Add a way for class String to determine whether it has only numbers / digits or not

Hello.

I was wondering whether we could have a method that does something similar, that is also easier to read.

Something like:

```ruby
x = '123'
x.only_numbers?
```

or

```ruby
x.only_digits?
```

I do not know whether the word “only” is good or not. Perhaps “all_numbers?” or “all_digits?” would be better. But the name is secondary probably, more important would be whether such a method would make sense for class String.

Anyway, thanks for reading!

---

#1 - 07/18/2016 05:38 AM - sawa (Tsuyoshi Sawada)

I am interpreting that this proposal is actually interested in integers with signatures (+/-) as well, not just digits.

If we have variants of methods like Kernel#Integer, Kernel#Float or usage with options so that they do not raise an error but return nil in case the string is not appropriate, then that should suffice. For example, if we allow these methods to take an optional second argument, that would suffice.

Method signature:

```ruby
Integer(string, raise_error = true)
```

Usage:

```ruby
Integer("123", false) # => 123
Integer("foo", false) # => nil
```
Robert's use case would be:

```ruby
if Integer(x, false)
```

#2 - 07/20/2016 02:31 AM - shyouhei (Shyouhei Urabe)

We looked at this issue at yesterday's developer meeting.

- For the proposed functionality: why do you want to detect numbers? Because it would make more sense to convert it to integer, not just detection.
- For Integer() extension: there is no constructor method that takes arguments like this.

#3 - 07/20/2016 03:13 AM - nobu (Nobuyoshi Nakada)

I feel that to_i extension might be better than Integer.

```
"foo".to_i #=> 0
"foo".to_i(fallback: nil) #=> nil
```