Mutability Experiments

Append and AppendTo with Variables

Append creates a copy:

In[121]:=
```
l = {};
Append[l, 1]
```
Out[122]=
```
{1}
```

In[123]:=
```
l
```
Out[123]=
```
{}
```

AppendTo mutates:

In[124]:=
```
l = {};
AppendTo[l, 1]
```
Out[125]=
```
{1}
```

In[126]:=
```
l
```
Out[126]=
```
{1}
```

Append and AppendTo with Symbols that Are Not Variables

Append still successfully creates a copy:

In[135]:=
```
Append[{}, 1]
```
Out[135]=
```
{1}
```

However, AppendTo cannot mutate a symbol that is not a variable:

In[136]:=
```
AppendTo[{}, 1]
```
AppendTo: {} is not a variable with a value, so its value cannot be changed.

Out[136]=
```
AppendTo[{}, 1]
```

Using NestList with Append and AppendTo

NestList works with Append:

In[127]:=
```
appendOne[l_] := Append[l, 1]
```

In[128]:=
```
initial = {};
results = NestList[appendOne, initial, 5]
```

Out[129]=
```
{{}, {1}, {1, 1}, {1, 1, 1}, {1, 1, 1, 1}, {1, 1, 1, 1, 1}}
```

In[130]:=
```
initial
```

Out[130]=
```
{}
```

NestList does not work with AppendTo:

In[131]:=
```
appendOneUsingAppendTo[l_] := AppendTo[l, 1]
```

In[132]:=
```
initial = {};
results = NestList[appendOneUsingAppendTo, initial, 5]
```

AppendTo: {} is not a variable with a value, so its value cannot be changed.

AppendTo: AppendTo[{}, 1] is not a variable with a value, so its value cannot be changed.

AppendTo: AppendTo[AppendTo[{}, 1], 1] is not a variable with a value, so its value cannot be changed.

General: Further output of AppendTo::rvalue will be suppressed during this calculation.

Out[133]=
```
{{}, AppendTo[{}, 1], AppendTo[AppendTo[{}, 1], 1],
 AppendTo[AppendTo[AppendTo[{}, 1], 1], 1],
 AppendTo[AppendTo[AppendTo[AppendTo[{}, 1], 1], 1], 1],
 AppendTo[AppendTo[AppendTo[AppendTo[AppendTo[{}, 1], 1], 1], 1], 1]}
```

Next Things to Look Into

Not only does AppendTo not work with NestList, but there is a warning in the AppendTo documentation that it can be slow. There are two functions called Reap and Sow that are suggested instead.