Stepping through a ‘for’ loop
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
Start with the first line - the ‘for’ statement

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name 
print "The end."
Look at the list

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
  print "Hello," , name  
print "The end."
Is it empty? No. Start with the first object

```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
```
Assign the first object to the variable ‘name’

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."

Create the string object “Andrew” and assign it to the variable named ‘name’
Then start the first line of the code block

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name 
print "The end."
```
This is the ‘print’ statement

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello,", name
print "The end."
```
print the string object “Hello,” and the value of the variable with name ‘name’

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."

Variable Names

Object

name

“Andrew”

Hello, Andrew
The print statement is finished. Python is at the end of the code block...

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```
... so go to the ‘for’ statement again

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello,", name  
print "The end."
Is it past the end? No, it’s on the second item.

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```
Assign the second object to the variable ‘name’

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```

Create the string object “Tsanwani” and assign it to the variable named ‘name’

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>“Andrew”</td>
</tr>
<tr>
<td></td>
<td>“Tsanwani”</td>
</tr>
</tbody>
</table>

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]=

print "Hello," , name
print "The end."

Then start the first line of the code block

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>&quot;Andrew&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Tsanwani&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
print the string object “Hello,” and the value of the variable with name ‘name’

```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," + name
print "The end."
```

Variable Names

- `name`

Objects

- “Andrew”
- “Tsanwani”

Hello, Andrew
Hello, Tsanwani
The print statement is finished. Python is at the end of the code block...

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]: 
    print "Hello," , name 
print "The end."
```
... so go to the ‘for’ statement again

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
```
Move the list pointer forward by one

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
Is it past the end? No, it’s on the third item.

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello,", name
print "The end."
```
Assign the third object to the variable ‘name’

for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello,", name  
print "The end."

Create the string object “Arno” and assign it to the variable named ‘name’
Then start the first line of the code block

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```

Variable Names

Objects

<table>
<thead>
<tr>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>“Andrew”</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Tsanwani”</td>
</tr>
<tr>
<td>“Arno”</td>
</tr>
</tbody>
</table>
print the string object “Hello,” and the value of the variable with name ‘name’

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print “The end.”
```

Variable Names

- name

Objects

- “Andrew”
- “Tsanwani”
- “Arno”

Hello, Andrew
Hello, Tsanwani
Hello, Arno
The print statement is finished.
Python is at the end of the code block...

```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello,", name  
print "The end."
```

Variable Names

<table>
<thead>
<tr>
<th>name</th>
</tr>
</thead>
</table>

Objects

<table>
<thead>
<tr>
<th>“Andrew”</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Tsanwani”</td>
</tr>
<tr>
<td>“Arno”</td>
</tr>
</tbody>
</table>
... so go to the ‘for’ statement again

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```
Move the list pointer forward by one

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```
Is it past the end? No, it’s on the fourth item.

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```
Assign the fourth object to the variable ‘name’

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello,", name  
print "The end."
```

Create the string object “Tebogo” and assign it to the variable named ‘name’
Then start the first line of the code block

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name 
print "The end."
```
print the string object “Hello,” and the value of the variable with name ‘name’

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," , name
print "The end."
```

Variable Names

<table>
<thead>
<tr>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

Objects

- “Andrew”
- “Tsanwani”
- “Arno”
- “Tebogo”

Hello, Andrew
Hello, Tsanwani
Hello, Arno
Hello, Tebogo
The print statement is finished. Python is at the end of the code block...

```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
```
... so go to the ‘for’ statement again

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello," name
print "The end."
```
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name 
print "The end."
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name  
print "The end."
So skip past the code block to the next statement.

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:    
    print "Hello," , name

print "The end."
```

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>&quot;Andrew&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Tsanwani&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Arno&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Tebogo&quot;</td>
</tr>
</tbody>
</table>
This is another print statement

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```
It prints the string “The end.”

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
    print "Hello," , name
print "The end."
```

```
Hello, Andrew
Hello, Tsanwani
Hello, Arno
Hello, Tebogo
The end.
```
Python looks for the next statement...

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:  
  print "Hello," , name  
print "The end."
```
... but there isn’t any, so Python stops.

```python
for name in ["Andrew", "Tsanwani", "Arno", "Tebogo"]:
    print "Hello,", name
print "The end."
```
Hello, Andrew
Hello, Tsanwani
Hello, Arno
Hello, Tebogo
The end.