RUBY BITS

EXPRESSIONS

PRESS START

ICONS

⚠	Bad Code
☠	Worse Code
✓	Good Code
1UP	Awesome Code

UNLESS

```ruby
if ! tweets.empty?
  puts "Timeline:"
  puts tweets
end
```

⚠

Unless is more intuitive

```ruby
unless tweets.empty?
  puts "Timeline:"
  puts tweets
end
```

✓

UNLESS WITH ELSE

```
unless tweets.empty?
  puts "Timeline:"
  puts tweets
else
  puts "No tweets found - better follow some people!"
end
```

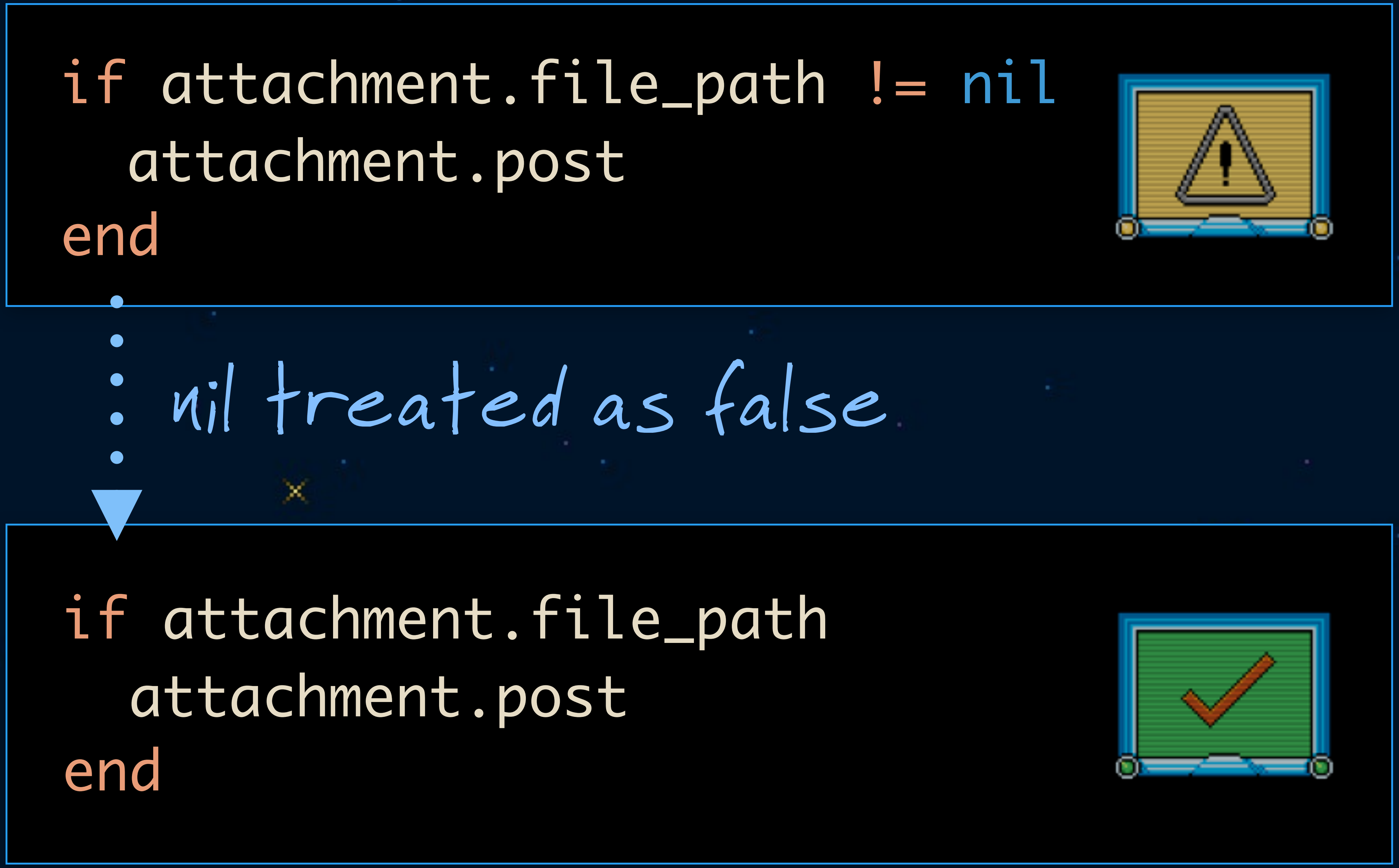
⚠

Unless with else is confusing

```
if tweets.empty?
  puts "No tweets found - better follow some people!"
else
  puts "Timeline:"
  puts tweets
end
```

✓

NIL IS FALSE-Y



```ruby
if attachment.file_path != nil
  attachment.post
end
```

⚠

nil treated as false

```ruby
if attachment.file_path
  attachment.post
end
```

✓

ONLY NIL IS FALSE-Y

`""` treated as "true"!

`0` treated as "true"!

`[]` treated as "true"!

```
unless name.length
  warn "User name required"
end
```

will never be false!

INLINE CONDITIONALS

```ruby
if password.length < 8
  fail "Password too short"
end
unless username
  fail "No user name set"
end
```

try inline if/unless

```ruby
fail "Password too short" if password.length < 8
fail "No user name set" unless username
```

SHORT-CIRCUIT – "AND"

```
if user
  if user.signed_in?
    #...
  end
end
```

⚠

use && instead

```
if user && user.signed_in?
 #...
end
```

✓

if user is nil
second half never runs

SHORT-CIRCUIT ASSIGNMENT

```
result = nil || 1
```

···▶ 1

```
result =   1 || nil
```

···▶ 1

```
result =   1 || 2
```

···▶ 1

DEFAULT VALUES - "OR"

```ruby
tweets = timeline.tweets
tweets = [] unless tweets
```

⚠

if nil, default to empty array

```ruby
tweets = timeline.tweets || []
```

✓

SHORT-CIRCUIT EVALUATION

```ruby
def sign_in
  current_session || sign_user_in
end
```

not evaluated unless current session is nil

but consider whether if/then would be more legible!

CONDITIONAL ASSIGNMENT

```ruby
i_was_set = 1
i_was_set ||= 2

puts i_was_set
```

···▶ 1

assigns IF there's no existing value

```ruby
i_was_not_set ||= 2

puts i_was_not_set
```

···▶ 2

CONDITIONAL ASSIGNMENT

```
options[:country] = 'us' if options[:country].nil?
options[:privacy] = true if options[:privacy].nil?
options[:geotag]  = true if options[:geotag].nil?
```

use conditional assignment

```
options[:country] ||= 'us'
options[:privacy] ||= true
options[:geotag]  ||= true
```

CONDITIONAL RETURN VALUES

```
if list_name
  options[:path] = "/#{user_name}/#{list_name}"
else
  options[:path] = "/#{user_name}"
end
```

⚠

assign the value of the if statement

```
options[:path] = if list_name
  "/#{user_name}/#{list_name}"
else
  "/#{user_name}"
end
```

✓

CONDITIONAL RETURN VALUES

```ruby
def list_url(user_name, list_name)
  if list_name
    url = "https://twitter.com/#{user_name}/#{list_name}"
  else
    url = "https://twitter.com/#{user_name}"
  end
  url
end
```

CONDITIONAL RETURN VALUES

```ruby
def list_url(user_name, list_name)
  if list_name
    "https://twitter.com/#{user_name}/#{list_name}"
  else
    "https://twitter.com/#{user_name}"
  end
end
```

CASE STATEMENT VALUE

```ruby
client_url = case client_name
  when "web"
    "http://twitter.com"
  when "Facebook"
    "http://www.facebook.com/twitter"
  else
    nil
end
```

CASE - RANGES

```ruby
popularity = case tweet.retweet_count
  when 0..9
    nil
  when 10..99
    "trending"
  else
    "hot"
end
```

CASE - REGEXPS

```
tweet_type = case tweet.status
  when /\A@\w+/
    :mention
  when /\Ad\s+\w+/
    :direct_message
  else
    :public
end
```

CASE - WHEN/THEN

```
tweet_type = case tweet.status
  when /\A@\w+/     then :mention
  when /\Ad\s+\w+/  then :direct_message
  else                   :public
end
```

RUBY
BITS
EXPRESSIONS
PRESS START