

# SICPy §2

Khooi Xin Zhe, Ang Jun Jie

19 April, 2021

A SICPy<sup>1</sup> program is a *module*, defined using Backus-Naur Form<sup>2</sup> as follows:

## SICPy §2

<i>module</i>	::=	<i>statement</i> ...	statement sequence
<i>statement</i>	::=	<i>name</i> = <i>expression</i>   <i>function</i>   <u>return</u> <i>expression</i>   <i>if-statement</i>   <i>expression</i>	assignment statement function declaration return statement conditional statement expression statement
<i>function</i>	::=	<u>def</u> <i>name</i> ( <i>parameters</i> ) : <i>statement</i> ...	function declaration
<i>parameters</i>	::=	ε   <i>name</i> [ , <i>name</i> ] ...	function parameters
<i>if-statement</i>	::=	<u>if</u> <i>expression</i> : <i>statement</i> ... [[ <u>elif</u> <i>expression</i> : <i>statement</i> ... ] ... <u>else</u> : <i>statement</i> ... ]	conditional statement
<i>expression</i>	::=	<i>number</i>   <u>True</u>   <u>False</u>   <u>None</u>   <i>string</i>   <i>name</i>   <i>expression</i> <i>binary-operator</i> <i>expression</i>   <i>unary-operator</i> <i>expression</i>   <i>expression</i> ( <i>expressions</i> )   <u>lambda</u> <i>name</i> [ , <i>name</i> ] ... : <i>expression</i>   <i>expression</i> <u>if</u> <i>expression</i> <u>else</u> <i>expression</i>   ( <i>expression</i> )	primitive number expression primitive boolean expression primitive list expression primitive string expression name expression binary operator combination unary operator combination function application lambda expression conditional expression parenthesised expression
<i>binary-operator</i>	::=	+   -   *   /   %   ==   >   <   >=   <=   and   or	
<i>unary-operator</i>	::=	not   +   -	
<i>expressions</i>	::=	ε   <i>expression</i> [ , <i>expression</i> ] ...	argument expressions

<sup>1</sup>SICPy is an adaptation of Source - the official language of the textbook Structure and Interpretation of Computer Programs, JavaScript Adaptation.

<sup>2</sup>We adopt Henry Ledgard's BNF variant that he described in A human engineered variant of BNF, ACM SIGPLAN Notices, Volume 15 Issue 10, October 1980, Pages 57-62. In our grammars, we bold and underline keywords, [ ] for optional syntaxes, italics for syntactic variables, ε for nothing, *x|y* for *x* or *y*, and *x...* for zero or more repetitions of *x*.