Outline of a simple scanner

- The parser asks the scanner for the next token
- The scanner reads the next character x from the character stream
- If x is a special character like ; , + - * ( ) the scanner returns the corresponding token
  - If x is =, peek at the next character y
    - If y is not =, return the token for = (ASSIGN)
    - If y is =, read the characer and return the token for == (EQUAL)

Outline of a simple scanner

- Some tokens have additional information attached
- When parsing we will want the type and the specific value (i.e. identifier token and identifier string, or constant token and constant value)
- If x is a letter, keep reading characters and stop before the first non-letter or non-digit character (assuming our language requires identifiers to start with a letter and then have only letters or digits)
  - If the string is a keyword, return the token for that keyword
  - If the string is a not a keyword, return the id token with the string
- If x is a digit, keep reading characters and stop before the first non-digit character
  - Return the token const with the numeric value