This tutorial covers the fundamentals of data input masking in Access 2013.

Data input masks use codes to dictate the type and layout of information input into a field. For example, a social security number always consists of 3 digits, a hyphen, 2 digits, another hyphen, and 4 digits. We might, then, want to create a field which looks like this ___-__-___, and would allow only digits to be entered into the provided placeholders (in this case, an underscore, ",_").

**Where to set up the Input Mask**

From **Design View**, add the desired field name and data type.

When you select the **Data Type** menu, the **Field Properties** form will appear at the bottom of the window. In the first tab (**General**) of that form, there is a line for **Input Mask**. The mask code will be entered there.

**The Basic Mask Rules (Syntax)**

To create the input mask in the above example, we can use the the mask code `000\-000\-000;1;_`. This code phrase has three sections separated by semicolons.
Section 1: The layout and format of the input mask

Here the zeros dictate that the user must input a digit in that place. The hyphens are simply the hyphens we see in the mask. They are preceded by a back-slash (\) so we know they are literal characters and not part of the mask code.

Section 2: This section governs whether or not the literal characters, in this case the hyphens (-) will be stored along with the data input into the field.

The number "1" indicates that the literal characters are NOT to be stored, and the number "0" indicates that the characters are to be stored.

Section 3: In this 3rd section, we provide the placeholder character that will show up in the field to denote the expected entries.

Here, we used the underscore (which is also the default character). This will produce a pre-entry display of ____, ____. We could have put a pound sign (#) in this position to produce ####\-####-####.

The Code Character Meanings (Lexical Semantic)

In our example, we used the zero (0) character to allow only digits to be entered in that place. This code character also denotes that an entry must be made. If a nine (9) was used (e.g. 999\-99\-9999;1;_) then no entry would be required in that position, but if one is made, it must be a digit.

<table>
<thead>
<tr>
<th>Character Code</th>
<th>Denotes</th>
<th>Mask Example</th>
<th>Produces</th>
<th>Accepted Input Examples</th>
<th>Rejected Input Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Digit</td>
<td>(000)&quot;000-0000;_\</td>
<td>(__) ____</td>
<td>(123) 867-5309</td>
<td>(+12) 867-5309 (916) cal-turk (967-8675)</td>
</tr>
<tr>
<td>9</td>
<td>Digit</td>
<td>(999)&quot;000-0000;#\</td>
<td>(###) #######</td>
<td>(123) 867-5309</td>
<td>The letters are invalid. The dashes are invalid.</td>
</tr>
<tr>
<td>#</td>
<td>Digit or plus or minus</td>
<td>###############;_\</td>
<td>044 8675 3091</td>
<td>04  8675 309</td>
<td>The 1st set requires 4 digits.</td>
</tr>
<tr>
<td>L</td>
<td>Letter ([a - z]) or ([A - Z]) Entry required</td>
<td>LLL000;_\</td>
<td>***** (CS168) (Hwy101)</td>
<td>(CS143) (CS143) ENGL98 (Mr.007)</td>
<td>First 3 characters must be numbers, and the last 3 must be digits</td>
</tr>
<tr>
<td>?</td>
<td>Letter ([a - z]) or ([A - Z]) Entry optional</td>
<td>LL?000;_\</td>
<td>***** (PHIL120) (CS 143) (ENG 098)</td>
<td>Only the 1st two letters are required. (ENGL 98) (Mr.007) (Jun2004)</td>
<td>All 3 digits are required. The period is invalid. A digit cannot be use in a letter place.</td>
</tr>
<tr>
<td>A</td>
<td>Letter or digit ([a - z]) or ([A - Z]) or ([0 - 9]) Entry required</td>
<td>AAAAA;AAAAAA;_\</td>
<td>- _______ (K743-JMP55QR6) Code-Cruncher 1234-56789010</td>
<td>K-1234abcd</td>
<td>4 letters are required before the dash.</td>
</tr>
<tr>
<td>a</td>
<td>Letter or digit ([a - z]) or ([A - Z]) or ([0 - 9]) Entry optional</td>
<td>aaaaaaaaa;_\ |***** (UserName) (User1234) (User 21)</td>
<td>Entries are optional so spaces are okay. (Me@large)</td>
<td>The &quot;@&quot; is invalid.</td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td>Any character Entry required</td>
<td>&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;&amp;;_\</td>
<td>Poe**EA[49R]</td>
<td>Fox M J 662GT</td>
<td>The spaces are invalid.</td>
</tr>
</tbody>
</table>
Examples of Setting Up an Input Mask

Example 1:

Sections 2 and 3 of the mask code phrase are not required, so for a simple mask, we need only supply the format code. In this case, we want a required 2 letter state entry. The "L" code denotes a required letter entry, so we simply enter LL on the Input Mask line.

Access supplies the default placeholders ( _ ), and so we get what we expect, __.

Because the "L" code denotes a required entry, if we do not supply 2 letters, we get an error, and we physically cannot enter more than 2 characters.
Example 2:

US zip codes have a 4 digit suffix that most people ignore. For this reason, we may not wish to require that this suffix be entered. We can use the nine (9) code character in these places to make that data optional.

Unlike the required input code character used above, when we skip over the last 4 entries in this example, we do NOT get an error. We simply move on to the next data field.