

65. Valid Number

Description

A **valid number** can be split up into these components (in order):

- 1. A **decimal number** or an **integer**.
- 2. (Optional) An `'e'` or `'E'`, followed by an **integer**.

A **decimal number** can be split up into these components (in order):

- 1. (Optional) A sign character (either `'+'` or `'-'`).
- 2. One of the following formats:
 - 1. One or more digits, followed by a dot `'.'`.
 - 2. One or more digits, followed by a dot `'.'`, followed by one or more digits.
 - 3. A dot `'.'`, followed by one or more digits.

An **integer** can be split up into these components (in order):

- 1. (Optional) A sign character (either `'+'` or `'-'`).
- 2. One or more digits.

For example, all the following are valid numbers:

`["2", "0089", "-0.1", "+3.14", "4.", "-.9", "2e10", "-90E3", "3e+7", "+6e-1", "53.5e93", "-123.456e789"]`, while the following are not valid numbers:
`["abc", "1a", "1e", "e3", "99e2.5", "--6", "-+3", "95a54e53"]`.

Given a string `s`, return `true` *if* `s` *is a valid number*.

Example 1:

Input: `s = "0"`
Output: `true`

Example 2:

Input: `s = "e"`
Output: `false`

Example 3:

Input: `s = "."`
Output: `false`

Constraints:

- `1 <= s.length <= 20`
- `s` consists of only English letters (both uppercase and lowercase), digits (`0-9`), plus `'+'`, minus `'-'`, or dot `'.'`.

