Lexical Rules, Tables and Sample Problem (identifier, literal, delimiter)

This reading contains the lexical rules and tokens for lesson 2.03. It also contains a sample quiz question that applies lexical analysis to a statement using these lexical rules and tables.

Current lexical rules:

identifier

starts with a letter or underscore, and is followed by zero or more letters, underscores, or digits

literal string

starts with a quote, followed by zero or more non-quote characters, and ends with a quote

literal integer

one or more digits

literal float

one dot, one or more digits and no other characters

longest token rule

when creating a token, create the longest token possible

whitespace rule

for whitespace not at the start of a line:

- if whitespace is inside a literal string it is part of the literal string
- otherwise, it ends the current token and no token is created for it

Current lexical tables:

delimiter

(	)	[	]	{
}	,	:	.	;
@	=	->	+=	-=
*=	/=	//=	%=	&=
|=	^=	>>=	<<=	**=

Sample lexical analysis quiz question:

You will be asked to tokenize a character sequence. For each token in a character sequence, type the token and its token kind in one answer field separated by a single space. Enter the tokens in the order that they would be created by the Python interpreter.

If you encounter a character that does not match the lexical rules or tables for any of the listed token kinds, type the single character that violates the rule followed by a space and then type UNKNOWN as its token kind. Start tokenizing again after this single character.

For example, given this character sequence:

```
len('hello')?3
```

What is the 1st token and token kind?

len identifier

What is the 2nd token and token kind?

(delimiter

What is the 3rd and token kind?

'hello' literal string

What is the 4th token and token kind?

) delimiter

What is the 5th token and token kind?

? UNKNOWN

What is the 6th token and token kind?

3 literal integer