

## Naming conventions

Now please take a moment to familiarize yourself with the naming conventions. Names you give to the model objects are important. A good naming system simplifies development process a lot. We recommend you to keep to Java naming conventions throughout the model, not just when writing Java code.



**The name of the object should indicate the intent of its use.**

Ideally a casual observer should be able to understand the role of the object from its name and, on the other hand, when looking for an object serving a particular purpose, he should be able to easily guess its name.



**You can compose a name of several words. Use mixed case with the first letter of each word capitalized. Do not use underscores.**

You should never use meaningless names like “a”, “b”, “x23”, “state1”, “event14”. One-character variable names should only be used for temporary “throwaway” variables such as loop indexes. Avoid acronyms and abbreviations unless they are more common than the long form, for example: NPV, ROI, ARPU. Remember that when using the object name in an expression or code you will not need to type it: AnyLogic [code completion](#) will do the work for you, therefore multi-word names are perfectly fine.



**Keep to one naming system. Names must be uniformly structured.**

It makes sense to develop a naming system for your models and keep to it. Large organizations sometimes standardize the naming conventions across all modeling projects.

A couple of important facts: Java is a case-sensitive programming language: Anylogic and AnyLogic are different names that will never match. Spaces are not allowed in Java names.

In the Table below we summarize the naming recommendations for various types of model objects.

Object	Naming rules	Examples
<ul style="list-style-type: none"><li>Java variable</li><li>Parameter of agent</li><li>Collection</li><li>Table function</li><li>Statistics</li></ul>	<p>First letter can be lowercase or uppercase (here we relax Java conventions), first letter of each internal word capitalized.</p> <p>Should be a noun.</p> <p>Use plurals for collections.</p> <p>Sometimes adding a suffix or prefix indicating the type of the object helps to understand its meaning and to avoid name conflicts. For example, <code>AgeDistribution</code> can be a custom distribution constructed from the <code>table</code> function <code>AgeDistributionTable</code>.</p>	<pre>rate Income DevelopmentCost inventory AgeDistribution AgeDistributionTable friends</pre>
<ul style="list-style-type: none"><li>Function</li></ul>	<p>First letter must be lowercase, first letter of each internal word capitalized.</p> <p>Should be a verb.</p> <p>If the function returns a property of the object, its name should start with the word “get”, or “is” for boolean return type. If the function changes a property of the object, it should start with “set”.</p>	<pre>resetStatistics getAnnualProfit goHome speedup getEstimatedROI setTarget</pre>

	There are some exceptions created to make the code more compact, such as AnyLogic system functions <code>time()</code> and <code>date()</code> , or functions <code>size()</code> of Java collection.	<code>inState</code> <code>isVisible</code> <code>isEnabled</code>
<ul style="list-style-type: none"> <li>Function argument</li> <li>Local variable in the code</li> </ul>	<p>If possible, short, lowercase. If consists of more than one word, first letter of each internal word should be capitalized</p> <p>Common names for temporary integer variables are <code>i</code>, <code>j</code>, <code>k</code>, <code>m</code>, and <code>n</code>.</p>	<code>cost</code> <code>sum</code> <code>total</code> <code>baseValue</code> <code>i</code> <code>n</code>
<ul style="list-style-type: none"> <li>Java constant</li> </ul>	All uppercase with words separated with underscores “_”.	<code>TIME_UNIT_MONTH</code>
<p>Classes:</p> <ul style="list-style-type: none"> <li>Agent type</li> <li>User's Java class</li> <li>Dynamic event</li> </ul>	<p>First letter <i>must</i> be capitalized, first letter of each internal word capitalized as well.</p> <p>Should be a noun except for process model components that have a meaning of action, in which case it can be a verb.</p>	<code>Consumer</code> <code>Project</code> <code>UseNurse</code> <code>RegistrationProcess</code> <code>HousingSector</code> <code>PhoneCall</code> <code>Order</code> <code>Arrival</code>
<ul style="list-style-type: none"> <li>Agent population, including the library objects</li> </ul>	<p>First letter must be lowercase, first letter of each internal word capitalized.</p> <p>Use plurals for replicated objects.</p>	<code>project</code> <code>consumers</code> <code>people</code> <code>doTriage</code> <code>registrationProcess</code> <code>stuffAndMailBill</code>
<p>Dynamic variables:</p> <ul style="list-style-type: none"> <li>Stock</li> <li>Flow</li> <li>Dynamic variable</li> </ul>	<p>Long multi-word variable names are very common in system dynamics models. As in Java and in AnyLogic spaces are not allowed in names, you should use other ways to separate words. We recommend to use mixed case with the first letter of each word capitalized. The use of underscore “_” is not recommended, although it is allowed.</p>	<code>BirthRate</code> <code>Population</code> <code>DrugsUnderConsideration</code> <code>TimeToImplementStrategies</code>
<ul style="list-style-type: none"> <li>Events (not dynamic)</li> <li>Statechart</li> <li>States</li> <li>Transition</li> </ul>	First letter can be lowercase or uppercase, first letter of each internal word capitalized.	<code>overflow</code> <code>at8AMeveryDay</code> <code>purchaseBehavior</code> <code>InWorkForce</code> <code>discard</code>