Python Subset Understood by ByePy

- looping and iteration
  
  ```python
  while, for v in range(array)
  ```

- flow control statements
  
  ```python
  continue, break, return
  ```

- conditional statements (if, elif, else) as well as expressions (v1, v2, else):
  
  ```python
  v1, if v2, else:
  ```

- variable assignment (v=c, c+=v, c[v1]=c[v2]) and references:
  
  ```python
  list([v1,...,vn]), indexed access and slicing
  ```

- a large range of built-in operators and functions
  
  ```python
  u, v, *k, <, <=, ==, and, or, len, max, ceil, sqrt
  ```

- embedded read-only queries
  
  ```sql
  SELECT $v as "resultn"
  ```

Interplay of the Python Interpreter and SQL Engine

![Diagram illustrating the interplay between Python and SQL execution](image)

Snakes on a Plan — ByePy

Bye, Python! — How we compile UDFs with complex control flow into one recursive SQL CTE.

A Collection of Compiled Python UDFs

<table>
<thead>
<tr>
<th>UDF Description</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force n-body simulation (binary-nut quad tree)</td>
<td>5.9x</td>
</tr>
<tr>
<td>March track border of 2D object (Marching Squares)</td>
<td>10.0x</td>
</tr>
<tr>
<td>margin buy/sell TPC-H order to maximize margin</td>
<td>3.4x</td>
</tr>
<tr>
<td>Markov-chain based robot control</td>
<td>3.6x</td>
</tr>
<tr>
<td>packing pack TPC-H inserts tightly into containers</td>
<td>16.0x</td>
</tr>
<tr>
<td>savings optimize supply chain of a TPC-H order</td>
<td>31.0x</td>
</tr>
<tr>
<td>vs. execute program on a simple virtual machine</td>
<td>24.0x</td>
</tr>
</tbody>
</table>

Compiling UDF F:
Intermediate Program Forms

1. Bring F into SSA form in which iterative as well as conditional control flow is exclusively expressed in terms of goto
2. translate the resulting graph of SSA blocks into a bundle of tail-recursive functions in ANF
3. form a central trampoline function which dispatches to the functions in the bundle, then loops back to itself, and
4. inline the functions into the trampoline, after which the recursive CTE $0$, can be read off this final intermediate form.

![Diagram illustrating the compiler stages](image)