

```

<prog> -> <more_EOL> <prog_body_with_def> EOF

<prog_body_with_def> -> epsilon
<prog_body_with_def> -> def id ( <param_list> ) : EOL indent <nonempty_prog_body>
                                dedent <more_EOL> <prog_body_with_def>
<prog_body_with_def> -> <command> <prog_body_with_def>

<nonempty_prog_body> -> <more_EOL> <command> <prog_body>

<prog_body> -> <command> <prog_body>
<prog_body> -> epsilon

<command> -> pass EOL <more_EOL>
<command> -> return <return_item> EOL <more_EOL>
<command> -> while expr : EOL indent <nonempty_prog_body> dedent <more_EOL>
<command> -> if expr : EOL indent <nonempty_prog_body> dedent else : EOL
                  indent <nonempty_prog_body> dedent <more_EOL>
<command> -> int expr EOL <more_EOL>
<command> -> float expr EOL <more_EOL>
<command> -> str expr EOL <more_EOL>
<command> -> ( expr EOL <more_EOL>
<command> -> id <not_sure1> EOL <more_EOL>

<return_item> -> expr
<return_item> -> epsilon
<return_item> -> None

<param_list> -> <param_item> <more_params>
<param_list> -> epsilon

<more_params> -> , <param_item> <more_params>
<more_params> -> epsilon

<param_item> -> int
<param_item> -> float
<param_item> -> str
<param_item> -> id
<param_item> -> None

<more_EOL> -> EOL <more_EOL>
<more_EOL> -> epsilon

<not_sure1> -> ( <param_list> )
<not_sure1> -> = <not_sure2>
<not_sure1> -> <op> expr
<not_sure1> -> epsilon

<not_sure2> -> id <not_sure3>
<not_sure2> -> int expr
<not_sure2> -> float expr
<not_sure2> -> str expr
<not_sure2> -> ( expr

<not_sure3> -> ( <param_list> )
<not_sure3> -> <op> expr
<not_sure3> -> epsilon

<op> -> +      <op> -> -
<op> -> *      <op> -> /
<op> -> //     <op> -> <
<op> -> >      <op> -> <=
<op> -> >=    <op> -> !=
<op> -> ==
```