## Character Class Subtraction

Character class subtraction is supported by the XML Schema, XPath, .NET (version 2.0 and later), and JGsoft regex flavors. It makes it easy to match any single character present in one list (the character class), but not present in another list (the subtracted class). The syntax for this is `[class-[subtract]]`. If the character after a hyphen is an opening bracket, these flavors interpret the hyphen as the subtraction operator rather than the range operator. You can use the full character class syntax within the subtracted character class.

The character class `[a-z-[aeiuo]]` matches a single letter that is not a vowel. In other words: it matches a single consonant. Without character class subtraction or intersection, the only way to do this would be to list all consonants: `[b-df-hj-np-tv-z]`.

The character class `[\p{Nd}-[^\p{IsThai}]]` matches any single Thai digit. The base class matches any Unicode digit. All non-Thai characters are subtracted from that class. `[\p{Nd}-[\P{IsThai}]]` does the same. `[\p{IsThai}-[^\p{Nd}]]` and `[\p{IsThai}-[\P{Nd}]]` also match a single Thai digit by subtracting all non-digits from the Thai characters.

## Nested Character Class Subtraction

Since you can use the full character class syntax within the subtracted character class, you can subtract a class from the class being subtracted. `[0-9-[0-6-[0-3]]]` first subtracts `0-3` from `0-6`, yielding `[0-9-[4-6]]`, or `[0-37-9]`, which matches any character in the string `0123789`.

The class subtraction must always be the last element in the character class. `[0-9-[4-6]a-f]` is not a valid regular expression. It should be rewritten as `[0-9a-f-[4-6]]`. The subtraction works on the whole class. E.g. `[\p{Ll}\p{Lu}-[\p{IsBasicLatin}]]` matches all uppercase and lowercase Unicode letters, except any ASCII letters. The `\p{IsBasicLatin}` is subtracted from the combination of `\p{Ll}\p{Lu}` rather than from `\p{Lu}` alone. This regex will not match `abc`.

While you can use nested character class subtraction, you cannot subtract two classes sequentially. To subtract ASCII characters and Greek characters from a class with all Unicode letters, combine the ASCII and Greek characters into one class, and subtract that, as in `[\p{L}-[\p{IsBasicLatin}\p{IsGreek}]]`.

## Negation Takes Precedence over Subtraction

The character class `[^1234-[3456]]` is both negated and subtracted from. In all flavors that support character class subtraction, the base class is negated before it is subtracted from. This class should be read as “(not 1234) minus 3456”. Thus this character class matches any character other than the digits 1, 2, 3, 4, 5, and 6.

## Notational Compatibility with Other Regex Flavors

Note that a regex like `[a-z-[aeiuo]]` does not cause any errors in most regex flavors that do not support character class subtraction. But it won’t match what you intended either. In most flavors, this regex consists of a character class followed by a literal `]`. The character class matches a character that is either in the range a-z, or a hyphen, or an opening bracket, or a vowel. Since the a-z range and the vowels are redundant, you could write this character class as `[a-z-[]` or `[-[a-z]` in Perl. A hyphen after a range is treated as a literal character, just like a hyphen immediately after the opening bracket. This is true in the XML, .NET and JGsoft flavors too. `[a-z-_]` matches a lowercase letter, a hyphen or an underscore in these flavors.

Strictly speaking, this means that the character class subtraction syntax is incompatible with Perl and the majority of other regex flavors. But in practice there's no difference. Using non-alphanumeric characters in character class ranges is very bad practice because it relies on the order of characters in the ASCII character table. That makes the regular expression hard to understand for the programmer who inherits your work. While `[A-[]` would match any uppercase letter or an opening square bracket in Perl, this regex is much clearer when written as `[A-Z[]`. The former regex would cause an error with the XML, .NET and JGsoft flavors, because they interpret `-[]` as an empty subtracted class, leaving an unbalanced `[`.