Software Explained Simply

Episode 3 - Conditional Logic and Control Flow

Outline

- Comparison operators
- Boolean operators

- Presence or “Truthyness” (Demo)

Conditional (Boolean) Logic

- Choosing between code paths based on the scenario
    - “What does a logged-in user see vs a guest?”
    - “Does this user have a trial account?”
    - “Is this user an admin?”

Comparisons

```
x = 3    (assignment)
```

Comparisons

```
x = 3     (assignment)

x == 3    (equality check)
```

Comparisons

```
x = 3    (assignment)

x == 3   (equality check)

=> true
```

Comparisons

```
x = 3     (assignment)

x == 3    (equality check)
=> true

x == 4
=> false
```

Comparisons

```
x = 3

x == "3"
```

Comparisons

```
x = 3

x == "3"
```

```
=> false
```

Comparisons

```
x = 3

x > 10
```

Comparisons

```
x = 3

x > 10
```

```
=> false
```

Comparisons

```
x = 3

x > 10
```

```
=> false

x < 5
```

Comparisons

```
x = 3

x > 10

=> false

x < 5

=> true
```

Comparisons

$x = 3$

$x > 3$

Comparisons

x = 3

x > 3

```
=> false
```

Comparisons

x = 3

x > 3

```
=> false
```

```
x >= 3
```

```
=> true
```

Control flow

Control Flow

```
if <some-condition-here>
  <code-to-run-if-true>
end
```

Control Flow

```
x = 3

if x > 1
  x = 10

end
```

Control Flow

```
x = 3

if x > 1
  x = 10
end

x

=> 10
```

Control Flow

```
x = 3

if x > 100
  x = 10
end

x

=> 3
```

Control Flow

```
age = 50

if age >= 18
  # this person can vote

end
```

Control Flow

```
age = 50

if age >= 18
  # this person can vote
else
  # this person cannot vote
end
```

Control Flow

```
color = 'red'


if color == 'blue'
   # blue code
elsif color == 'green'
  # green code
else
  # all other colors code
end
```

Ruby vs JavaScript

```ruby
age = 50

if age >= 18
  # Can vote
else
  # Cannot vote
end
```

```javascript
var age = 50;

if (age >= 18) {
  // Can vote
}
else {
  // Cannot vote
}
```

Boolean Operators

And (&&)

- Returns true only if **all** parts of the statement are true
- “If the temperature is below 32 degrees F, ***and*** there is precipitation falling from the sky, then it is snowing.”

And (&&)

```
3 == 3 && 10 > 5

=> true
```

And (&&)

```
3 == 3 && 10 > 5

=> true
```

```
3 == 3 && 5 > 10

=> false
```

Or (| |)

- Returns true if **any** parts of the statement are true
- “If the temperature is below 32 degrees F, ***or*** there is precipitation falling from the sky, then the weather is bad.”

Or (||)

```
3 == 3 || 10 > 5

=> true
```

Or (||)

```
3 == 3 || 10 > 5

=> true
```

```
3 == 3 || 5 > 10

=> true
```

Not (!)

- Negates a statement
- x != 3
- `!true` evaluates to `false`, `!false` evaluates to `true`

Not (!)

“It is snowing”

```
if temperature_below_32 && there_is_precipitation
  # it is snowing

end
```

Not (!)

```
# it is snowing

temperature_below_32 = true

there_is_precipitation = true

_

temperature_below_32 && there_is_precipitation
```

Not (!)

```
# it is snowing

temperature_below_32 = true

there_is_precipitation = true

!

temperature_below_32 && there_is_precipitation

=> true && true
```

Not (!)

```
# it is snowing

temperature_below_32 = true

there_is_precipitation = true

!

temperature_below_32 && there_is_precipitation

=> true && true

=> true
```

Not (!)

```
# it is snowing

temperature_below_32 && there_is_precipitation
```

Not (!)

```
# it is snowing

temperature_below_32 && there_is_precipitation
```

```
# it is NOT snowing

!(temperature_below_32 && there_is_precipitation)
```

Not (!)

```
# it is NOT snowing

temperature_below_32 = true

there_is_precipitation = false

_

!(temperature_below_32 && there_is_precipitation)
```

Not (!)

```
# it is NOT snowing

temperature_below_32 = true

there_is_precipitation = false
```

_

```
!(temperature_below_32 && there_is_precipitation)

=> !(true && false)
```

Not (!)

```
# it is NOT snowing

temperature_below_32 = true

there_is_precipitation = false
```

_

```
!(temperature_below_32 && there_is_precipitation)

=> !(true && false)

=> !(false)
```

Not (!)

```
# it is NOT snowing

temperature_below_32 = true

there_is_precipitation = false
```

_

```
!(temperature_below_32 && there_is_precipitation)

=> !(true && false)

=> !(false)

=> true
```

Rewriting “Not”s

De Morgan’s Law: `!(A && B) = !A || !B`

Rewriting “Not”s

De Morgan’s Law: `!(A && B) = !A || !B`

```
!(temperature_below_32 && there_is_precipitation)
```

Rewriting “Not”s

De Morgan’s Law: `!(A && B) = !A || !B`

```
!(temperature_below_32 && there_is_precipitation)

=> !temperature_below_32 || !there_is_precipitation
```

Rewriting “Not”s

```
temperature_below_32 = true

there_is_precipitation = false

-

!temperature_below_32 || !there_is_precipitation
```

Rewriting “Not”s

```
temperature_below_32 = true

there_is_precipitation = false

-

!temperature_below_32 || !there_is_precipitation

=> !true || !false
```

Rewriting “Not”s

```
temperature_below_32 = true

there_is_precipitation = false

-

!temperature_below_32 || !there_is_precipitation

=> !true || !false

=> false || true
```

Rewriting “Not”s

```
temperature_below_32 = true

there_is_precipitation = false

-

!temperature_below_32 || !there_is_precipitation

=> !true || !false

=> false || true

=> true
```

Boolean Operators Recap

- `&&`, `x && y`, true if x *and* y are both true
- `||`, `x || y`, true if either x *or* y are true
- `!`, `!true`, negates a statement or value

Boolean Operators Recap

```
!((x && y) || (!x && z)) && (x || y || !z)
```

Presence

- Does it exist or not?
    - “Does this user exist?”
    - “Is the user logged in?”
    - “Does the user have any posts?”

Presence

“Does the user have any friends?”

```
friends = ['Alice', 'Bob']
```

Presence

“Does the user have any friends?”

```
friends = ['Alice', 'Bob']

_
```

```
if <user-has-friends???>

  # do things with the friends

end
```

Demo

Episode Recap

- Boolean operators and equality (`&&`, `||`, `!`, `>=`)
- Base control flow (`if / else`)
- Presence or existence (“truthyness”)

Homework

- Practice using equality, comparison, and boolean operators
  - Evaluate: `!((x && y) || (!x && z)) && (x || y || !z)`
  - `x = false, y = true, z = false`
- Practice modeling sentences using boolean statements
  - Ex: “If it’s a Sunday and the hour is before 12, then the restaurant serves brunch.”
  - `is_sunday = true, hour = 9`
- Practice diverting code using `if/elsif/else` statements
- Presence or existence (“truthyness”)

Thanks!

@johnmosesman