**Chapter 2 notes.**

**Variables, expressions, and statements**

* A *value* is one of the basic things a program works with, like a letter or a number.
* One of the most powerful features of a programming language is the ability to manipulate *variables*. A variable is a name that refers to a value.
* The interpreter uses keywords to recognize the structure of the program, and they cannot be used as variable names.

A screenshot of a cell phone

Description automatically generated

* *statement* is a unit of code that the Python interpreter can execute. We have seen two kinds of statements: print being an expression statement and assignment.
* *Operators* are special symbols that represent computations like addition and multiplication. The values the operator is applied to are called *operands*. The operators +, -, \*, /, and \*\* perform addition, subtraction, multiplication, division, and exponentiation.
* An *expression* is a combination of values, variables, and operators. A value all by itself is considered an expression.
* When more than one operator appears in an expression, the order of evaluation depends on the *rules of precedence*. For mathematical operators, Python follows mathematical convention. The acronym *PEMDAS* is a useful way to remember the rules.
* The *modulus operator* works on integers and yields the remainder when the first operand is divided by the second. In Python, the modulus operator is a percent sign (%).
* The + operator works with strings, but it is not addition in the mathematical sense. Instead it performs *concatenation*, which means joining the strings by linking them end to end.
* Python provides a built-in function called input that gets input from the keyboard1. When this function is called, the program stops and waits for the user to type something. When the user presses Return or Enter, the program resumes and input returns what the user typed as a string.
* The sequence \n at the end of the prompt represents a *newline*, which is a special character that causes a line break. That’s why the user’s input appears below the prompt. If you expect the user to type an integer, you can try to convert the return value to int using the int() function.

**2.14 Glossary**

**assignment** A statement that assigns a value to a variable.

**concatenate** To join two operands end to end.

**comment** Information in a program that is meant for other programmers (or

anyone reading the source code) and has no effect on the execution of the

program.

**evaluate** To simplify an expression by performing the operations in order to yield

a single value.

**expression** A combination of variables, operators, and values that represents a

single result value.

**floating point** A type that represents numbers with fractional parts.

**integer** A type that represents whole numbers.

**keyword** A reserved word that is used by the compiler to parse a program; you

cannot use keywords like if, def, and while as variable names.

**mnemonic** A memory aid. We often give variables mnemonic names to help us

remember what is stored in the variable.

**modulus operator** An operator, denoted with a percent sign (%), that works on

integers and yields the remainder when one number is divided by another.

**operand** One of the values on which an operator operates.

**operator** A special symbol that represents a simple computation like addition,

multiplication, or string concatenation.

**rules of precedence** The set of rules governing the order in which expressions

involving multiple operators and operands are evaluated.

**statement** A section of code that represents a command or action. So far, the

statements we have seen are assignments and print expression statement.

**string** A type that represents sequences of characters.

**type** A category of values. The types we have seen so far are integers (type int),

floating-point numbers (type float), and strings (type str).

**value** One of the basic units of data, like a number or string, that a program

manipulates.

**variable** A name that refers to a value.