RAGNARSO...

Initialize multiple variables at once with conditional statement

This works:

```
@selected_date =
  if params[:date].present?
    Date.parse(params[:date])
  else
    Date.today.beginning_of_month
  end
```

This also works:

```
start_date, end_date = 1.month.ago, Date.today
```

This doesn't:

```
start_date, end_date =
  if @selected_date.month == Date.today.month
    1.month.ago, Date.today
  else
    @selected_date.beginning_of_month, @selected_date.end_of_month
  end
```

Is it possible to initialize multiple variables this way?

The block has to return the array:

```
start_date, end_date =
    if @selected_date.month == Date.today.month
      [1.month.ago, Date.today]
    else
      [@selected_date.beginning_of_month, @selected_date.end_of_month]
    end
```

I will create 2 different methods, like

```
def start_date
  @selected_date.month == Date.today.month ? 1.month.ago : @selected_date.beginning_of_month
end
```

those @selected_date.month == Date.today.month can be a method too, like

```
def current_month?
  @selected_date.month == Date.today.month
end
```

and then

```
def start_date
  current_month? ? 1.month.ago : @selected_date.beginning_of_month
end
```

and selected_date as attr_reader to reduce using @, will be nice to have whole code to check hard to discuss having only few lines

...

Cool and simple, thanks!