## Match Numbers

Write a program which finds all **integer** and **floating-point numbers** in a string.

### Compose the Regular Expression

A number has the following characteristics:

* Has either **whitespace** before it or the **start** of the string (match either **^** or what's called a [positive lookbehind](http://www.regular-expressions.info/lookaround.html)). The entire syntax for the **beginning** of your **RegEx** might look something like "(^|(?<=\s))".
* The number might or might not be negative, so it might have a hyphen on its left side ("-").
* Consists of **one or more digits**.
* Might or might not have **digits after the** **decimal point**
* The decimal part (if it exists) consists of a period (".") and **one or more digits** after it. Use a **capturing group**.
* Has either **whitespace** before it or the **end** of the string (match either **$** or what's called a [positive lookahead](http://www.regular-expressions.info/lookaround.html)). The syntax for the **end** of the **RegEx** might look something like "($|(?=\s))".

Let's see how we would translate the above rules into a **regular expression**:

* First off, we need to establish what needs to exist **before** our number. We can't use \b here, since it includes "-", which we need to match **negative numbers**.   
  Instead, we'll use a **positive lookbehind**, which **matches** if there's something **immediately behind** it. We'll match if we're either at the **start** of the string (^), or if there's any **whitespace** **behind** the string:  
  
* Next, we'll check whether there's a **hyphen**, signifying a **negative number**:  
  Since having a negative sign **isn't required**, we'll use the "?" quantifier, which means "**between 0 and 1 times**".
* After that, we'll match any integers – naturally, consisting **one or more digits**:  
  
* Next, we'll match the **decimal** part of the number, which **might or might not exist** (note: we need to escape the **period** character, as it's used for something else in RegEx):  
  
* Finally, we're going to use the same logic for the end of our string as the start – we're going to match **only** if the number has **either a whitespace or the end of the string ("**$**")**:  
  

You can follow the table below to help with composing your RegEx:

|  |  |
| --- | --- |
| **Match ALL of these** | **Match NONE of these** |
| 1 -1 123 -123 123.456 -123.456 | 1s s2 s-s -1- \_55\_ s-2 s-3.5 s-1.1 |