8. ADVANCED FEATURES

Text Boundaries

Introduction

Text Boundaries refer to the beginning and end of strings, or the beginning and end of words within a string. Sometimes if we are looking for a pattern in a String, we might only want to match it if it comes at the start or at the end of a String or word.

Word Boundaries

Code	Description
\b	Matches to a pattern if it's at the start or at the end of a single word.
\B	Matches to a pattern if it's in the middle of a single word.

- If the word we are considering is "`Hello`", then "**\b**`He`" will match, as well as "`llo`**\b**", but "**\b**`el`" will not match, and neither will "`ll`**\b**".
- And with the "**\B**", it's the opposite, if the word is "`Hello`", then "**\B**`el`" will match, and so will "`ll`**\B**", but "**\B**`He`" and "`llo`**\B**" will not match.

String Boundaries

Code	Description
^	Matches the beginning of an input String.
$	Matches the end of an input String.

- If, for example, the String is "`Hello, World!`", then the expression "**^**`Hell`" will match, but "**^**`ello`" will not match.
- And with the "**$**", the expression "`rld!`**$**" will match, but the expression "`world`**$**" will not match.
- We can use both together, "**^**`X{3}`**$**" matches the standalone String "`XXX`".

Why do we need Text Boundaries?

If we don't use Text Boundaries, and the Regular Expression pattern matches the middle, or end of the String it is being compared to, the output will differ depending on which programming language being used, e.g.:

Position of Text	RegEx Code	*C*	*Python*	*Java*
Text from the middle of the String	`RegEx_Pattern = "llo"` `Test_Message = "Hello, World!"`	✓	✗	✓
Text from the end of the String	`RegEx_Pattern = "World!"` `Test_Message = "Hello, World!"`	✓	✗	✓

So in ***Python***, if the pattern is from the middle or end of the string, it would indicate that the two Strings are not an exact match (which is correct), whereas in either ***Java*** or ***C***, the outcome would be that the two Strings are an exact match (they really aren't an ***exact*** match), so they will ignore the extra characters.