Learning AIML isn't that difficult. It is a simple and great way to create a chat bot for your site! This tutorial is based upon the tutorial by Dr. Richard S. Wallace, and provides a really simplistic view of how AIML works. It also serves as a good starter guide for those who want to have your own bot, but don't want to go through the hassle of reading through long articles.

The Basics

To understand how AIML works, we must know that the basic unit used to described AIML is *categories*.

Each *category* contains a *pattern*, and a *template*.

> *Pattern* is a string that the user will ask. In short, it is your question.
>
> *Template* is a string that your chat bot will use to reply. In short, it is your answer.

How it works?

When visitor enters a question, the chat bot will search through the pattern of each category. When it finds a pattern which match, it would reply what ever it finds in the template.

Example

```
<aiml>
    <category>
        <pattern>Who are you</pattern>
        <template>I am Alice the chatbot!</template>
    </category>
</aiml>
```

The Intermediate

By now, you should be comfortable with the making your chat bot answer simple question. But you would have also noticed other problems. Everyone ask a question in a different way. For example, "*Hello* there", and "*Hi there*".

Computers, being just 1s and 0s, would not be able to tell that the two questions are actually the same. Hence, we make use of additional tag in the *template* tag. It's called the *srai* tag.

The *srai* tag can redirect the *pattern* matched at one *category*, to another *category*.

Example

```
<aiml>


    <category>
        <pattern>Hello</pattern>
        <template>Hi, How are you? I am Alice the chat
bot.</template>
    </category>


    <category>
```

```
        <pattern>Hi</pattern>
        <template><srai>Hello</srai></template>
    </category>

</aiml>
```

How it works?

Following the above example, when the user first say "*Hello*", the bot picks up the answer, "*Hi, How are you? I am Alice the chat bot*".

If, however, the user say "*Hi*", the bot would encounter an *srai* tag. The bot will then look for another *category*, with a *pattern* that matches what was specified in the *srai* tag.

In this case, it finds the same answer, "*Hi, How are you? I am Alice the chat bot*".

The Wild Card

By now, you should be familiar with using the *srai* tag. You might find that using the *srai* tag, you can group similar questions together.

Consider this problem: People can ask about an item in several ways. Example, "*What is table?*", or "*What is the table*", or "*What is the definition of table*", or "*Tell me about table.*". You know you can group all of them to the same definition using the *srai* tag.

But then, multiply that by the number of items you have to define. You would have to do the entire set of same question for chair, sofa, television, bed, wardrobe, and many more.

So, instead of repeating it many times over, we can use a wild card to represent the item.

Example

```
<aiml>
    <category>
        <pattern>What is *</pattern>
        <template><srai>DEFINE <star/></srai></template>
    </category>
    <category>
        <pattern>Tell me about *</pattern>
        <template><srai>What is <star/></srai></template>
    </category>

    <category>
        <pattern>DEFINE Table</pattern>
        <template>A table is a furniture for you to work
on.</template>
    </category>
    <category>
        <pattern>DEFINE Chair</pattern>
        <template>A chair is a furniture for you to sit
on.</template>
    </category>
</aiml>
```

How it works?

When you ask a question, it starts with "*What is*", and ends with a word. The pattern is matched with the "*What is **". Any string after "*What is*" is matched to the "*".

The wild card now stores the string. In the *template* tag, it calls the *srai* tag, and the string in wild card is being represented in the *star* tag.

Hence, a input "*What is chair*", will be the same as "*DEFINE chair*", which will give the same output, "*A chair is a furniture for you to sit on.*".

Author's Note:

The wild card may be difficult to comprehend for most people, especially people with no programming background. I would suggest you try the above example a few times and play around with it more. It might help in your understanding of the wild card and the *star* tag.