Looks the same as a variable

<xsl:param name="phone">555-1234</xsl:param>
<xsl:param>

XSL Parameters
Pass parameters to template

```xml
<xsl:call-template name="phoneNum">
   <xsl:with-param name="phone" select="555-1234" />
</xsl:call-template>

<xsl:template match="phoneNum">
   <xsl:param name="phone" />
   ...
</xsl:template>
```
Variable vs. Param

- Variables
  - Constants (values cannot be changed)
  - Only way they can be changed is by declaring it inside of a for-each loop, in which case its value is updated for each iteration.
Variable Example

```xml
<xsl:variable name="mailLink">
  <a href=mailto:{$email}>
    <xsl:value-of select="$email" />
  </a>
</xsl:variable>
```

Notice the use of { and }
- Used because XML will not allow you to nest angle brackets `<>
- If you tried to use value-of instead of `{ }` then you would get an error
Finally, notice that the above variable has tags in it. It has the anchor tag <a>

To copy this variable value, it would look like this

<p>Please contact us at
  <xsl:copy-of select="$mailLink" />
</p>

Notice the use of copy-of

Use copy-of when there are elements (tags) as part of the value of a variable.
XSL Attributes

```
<book id="1234567">
   <author></author>
</book>
<book id="7654321">
   <author></author>
</book>

Select the ID of each book

<xsl:for-each select="book">
   <b><xsl:value-of select="@id" /></b>
</xsl:for-each>

Use the @ symbol to access attribute values
Create a Variable

■ Variable Creation

```xml
<xsl:variable name="email">rjglotzbach@tech</xsl:variable>

<xsl:variable name="subject">Wrox Press</xsl:variable>

<xsl:variable name="address">123 N. South St.</xsl:variable>

To get the value of the variable
```xml
<xsl:value-of select="$email"/>

Precede the variable name with $
Parameters

- Assigned a value from `<xsl:with-param>`

- Can be assigned just like a variable (as in the preceding example), in which case, it behaves just like a variable.
If time permits...