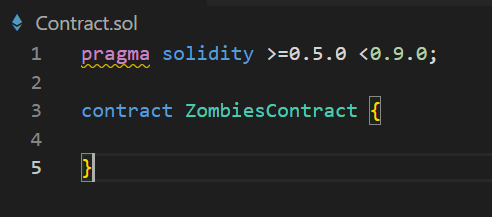
# Chapter 2: Contracts

* Solidity's code is encapsulated in contracts. A contract is the fundamental building block of Ethereum applications — all variables and functions belong to a contract.

**Version Pragma**

* All solidity source code should start with a "version pragma" — a declaration of the version of the Solidity compiler this code should use.

Example:  


**Chapter 3: State Variables & Integers**

* State variables are permanently stored in contract storage. This means they're written to the Ethereum blockchain. Think of them like writing to a DB.

**Unsigned Integers: uint**

* The uint data type is an unsigned integer, meaning its value must be non-negative. There's also an int data type for signed integers.

# Chapter 4: Math Operations

* The following operations are the same as in most programming languages:
  + Addition: x + y
  + Subtraction: x – y
  + Multiplication: x \* y
  + Division: x / y
  + Modulus / remainder: x % y (for example, 13 % 5 is 3, because if you divide 5 into 13, 3 is the remainder)
  + Solidity also supports an exponential operator (i.e. "x to the power of y", x^y) e.g. uint x = 5 \*\* 2; // equal to 5^2 = 25

# Chapter 5: Structs

* Structs allow you to create more complicated data types that have multiple properties.
* Strings are used for arbitrary-length UTF-8 data. Ex. string greeting = "Hello world!"