*The first version of HTML page:*

*# The total HTML.*

PAGE\_HTML = """

<p>Welcome, {name}!</p>

<p>Products:</p>

<ul>

{products}

</ul>

"""

*# The HTML for each product displayed.*

PRODUCT\_HTML = "<li>{prodname}: {price}</li>\n"

**def** make\_page (username, products):

product\_html = ""

**for** prodname, price in products:

product\_html += PRODUCT\_HTML.format(

prodname=prodname, price=format\_price(price))

html = PAGE\_HTML.format(name=username, products=product\_html)

**return** html

1. static text is broken into separate pieces

front-end:前端的

1. The better way to produce HTML pages is with *templates*

The second version of HTML:

**<p>**Welcome, {{user\_name}}!**</p>**

**<p>**Products:**</p>**

**<ul>**

{% for product in product\_list %}

**<li>**{{ product.name }}:

{{ product.price|format\_price }}**</li>**

{% endfor %}

**</ul>**

1. The static pieces are indicated by the double-quote notation.(双引号之内的都是静态)
2. To use HTML templates in our programs, we need a *template engine*: a function that takes a static template describing the structure and static content of the page, and a dynamic *context* that provides the dynamic data to plug into the template.

Supported Syntax

1. Template engines use the dot that will access object attributes or dictionary values
2. We can use *filters* to modify values

**<p>**Short name: {{story.subject|slugify|lower}}**</p>**