to a pathcode save location. If you clicked Yes, continue to step 5. If you clicked No, OMW saves the object to pathcode save location. On the Select Folder form, select the folder to which you want to save the object, and then click Open. Click the Restore button in the

center column. The jdeomw dialog box displays and provides options for you to choose from where you want to restore the object. You can choose to restore the object from a zip file, or to restore the object from the Save location specified by the OMW administrator. Click Yes to restore the object from a zip file, or click No to restore the object from a pathcode save location. If you clicked Yes, continue to step 6. If you clicked No, OMW restores the object from the pathcode save location. On the Select Folder form, select the PAR file from which you want to restore the object or project, and then click Open. OMW restores the object or project. To perform an Advanced Get on Projects, every object in the project must reside in the pathcode from which you have selected the project. From the Row menu, select Advanced, and then select Advanced Get. You are prompted to decide whether you want to overwrite local specifications. Click one of the following options: Yes If you select Yes, go to step 5. No If you select No, continue with step 4. Select the location of the object that you want to get, and then click Select. From these forms, you can select one or many objects to Get. When you Get an object you copy the object and its specifications to your work area without checking out the object. Object Librarian is the only category for which Quick Gets is enabled. Select the objects you want to Get, click the Row menu, and then click Quick Gets. The Overwrite Specifications Confirmation form displays with the following options: Default " This option copies an object and its specifications from the configured checkout location to your work area without checking the object out. Advanced - Advanced Get on Projects enables you to get projects from any pathcode. Enter criteria for which you want to search, and then click Select. Additionally, you can access Quick Gets using the following method: From the Row menu, select Select by Object. Choose the objects you want to get, and then click the Row menu, and then click Quick Gets. From the Row menu, select Quick Gets. Default " This option copies an object and its specifications to your work area without checking the object out. Only one user at a time can check out an object. Checkout fails if the object is already checked out or if the token is unavailable. If the token is unavailable, you can join the token queue for the desired object. If you join the token queue, you will be notified when the token is available and your project will receive the token. Check in an object when you want to upload its specifications to the server and make it publicly available. When you check in an object, the system records the project in which the object resides and ensures that only changes made under the current project are transferred when the project is advanced to a status that triggers a transfer. If you move an object from one project to another using the drag-and-drop method, the system tracks the change and records the new project for the object. However, consider the following scenario: You add an object to a project and check it out. You change the object and check it in. You use the right-facing arrow in the center column to remove the object from the project. You later add the object to a different project. In this scenario, the system cannot track the object because it passes out of a project completely. When you drag-and-drop an object, the system updates its tables in such a way that the transfer can occur. This is not the case when you remove an object from a project and then add it to a different project later. If an object is checked out, you can erase the checkout. When you erase a checkout, local changes are not uploaded to the server. Erasing the checkout for an object does not release its token, but it does enable other developers who are assigned to the same project to check out the object. Click the Checkout button in the center column. Additionally, data about the object that appears in the information window is updated to reflect its checked out status. If the object is unavailable, the system asks if you want to be added to the token queue for the object. If you select to join the queue, the system alerts you when the token is released and assigns the token to your project. Additionally, if you have joined a token queue, your position in the queue appears here.

### Chapter 2 : Manning | Working with Objects

*Working with Objects. 06/05/; 2 minutes to read Contributors. In this article. We have discussed how Windows PowerShell uses objects to transfer data between cmdlets, and demonstrated a few ways to view detailed information about objects by using Get-Member and Format cmdlets to view particular properties of objects.*

The mouse pointer changes to a crosshair. Drag the crosshair from a starting point until the object is the desired size. Release the mouse button to end the drawing object and turn off the Drawing tool. Hold down the Shift key to create straight lines, perfect circles, and perfect squares. AutoShapes are inserted on their own layer with the In front of text wrapping style applied. Using WordArt, you can create text graphics that bend, slant, and appear metallic or wooden and much more. WordArt can even be shadowed, skewed, rotated, and stretched. Here are just a few examples of what WordArt allows you to do: Place the insertion point where you want to insert WordArt. Click the WordArt button on the Drawing toolbar. The WordArt gallery opens. The Edit WordArt Text dialog box appears. Edit the font, size, and style. To select several objects, hold down the Shift key and click each object, or use the Select Objects tool. Fill color allows you to color all selected drawing objects. No fill is the color white. Change the line color of a selected object. Change the text color of a selected object. Change the line style of a selected object. This includes solid and dotted lines. Change the style of arrow. Give selected objects some depth. Give selected objects a 3-D effect. Using both AutoShapes and WordArt, try to create a stop sign.

### Chapter 3 : Working with Objects and Object Points - Apache OpenOffice Wiki

*Working with Objects. 03/30/; 2 minutes to read Contributors. all; In this article. The Entity Framework enables you to query, insert, update, and delete data, which is expressed as typed common language runtime (CLR) objects that are instances of entity types.*

A factory method is used like this: Class factory methods usually just call straight through to alloc and the relevant init method, and are provided for convenience. You can create an NSString instance, for example, using a special literal notation, like this: You can also create an NSNumber using a boxed expression, like this: Objective-C Is a Dynamic Language As mentioned earlier, you need to use a pointer to keep track of an object in memory. The id type defines a generic object pointer. Consider the following code: Rewriting the code to use a static type: Because the class of an object is determined at runtime, it makes no difference what type you assign a variable when creating or working with an instance. When the sayHello method is called on each object, the correct implementations will be used; for secondPerson, this means the XYZShoutingPerson version. If you do send a message to nil, obviously nothing happens. If you expect a return value from a message sent to nil, the return value will be nil for object return types, 0 for numeric types, and NO for BOOL types. Returned structures have all members initialized to zero. If you need to check to make sure an object is not nil that a variable points to an object in memory , you can either use the standard C inequality operator: Similarly, if you need to check for a nil variable, you can either use the equality operator: As with any executable written in C, this function represents the starting point for your application. Add a variety of other greetings and call each of them on the instance you created above. Implement the XYZPerson class person factory method you declared in the previous chapter, to return a correctly allocated and initialized instance of the XYZPerson class, then use the method in main instead of your nested alloc and init. Rather than using [[XYZPerson alloc] init] in the class factory method, instead try using [[self alloc] init]. Test this by checking that: Use a branch if statement to check whether the variable is automatically assigned as nil. Terms of Use Privacy Policy Updated: Please try submitting your feedback later. Thank you for providing feedback! Your input helps improve our developer documentation. How helpful is this document?

## Chapter 4 : 3D visualization: working with objects, paths, and extrusion

*Working with Objects* The majority of work in an Objective-C application happens as a result of messages being sent back and forth across an ecosystem of objects. Some of these objects are instances of classes provided by Cocoa or Cocoa Touch, some are instances of your own classes.

They can replace tedious alternatives like hardcoding coordinates like spawn points in your source code or maintaining additional data files for storing gameplay elements. To start using objects, add an Object Layer to your map. They are useful for marking rectangular areas and assigning custom properties to them. They are also often used for specifying collision boxes. Place a rectangle by clicking-and-dragging in any direction. Holding Shift makes it square and holding Ctrl snaps its size to the tile size. If the rectangle is empty width and height are both 0, it is rendered as a small square around its position. This is mainly to keep it visible and selectable. I Points are the simplest objects you can place on a map. They only represent a location, and cannot be resized or rotated. Simply click on the map to position a point object. C Ellipses work the same way as rectangles, except that they are rendered as an ellipse. Useful for when your area or collision shape needs to represent a circle or ellipse. P Polygons are the most flexible way of defining the shape of an area. They are most commonly used for defining collision shapes. When placing a polygon, the first click determines the location of the object as well as the location of the first point of the polygon. Subsequent clicks are used to add additional points to the polygon. Right click or press Enter to finish creating the polygon. Polygons needs to have at least three points. You can press Escape to cancel the creation of the polygon. When you want to change a polygon after it has been placed, you need to use the Edit Polygons tool. L Polylines work very similar to polygons, except that they are rendered as a line and require only two points. While they can represent collision walls, they are also often used to represent paths to be followed. Despite its name, the Edit Polygons tool is also used to edit polylines. T Tiles can be inserted as objects to have full flexibility in placing, scaling and rotating the tile image on your map. Like all objects, tile objects can also have custom properties associated with them. This makes them useful for placement of recognizable interactive objects that need special information, like a chest with defined contents or an NPC with defined script. To place a tile object, first select the tile you want to place in the Tilesets view. Then use the Left mouse button on the map to start placing the object, move to position it based on the preview and release to finish placing the object. New in Tiled 1. V Can be used to quickly insert multiple instances of the template selected in the Templates view. See Creating Template Instances. X Text objects can be used to add arbitrary multi-line text to your maps. It packs a lot of functionality, which is outlined below. When pressing and dragging on an object, this object is selected and moved. When this prevents you from starting a rectangular selection, you can hold Shift to force the selection rectangle. When you need to select an object below another object, first select the higher object and then hold Alt while clicking at the same location to select lower objects. You can also hold Alt while opening the context menu to get a list of all objects at the clicked location, so you may directly select the desired object. Hold Ctrl to toggle snapping to the tile grid. Hold Alt to force a move operation on the currently selected objects, regardless of where you click on the map. This is useful when the selected objects are small or covered by other objects. The selected objects can also be moved with the arrow keys. By default this moves the objects pixel by pixel. Hold Shift while using the arrow keys to move the objects by distance of one tile. Note that you can only change width and height independently when resizing a single object. When having multiple objects selected, the aspect ratio is constant because there would be no way to make that work for rotated objects without full support for transformations. Before rotating, you can drag the rotation origin to another position if necessary. Hold Shift to rotate in degree increments. Click any selected object again to go back to resize mode. When you have multiple Object Layers, the context menu also contains actions to move the selected objects to another layer. For tile objects, this also flips their images. E Polygons and polylines have their own editing needs and as such are covered by a separate tool, which allows selecting and moving around their nodes. You can select and move the nodes of multiple polygons at the same time. The Delete key can also be used to delete the selected nodes, or the selected objects if no nodes are selected. This will convert

a polygon into a polyline. Future Extensions Here are some ideas about improvements that could be made to the above tools: For the Insert Tile tool, show the preview already before pressing the left mouse button Many improvements could be made to the support for editing polygons and polylines, like allowing to rotate and scale the selected nodes If you like any of these plans, please help me getting around to it faster by becoming a patron. The more support I receive the more time I can afford to spend improving Tiled!

## Chapter 5 : Working with Objects

*Understanding Object Addition. An object must exist within one of your projects before you can work with it. You can add an existing object to a project, or you can create a new object for a project.*

When you do work in Java, you primarily use objects to get the job done. You create objects, modify them, change their variables, call their methods, and combine them with other objects. You develop classes, create objects out of those classes, and use them with other classes and objects. Today, you work extensively with objects as you undertake these essential tasks: These objects, which also are called instances, are self-contained elements of a program with related features and data. For the most part, you use the class merely to create instances and then work with those instances. In this section, you learn how to create a new object from any given class. The String class is unusual in that respect. This shortcut is available only for strings and classes that represent primitive data types, such as Integer and Double. To create instances for all other classes, the new operator is used. NOTE What about the literals for numbers and characters? Using new To create a new object, you use the new operator with the name of the class that should be used as a template. The name of the class is followed by parentheses, as in these three examples: They can be empty, however, in which case the most simple, basic object of that class is created. The parentheses also can contain arguments that determine the values of instance variables or other initial qualities of that object. Here are two objects being created with arguments: If you try to create a new instance of a class with the wrong number or wrong type of arguments, or if you give it no arguments and it needs them, an error occurs when the program is compiled. The StringTokenizer class in the java. You divide a string into tokens by applying a character or characters as a delimiter. This program creates StringTokenizer objects by using new in two different ways and then displays each token the objects contain. StringTokenizer st1, st2; 9: Run the application by choosing Run, Run File to see the output displayed in Figure 3. Two different StringTokenizer objects are created using different arguments to the constructor. The first object is created using new StringTokenizer with one argument, a String object named quote1 line This creates a StringTokenizer object that uses the default delimiters, which are blank spaces, tabs, newlines, carriage returns, or formfeed characters. If any of these characters is contained in the string, it is used to divide the string. Because the quote1 string contains spaces, these are used as delimiters dividing each token. Lines 12-14 display the values of all three tokens: The second StringTokenizer object in this example has two arguments when it is constructed in line 16-a String object named quote2 and an at-sign character. This second argument indicates that the character should be used as the delimiter between tokens. The StringTokenizer object created in line 17 contains three tokens: How Objects Are Constructed Several things happen when you use the new operator. The new instance of the given class is created, memory is allocated for it, and a special method defined in the given class is called. This method is called a constructor. A constructor is a way to create a new instance of a class. A constructor initializes the new object and its variables, creates any other objects that the object needs, and performs any additional operations the object requires to initialize itself. A class can have several different constructors, each with a different number or type of argument. When you use new, you can specify different arguments in the argument list, and the correct constructor for those arguments is called. In the TokenTester project, multiple constructor definitions enabled the StringTokenizer class to accomplish different things with different uses of the new operator. When you create your own classes, you can define as many constructors as you need to implement the behavior of the class. No two constructors in a class can have the same number and type of arguments, because this is the only way constructors are differentiated from each other. If a class defines no constructors, a constructor with no arguments is called by default when an object of the class is created. The only thing this constructor does is call the same constructor in its superclass. A Note on Memory Management If you are familiar with other object-oriented programming languages, you might wonder whether the new operator has an opposite that destroys an object when it is no longer needed. Memory management in Java is dynamic and automatic. When you create a new object, Java automatically allocates the proper amount of memory for that object. Under most circumstances, when you are finished with an object you have created,

## DOWNLOAD PDF WORKING WITH OBJECTS

Java can determine that the object no longer has any live references to it. As a program runs, the JVM periodically looks for unused objects and reclaims the memory that those objects are using. This process is called dynamic garbage collection and occurs without any programming on your part.

### Chapter 6 : Working with Objects | Microsoft Docs

*Working With Objects is the source on a method which takes an evolutionary step forward in object-oriented development practices. OOram adds the intuitively simple but powerful concept of a role to object-oriented models and methods.*

**Resources** **Objects** The secret to being successful in MS Publisher is understanding how to manipulate objects. Each text box, line, graphic, header, etc.,. All objects have several characteristics that enable them to be changed in regards to position, size, stacking arrangement, and to sometimes change the contents inside the object. When you click on any object you will immediately notice several things. First, a border forms around the object with four white circles and four white squares. When you click on any of the circles you are able to pull or push the object to change size, with the image keeping its original aspect ratio. If you push or pull on the white squares it will change the size of the object, but it will not keep the original aspect ratio. The white circle at the top enables the user to rotate the object to any angle. Objects may be grouped together by clicking on one object, then holding down the CTRL key and selecting additional objects. When the group symbol is clicked on the Ribbon then all of the objects are transformed into a single object. This can be extremely useful if a large number of objects need to be repositioned throughout the publication. Likewise, if you click on the symbol when the objects are grouped together, you can ungroup them. Once the ungrouping symbol has been clicked, you can then click on the individual objects and change them. Most of the pre-designed templates in Publisher contain many grouped objects, such as pictures and captions, so knowing how to use this feature is essential. You can also right click on any object and a dialog box with various options will open up, which can sometimes be faster than searching for the right option in the menus at the top of the workspace. From these options you can change the graphic, delete the object, group objects, copy, paste, change the stacking order, and more.

**More With Objects** The other important thing to note when working with objects is to use the Ribbons effectively. When you click on an object it is critical to know that you are not confined just to that Ribbon. For example, when you click on a text box for example, you can also then click on the purple Text Box Tools tab at the top of the Ribbon. The Ribbon will then change to show you a wide range of tools that can be used to configure the text in that box. There are options for fonts, font size, alignment, WordArt, and more. Also, you can select the orange Drawing Tools tab which will then allow you to add shapes, borders, and effects to your text box. When working with an image object the same principals apply in that you are not confined to the Ribbon that is showing when the object is clicked on. You can click on the light purple Picture Tools tab located at the top of the Ribbon. By clicking on the various options you can do basic image editing and formatting, add effects, captions, and wrap text.

### Chapter 7 : Working with Amazon S3 Objects - Amazon Simple Storage Service

*Or click the object name or style icon in the left menu bar, or left-click the object on the map, and then click "Edit" on the pop-up window. Or click "Bulk Ops" below the list of objects, select the desired object(s)\* and click "Change Attributes".*

Number of resources in the hits. Guaranteed to be greater than 0 and less than or equal to 1. The query builder creates a `CompassQuery` which can then be used to add sorting and executing the query. Using the `CompassQueryBuilder`, simple queries can be created i. The following code shows how to use a query string query builder and using the `CompassQuery` add sorting to the result. Also sorting requires more memory to keep sorting properties available. For numeric types, each property sorted requires four bytes to be cached for each resource in the index. For String types, each unique term needs to be cached. If we take the following mapping for example: With the `CompassQueryBuilder`, most of the queries can directly work with either level of the mappings. Here are some samples: This can be used to build a frequency based list of terms showing how popular are different tags as different blogging sites do for example. Here is a simple example of how it can be used: `CompassSearchHelper` `Compass` provides a simple search helper providing support for pagination and automatic hits detach. The search helper can be used mainly to simplify search results display and can be easily integrated with different MVC frameworks. `CompassSearchHelper` is thread safe. Here is an example of how it can be used: `Core` comes with the `CompassHighlighter` interface. It provides ways to highlight matched text fragments based on a query executed. The following code fragment shows a simple usage of the highlighter functionality please consult the javadoc for more information: When working with pure hits results, `CompassHits` can be detached, and then used outside of a transactional context, the question is: Each highlighting operation as seen in the previous code is also cached within the hits object. When detaching the hits, the cache is passed to the detached hits, which can then be used outside of a transaction. Here is an example:

### Chapter 8 : Word XP: Working with Objects

*Video created by Rice University for the course "An Introduction to Interactive Programming in Python (Part 2)". Learn the basics of object-oriented programming in Python using classes, work with tiled images Learn online and earn valuable.*

You store these objects in one or more buckets. An object consists of the following:

- Key** – The name that you assign to an object. You use the object key to retrieve the object.
- The version ID** is a string that Amazon S3 generates when you add an object to a bucket. For more information, see [Object Versioning](#).
- Value** – The content that you are storing. An object value can be any sequence of bytes. Objects can range in size from zero to 5 TB. For more information, see [Uploading Objects](#).
- Metadata** – A set of name-value pairs with which you can store information regarding the object. You can assign metadata, referred to as user-defined metadata, to your objects in Amazon S3. Amazon S3 also assigns system-metadata to these objects, which it uses for managing objects. For more information, see [Object Key and Metadata](#).
- Subresources** – Amazon S3 uses the subresource mechanism to store object-specific additional information. Because subresources are subordinates to objects, they are always associated with some other entity such as an object or a bucket. For more information, see [Object Subresources](#).

Amazon S3 supports both the resource-based access control, such as an Access Control List ACL and bucket policies, and user-based access control. For more information about working with objects, see the following sections. Your Amazon S3 resources for example buckets and objects are private by default. You need to explicitly grant permission for others to access these resources. For example, you might want to share a video or a photo stored in your Amazon S3 bucket on your website. That works only if you either make the object public or use a presigned URL on your website. For more information about sharing objects, see [Share an Object with Others](#).

## Chapter 9 : ES6 -Working with Objects “ The Chronicles of Nerdia

*WordArt drawing objects. Also included on the Drawing toolbar is the WordArt [www.nxgvision.com](http://www.nxgvision.com) WordArt, you can create text graphics that bend, slant, and appear metallic or wooden and much more.*

Apart from color, size, and shape of the symbols, one must decide whether to drape the symbols or billboard them. The following table summarizes each symbol layer type and which 3D symbols they may be applied to. Creating a volumetric symbol layer is fundamentally the same as creating a flat symbol layer with two major exceptions. The size of volumetric symbols is always defined in meters, not points or pixels. Defining size The three volumetric symbol layer types are always defined in 3D real-world space, not in screen space as is the case with their flat symbol layer counterparts. Therefore, unlike billboarded flat icons, the size of 3D objects, paths, and extrusions appear either larger or smaller based on their distance from the camera. In the following sample, we use a sphere primitive in an `ObjectSymbol3DLayer` to symbolize world cities. Notice the size is set to a value of 50, m. This can be done to improve the above sample by varying the size of the spheres based on the population of each city. The same principles for defining visual variables in 2D apply to defining them in 3D. In 3D, however, the perception of size can be problematic. Because of perspective, it is difficult to process spatial size differences between features. For example, suppose you are looking at two buildings that are both feet tall. One is about yards in front of you and the other is one mile away. The building located a mile away will appear much smaller than the one only feet away even though both are the same size. The same principle makes it difficult to judge sizes in 3D rendering. Adding a color ramp for the same attribute helps us more easily identify the spatial patterns that would otherwise be difficult to understand. Of course you have the liberty to use color for a second attribute instead of the same one used for size, but that is ultimately up to you as the developer and cartographer. Working with the axis property in size visual variables Setting a single size value on an `ExtrudeSymbol3DLayer` and a `PathSymbol3DLayer` suffices since the extrusion on polygons only impacts height, and paths use a single size for setting the diameter of a tube. But working with 3D objects can be more complex. Objects like cylinders and pyramids have three axes to consider: The axis property on the size visual variable allows us to define each of these values. In the hurricane sample below, we set a constant size on the width and depth axes to make all cylinder diameters the same size 50km. In the following sample two attributes are mapped with size on different axes: Based on the way the visual variables are defined, we would expect areas with a high percentage of democrats to have tall, deep blue, skinny cylinders since conservatives tend to favor the Republican party. That certainly appears to be the case in areas like San Francisco and New York. Interestingly, we see a few pockets of low population areas where there appears to be a higher percentage of conservatives and democrats. Limitations While using the axis in size to depict more than one attribute can pave the way for some complex and fascinating visualizations, it can also create very complicated, misleading, and confusing scenes. As is the case when working with other visual variables for thematic purposes, it is best to stay as simple as possible. While the possibilities appear to be endless, a general rule of thumb is maps that visualize more than two or three attributes at once become very difficult to interpret. Therefore, while visual variables can be powerful, be sure to use them with caution in thematic mapping. Because 3D symbol layers use meters to render the width, depth, and height of features, you can use visual variables to map the sizes of features as they exist in the real world. This is where the axis property on the `sizeInfo` object becomes particularly powerful.