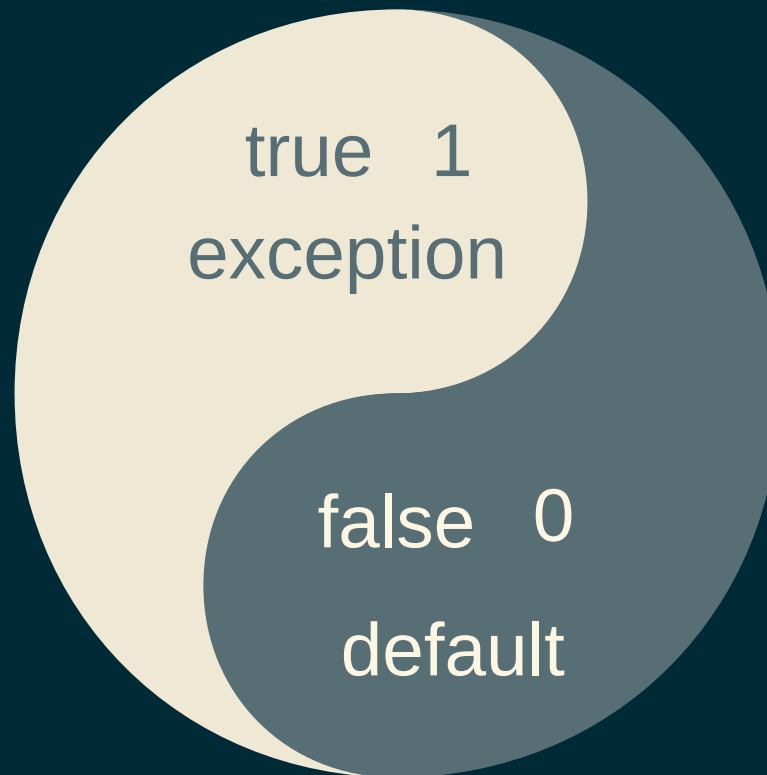


Always default to false

Paul Grayson
Las Vegas Ruby User Group
2014-01-29

Be opinionated!

- There is a right way
- Consistency aids memory



The default is already false or null in many contexts

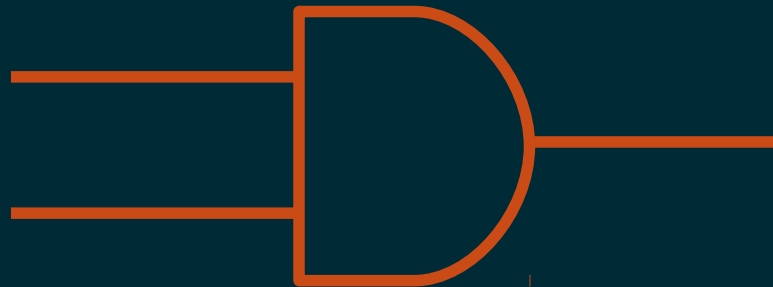
Ruby: @var, if x, methods, [], etc. -> nil

SQL: NULL

Javascript: undefined

C#/Java: bool/boolean defaults false

Originates in digital circuits?
- default is OFF



Options should be *opt in*


☒ YES, please add me to your mailing list

Fields, variables, conditionals

DO THIS:

```
if on_mailing_list  
  send_email  
end
```


*Body does not run
by default*



NOT THIS:


```
if !on_mailing_list  
  withhold_email  
end
```

*How would you even
write that method?*



Be careful with true default

```
<% if !out_of_stock %>  
  <div id="order_form">  
    ...  
  </div>  
<% end %>
```



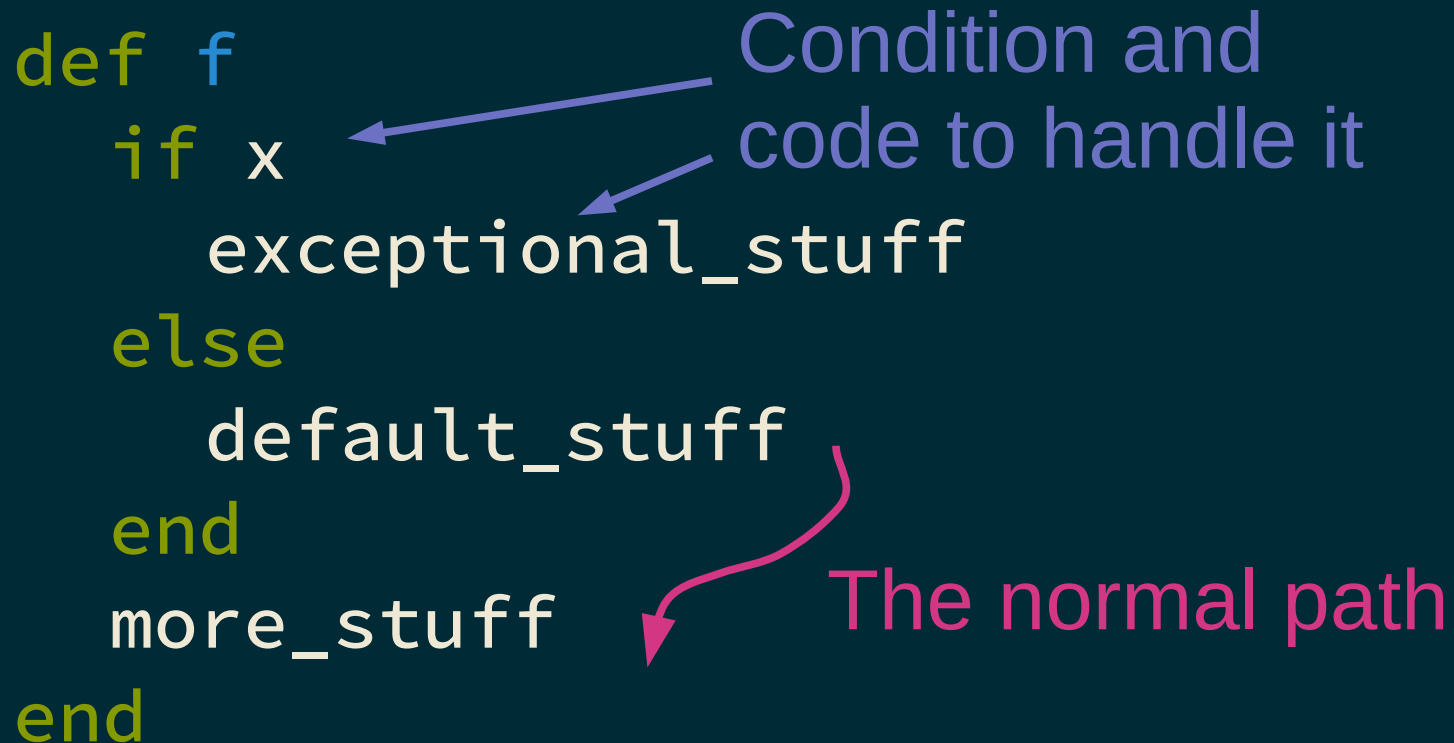
Must remember to not put
anything essential in here.

Emphasize connections

```
def f
  if x
    exceptional_stuff
  else
    default_stuff
  end
  more_stuff
end
```

Condition and code to handle it

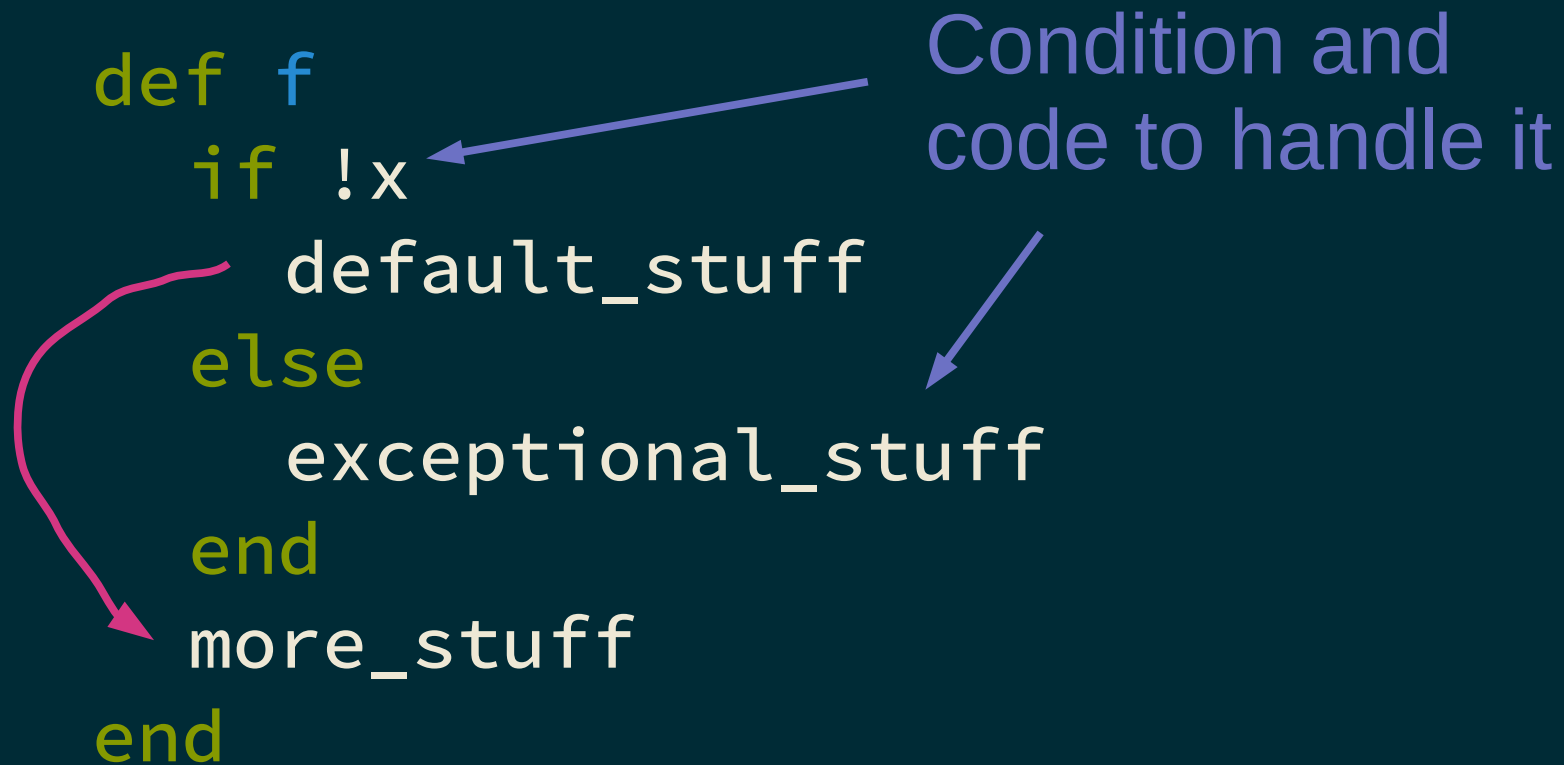
The normal path



true default complicates connections

```
def f
  if !x
    default_stuff
  else
    exceptional_stuff
  end
  more_stuff
end
```

Condition and code to handle it




The normal path

The “guard clause” pattern

```
def f
  return y if x
  normal_stuff
  ...
end
```

Condition and
code to handle it



The normal path



Summary

- Default *false* for all fields
- Test for unusual conditions (when possible)
- Beware of conditionals with *true* default

