1. Introduction
   1. Hello everyone, we are group 13, and I’m Jisu and I’m Donggyu. We are going to present about multiple syntax styles.
2. Overview
   1. Currently, Amy syntax supports brace-style in if and match statements, and end marker-style in object keyword. Our goal is to allow both styles in these three keywords. In this example, you can see that no braces are here for if and match keywords, and used braces to object keyword which is opposite to the original implementation.
3. Background
   1. There are several methods to identify the scope of the statements. Languages like Python or Haskell use indentation for it, while languages like C, Java, JavaScript use brackets. Some languages like PHP support the end markers to substitute the braces. Codes using indentation are neat and easy to write, but when the code is nested deeply, it is hard to recognize the exact indent needed. Codes using curly brackets are better to specify the statement blocks, but it tends to have unnecessary lines. End markers are used to reduce these unnecessary lines that braces have.
4. Affected Compiler Phases
   1. Our extension only affects Lexer and Parser of overall phases. We have to change Lexer to add the keywords of *end if* and *end match,* and change parser to accept the both brace style and end marker styles to produce same Abstract Syntax Tree. Other phases are not affected.
5. Affected Compiler Phases - Lexer
   1. We changed Lexer as shown. Added new words like ‘end if’ or ‘end match’ as KeywordToken.
6. Affected Compiler Phases - Parser
   1. Something
7. Conclusion -1
   1. Using this extension, users can use both braces and end markers in their taste for if, match, and object keywords. This can help the newcomers of Amy language to learn the language more quickly by using their own familiar styles. As you can see, this is an example of C# using end markers as preprocessor in if statement, and This is typical Java code using braces for if statement. A user can choose the style which is familiar with.
8. Conclusion -2
   1. And this extension may help to increase the readability of code. In this nested if and match statements in brace style, it is hard to recognize which scope the below else statement is in, but by using end marker style, it is quite better for finding the context.
9. Future works and expansions
   1. In future works, we can adopt indentation which does not need any braces or end markers which is used in this python example, or we may add compiler options to force to one format for the unity of the code just like scalac compiler provides.
10. End
    1. Thank you all for listening to our presentation, and do you have any questions?