Example: Computing factorial of a number $n$.

$n! = n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdots \cdot 1$

$n$ must be an int $\geq 1$.

Assume $0! = 1$

\[1! = 1 \cdot 1 = 1\]
\[2! = 2 \cdot 1 = 2 \cdot 1\]
\[3! = 3 \cdot 2 \cdot 1 = 3 \cdot (2 \cdot 1)\]
\[4! = 4 \cdot 3 \cdot 2 \cdot 1 = 4 \cdot (3 \cdot 2 \cdot 1)\]

factorial($4$) = $4 \cdot$ factorial($3$)

factorial($3$) = $3 \cdot$ factorial($2$)

factorial($2$) = $2 \cdot$ factorial($1$)

factorial($1$) = $1 \cdot$ factorial($0$)

factorial($0$) is defined as $1$.

$0! = \text{factorial}(0)$

= base case where problem boils down to one known number.

$[17]$
```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

After we return a value, remember that symbol table goes away.

Base case where recursion stops.

**Factorial symbol table**

```
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>3</td>
</tr>
<tr>
<td>n</td>
<td>2</td>
</tr>
<tr>
<td>n</td>
<td>1</td>
</tr>
<tr>
<td>n</td>
<td>0</td>
</tr>
</tbody>
</table>
```

Upon return where symbol table goes away.
def factorial(n):
    if n == 0:
        ans = 1
    else:
        ans = n * factorial(n-1)
    return ans