Conditionals

Boolean Logic
Relational Expressions
Logical Operators
Numerical Precision

Chapter 5, Examples 5-10, 5-11, 5-12, 5-13, 5-15, 5-16, 5-17, 5-18, 5-19, Robot 3 (p. 84-86).

Pick a number
Boolean Expressions

- An expression that evaluates to **true** or **false**.
- **Relational** boolean expressions:
  - is your number greater than 50?
  - is your number less than 40?
  - is your number equal to 73?
  - is your number greater than or equal to 75?
  - is your number less than or equal to 37?
  - is your number not equal to 99?

Relational Operators

\[
\begin{align*}
&> & \text{is your number greater than 50?} \\
&\text{number} & > 50 \\
&< & \text{is your number less than 40?} \\
&\text{number} & < 40 \\
&=== & \text{is your number equal to 73?} \\
&\text{number} & === 73 \\
&\geq & \text{is your number greater than or equal to 75?} \\
&\text{number} & \geq 75 \\
&\leq & \text{is your number less than or equal to 37?} \\
&\text{number} & \leq 37 \\
&\neq & \text{is your number not equal to 99?} \\
&\text{number} & \neq 99
\end{align*}
\]
// my number to guess
let number = 73;

// a relational expression used to guess my number
print(number > 50);

---

**JavaScript Equality Relational Operators**

- In JavaScript, there are two equality operators: `==` and `===`
  - You should use `===` because it's a strict comparison
  - Same goes for `!=` and `!==` operators, use `!==`
    (In other languages, there is only `==` and `!=`, and these are strict)
True or False?

```javascript
let a = 8;
let b = 5;
6 < a
a / 2 > 6
a * 5 === 8 * b
b !== 5
```

A. true  
B. false  
C. neither true or false

---

**Conditional Statement**

- *if* a boolean expression is **true**, *then* execute a block of code

```javascript
if (number > 50) {
// code to execute if true
}
```
animation

using a conditional statement to “loop” an animation

Starter: https://editor.p5js.org/cs105/sketches/Mcj4bsUNj
https://editor.p5js.org/cs105/sketches/xkfOldfyl

conditional_dot1 (if)

...  

// the conditional statement  
if (mouseX > width / 2) {  
  fill("#FF0000"); // red  
  ellipse(0.75 * width, 50, 30, 30);  
}  

...  

https://editor.p5js.org/cs105/sketches/DorsbS_Hk
Nested Code Blocks

```javascript
function draw() {
  background(220);

  // print debug info
  print(mouseX, mouseX > width / 2);

  // the conditional statement
  if (mouseX > width / 2) {
    fill("#FF0000"); // red
    ellipse(0.75 * width, 50, 30, 30);
  }

  // separating line
  line(width / 2, 0, width / 2, height);
}
```

see “06 Conditionals (trace)”
What can this code draw?

```
function draw() {
  background(200);
  fill(0); // black
  if (mouseX < 50) {
    fill(255); // white
  }
  ellipse(mouseX, mouseY, 20);
}
```

A, B, and C are all possible.

Stop for Gas Analogy

- A single if statement is like deciding if you should stop for gas
  - the if code block is something you do if a condition is true
    (if tank is low, then stop and get gas)
  - Regardless, you still continue down the road ...
### conditional_dot2 (if else)

```javascript
// the conditional statement
if (mouseX > width / 2) {
    fill("#FF0000"); // red
    ellipse(0.75 * width, 50, 30, 30);
} else {
    fill("#0000FF"); // blue
    ellipse(0.25 * width, 50, 30, 30);
}

```

### Detour Analogy

- **if else** is like having to take a bridge construction detour:
  - the **if code block** is an alternative path
    (if bridge is under construction, then take detour road)
  - the **else code block** is the default path
    (else cross bridge like normal)
  - Regardless of choice, you end up on the same road after
What can this code draw?

```javascript
function draw() {
  background(200);
  if (mouseY > 50) {
    fill(0); // black
  } else {
    fill(255); // white
  }
  ellipse(mouseX, mouseY, 20, 20);
}
```

---

### conditional_dot3 (if else if)

```javascript
// the conditional statement
if (mouseX > width / 2) {
  fill("#FF0000"); // red
  ellipse(0.75 * width, 50, 30, 30);
} else if (mouseX < width / 2) {
  fill("#0000FF"); // blue
  ellipse(0.25 * width, 50, 30, 30);
} else {
  background(random(255));
}
```

[Visit the P5.js sketch online](https://editor.p5js.org/cs105/sketches/M2WPtcp1I)
Two Common Logic Errors with if statements

Adding semicolon after the boolean expression means **no code block**. A code block with no "if" means always execute.

```javascript
if (mouseX > 50); {  
    fill("#FF0000"); // red  
    ellipse(0.75 * width, 50, 30, 30);  
}
```

Missing "else" makes **two** different (and independent) if statements.

```javascript
// the conditional statement
if (mouseX > 50) {  
    fill("#FF0000"); // red  
    ellipse(0.75 * width, 50, 30, 30);  
} else {  
    background(random(255));  
}
```

**shaper**

checking what key was pressed:

```javascript
if (key === 'c') {  
    shape = 1;  
} else if (key === 'r') {  
    shape = 2;  
} else if (key === 't') {  
    ...  
}
```

using a variable to save 'state':

```javascript
if (shape === 1) {  
    ellipse( ... );  
} else if (shape === 2) {  
    rect( ... );  
} else if (shape === 3) {  
    triangle( ... );  
}
```

[https://editor.p5js.org/cs105/sketches/d69-jjtUT](https://editor.p5js.org/cs105/sketches/d69-jjtUT)
Equality vs. Assignment Operators

Equality Operator

- is the left value equal to the right value?
- e.g. state === 1 means "is state equal to 1?"
- left and right can be anything that reduces to a single value (a variable, a function that returns a value, a number, an expression)

Assignment Operator

- assign the right side value to the variable on left
- e.g. state = 1 means “assign 1 to state”
- left must be a variable, right can be anything that reduces to a single value (a variable, a function that returns a value, a number, an expression)

What does this code draw after 10 frames?

```javascript
let a = 0;
function draw() {
  background(200);
  if (a < 5) {
    ellipse(50, 50, 50, 50);
  } else {
    line(0, 0, 99, 99);
  }
  a = a + 1;
}
```
What does this code draw after 10 frames?

```javascript
let a = 0;
function draw() {
    background(200);
    if (a < 5) {
        ellipse(50, 50, 50, 50);
    } else if (a > 5) {
        line(0, 0, 99, 99);
    } else {
        a = a + 1;
    }
}
```

Remember: the default canvas size is 100 by 100.

---

**Numerical Representation**

- A civil engineer doesn't care about the difference between 10 meters and 10.0001 meters
- 0.0001 meters is a huge difference for a microchip designer, but all measurements will be less than about 0.01 meters
- A physicist needs to use the speed of light (about 300000000) and Newton's gravitational constant (0.0000000000667) in the same equation
Numerical Precision

- How many numbers are there between 0 and 1?
- How many decimal points in one-third (1/3)?
- Computers can not always do exact math
- This has implications for equality testing ...

```javascript
let y = 0;

function setup() {
    createCanvas(50, 200);
    background(220);
}

function draw() {
    if (y === 1) {
        // this code block may never be executed due to numerical precision
        stroke(255, 0, 0);
        print("y is exactly 3");
    }
    line(0, y * 100, width, y * 100);
}

function keyPressed() {
    y = y + 0.1;
    print(y);
}

https://editor.p5js.org/cs105/sketches/XiSJxkZip
```
Guessing Game

Boolean Expressions

- An expression that evaluates to true or false.
- boolean expressions with logical operators:
  - is your number greater than 50 and less than 60?
  - is your number less than 40 or greater than 80?
  - is your number not less than 32?
**Logical Operators**

is your number greater than 50 and less than 60?

number > 50 && number < 60

is your number less than 40 or greater than 80?

number < 40 || number > 80

is your number not less than 32?

! (number < 32)

---

`&&` means "and"

`||` means "or"

`!` means "not"
Logical Operator “Truth Table”

<table>
<thead>
<tr>
<th>Boolean Expression</th>
<th>Evaluates To</th>
</tr>
</thead>
<tbody>
<tr>
<td>true &amp;&amp; true</td>
<td>true</td>
</tr>
<tr>
<td>true &amp;&amp; false</td>
<td>false</td>
</tr>
<tr>
<td>false &amp;&amp; false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td></td>
</tr>
<tr>
<td>true</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td></td>
</tr>
<tr>
<td>!true</td>
<td>false</td>
</tr>
<tr>
<td>!false</td>
<td>true</td>
</tr>
</tbody>
</table>

Logical Operator Examples

let a = 15;
let b = 5;

(a > 10 && b < 10)

(a < b || b < a)

(a === 10 && a < 20 && b > 0)

(a < 10 || 12 > b * 2)

(a + b > 16 || b - a > 0)

(a > 10 && a < 12 || b > 3 && b < 7)

!(a > 10)
// top-right
if (mouseX > width / 2 && mouseY < height / 2) {
    fill("#FF0000"); // red
}

// left side
} else if (mouseX < width / 2) {
    fill("#0000FF"); // blue

// anything else
} else {
    noFill();
    background(random(255)); // flash
}

// draw circle using fill set in conditional
ellipse(50, 50, 30, 30);

https://editor.p5js.org/cs105/sketches/_Fd3c-OjW

extend shaper to work with CAPS LOCK on

// if little c OR big C
if (key === 'c' || key === 'C') {
    shape = 1;
    ...

https://editor.p5js.org/cs105/sketches/d69-jjtUT
What is drawn on the 6th frame?

```javascript
let a = 0;
let b = 0;
function draw() {
    background(220);
    if (a < 10 || b > 20) {
        ellipse(25, 25, 30, 30);
    } else {
        rect(60, 60, 30, 30);
    }
    a = a + 1;
}
function mousePressed() {
    b = 100;
}
```

What is drawn on the 27th frame?

```javascript
let a = 0;
let b = 0;
function draw() {
    background(220);
    if (a < 10 || b > 20) {
        ellipse(25, 25, 30, 30);
    } else {
        rect(60, 60, 30, 30);
    }
    a = a + 1;
}
function mousePressed() {
    b = 100;
}
```
What is drawn 28 frames after a mouse press?

Creating a rectangle hit test

- Also called a “rollover” or “mouseover”
- (blackboard)
Boolean Variables

- The result of a relational expression is either true or false
- “Boolean” variables store true or false

```javascript
let a = 9 < 10;
let b = false;
print(a); // prints true
print(b); // prints false
```
let day = true;

function draw() {
    if (day === true) {
        background("#83E7FF"); // blue sky
    } else {
        background("#01396F"); // night sky
    }
}

function mousePressed() {
    // "not day" means the opposite
    // of day's current boolean value
    day = !day;
}

What is drawn on the 3rd frame?
mouselIsPressed, mousePressed()

- mousePressed() is an event function
  - it's called once when the mouse button is pressed
- mouseIsPressed is a built-in boolean variable
  - it's true when the mouse button is pressed, false otherwise
- Same for keyIsPressed and keyPressed()

draw (extra demo)

draw when mouse button is held down
erase when any key is held down
erase whole drawing with “x” key

pmouseX
pmouseY

mouselIsPressed
keyIsPressed
keyPressed()