

# In this lecture, we will discuss...

- ✧ Different kinds of strings supported by Ruby
- ✧ Many methods supported by the String API
- ✧ Symbols



# Strings

- ✧ Single-quote literal strings are **very** literal
  - Allow escaping of `\` with `\`
  - Show (almost) everything else **as is**
- ✧ Double-quoted strings
  - Interpret special characters like `\n` and `\t`
  - Allow string interpolation!

**Don't bother concatenating with +**



# Strings / Interpolation

```
single_quoted = 'ice cream \n followed by it's a party!'
double_quoted = "ice cream \n followed by it's a party!"

puts single_quoted # => ice cream \n followed by it's a party!
puts double_quoted # => ice cream
                    # => followed by it's a party!

def multiply (one, two)
  "#{one} multiplied by #{two} equals #{one * two}"
end
puts multiply(5, 3)
# => 5 multiplied by 3 equals 15
```

Interpolation (only available for double-quoted strings)



# More Strings

- ✧ String methods ending with **!** modify the existing string
  - Most others just return a new string
- ✧ Can also use `%Q{long multiline string}`
  - Same behavior as double-quoted string

Very Important to Master String API



# More Strings

```
my_name = " tim"
puts my_name.lstrip.capitalize # => Tim
p my_name # => " tim"
my_name.lstrip! # (destructive) removes the leading space
my_name[0] = 'K' # replace the first character
puts my_name # => Kim


cur_weather = %Q{It's a hot day outside
               Grab your umbrellas...}

cur_weather.lines do |line|
  line.sub! 'hot', 'rainy' # substitute 'hot' with 'rainy'
  puts "#{line.strip}"
end
# => It's a rainy day outside
# => Grab your umbrellas...
```



# Strings API



 **include? other\_str → true or false**  
Returns `true` if *str* contains the given string or character.

```
"hello".include? "lo"    #=> true
"hello".include? "ol"    #=> false
"hello".include? ?h      #=> true
```



# Symbols

- ✧ `:foo`- highly optimized strings
- ✧ **Constant names** that you don't have to pre-declare
  - **“Stands for something”** string type



# Symbols

- ✧ Guaranteed to be **unique** and **immutable**
- ✧ Can be converted to a `String` with `to_s`
  - **Or** from `String` to `Symbol` with `to_sym`





# Symbols

```
~$ irb
irb(main):001:0> "hello".methods.grep /case/
=> [:casecmp, :upcase, :downcase, :swapcase, :upcase!, :downcase!, :swapcase!]
```



# Summary

- ✧ Interpolation lets you **finish your thought**
- ✧ Strings have a lot of **really useful API**

## What's next?

- ✧ Arrays

