Grow with Google: Lesson 12

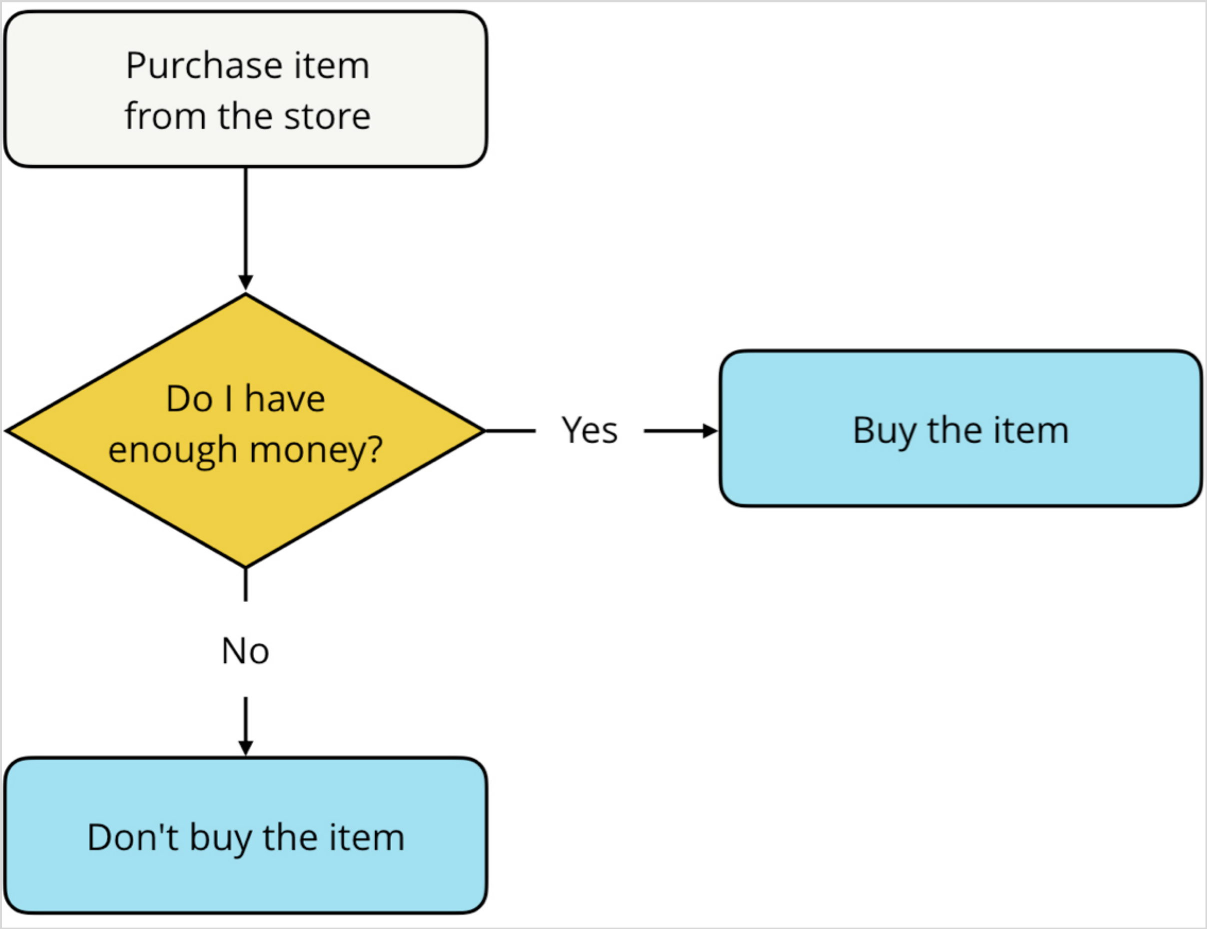
Conditionals

1. Intro to Conditionals

- The goal of code is to solve a problem.
- The steps you use to solve that problem is an algorithm.

2. Quiz: Flowcharts (3-1)

- A flowchart outlines the solution to a problem through a series of logical statements.
- The order in which statements are evaluated and executed is called the control flow.
- Booleans are like the yes and no on the flow chart.



3. Flowchart to Code

- We can represent decision points and multiple results with `if...else` statements.

```
if (money>= price) {
  console.log(“Buy the hammer.”);
} else {
  console.log(“Don’t buy the hammer.”);
}
```

4. If...Else Statements

- The value inside the if statement is always converted to true or false.
    - Depending on the value, the code inside the if statement is run or the code inside the else statement is run, but not both.
- You can use an if statement on its own, but an else statement must be preceded by an if statement.

```
if (// this expression is true) {
  // run this code
} else {
  // run this code
}
```

5. Else If Statements

- Add additional conditionals with an `else if` statement.
- The final else acts as the default.

```
var weather = "sunny";

if (weather === "snow") {
  console.log("Bring a coat.");
  // First condition.
} else if (weather === "rain") {
  console.log("Bring a rain jacket.");
  // Second condition.
} else {
  console.log("Wear what you have on.");
  // Weather is presumed to be sunny by default.
}
```

6. Quiz: Even or Odd (3-2)

```
var number = 2;
```

```
if (number % 2 === 0) { // Modulo returns a remainder.
  console.log("even"); // True
} else {
  console.log("odd"); // This code is never run.
}
```

7. Quiz: Musical Groups (3-3)

```
var musicians = 1;

if (musicians <= 0) {
  console.log("not a group");
} else if (musicians == 1) {
    console.log("solo");
} else if (musicians == 2) {
    console.log("duet");
} else if (musicians == 3) {
    console.log("trio");
} else if (musicians == 4) {
    console.log("quartet");
} else {
    console.log("this is a large group");
} // returns solo
```

There must be a more efficient way to do this.

8. Quiz: Murder Mystery (3-4)

```
var room = "ballroom";
var suspect = "Mr. Kalehoff";

var weapon = "";
var solved = false;

if (room === "ballroom") {
    weapon = "poison";
    if (suspect === "Mr. Kalehoff") {
        solved = true;
    }
} else if (room === "gallery") {
    weapon = "trophy";
    if (suspect === "Ms. Van Clarke") {
        solved = true;
    }
} else if (room === "billiards room") {
    weapon = "pool stick";
    if (suspect === "Mrs. Sparr") {
        solved = true;
    }
} else {
    weapon = "knife";
    if (suspect === "Mr. Parkes") {
        solved = true;
    }
}

if (solved) {
  console.log(suspect + " did it in the " + room + " with the " + weapon +
"!");
}
```

Took me too long to do this because I used the `&&` operators instead of nested `if` loops. Udacity really wants you to take this step by step, even if it's inefficient.

9. More Complex Problems

- We can test for multiple conditions with logical operators.

10. Logical Operators

```
&&    Logical AND  value1 && value2    Returns true if both value1 and value2
evaluate to true.
||    Logical OR   value1 || value2    Returns true if either value1 or value2
(or even both!) evaluates to true.
! Logical NOT !value1 Returns the opposite of value1. If value1 is true,
then !value1 is false.
```

11. Logical AND and OR

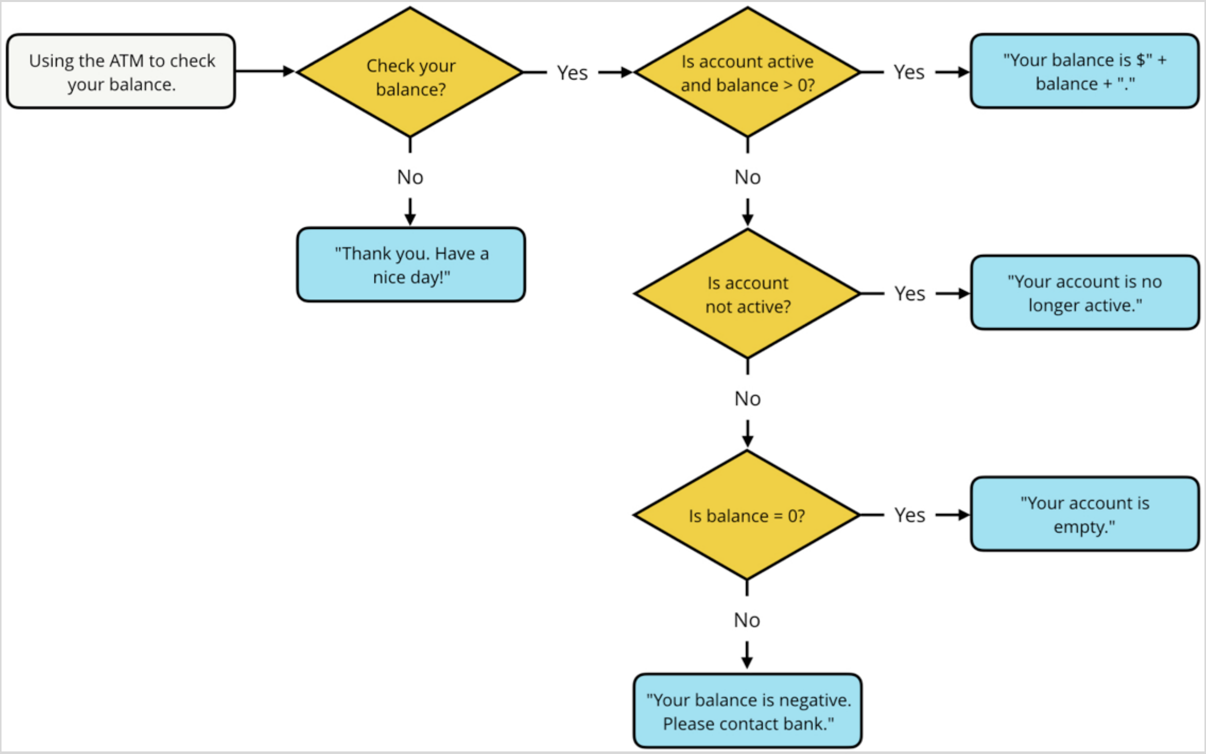
- Truth tables are used to represent the result of all the possible combinations of inputs in a logical expression.

```
&& (AND)
A     B       A && B
true  true    true
true  false   false
false true    false
false false   false

|| (OR)
A     B       A || B
true  true    true
true  false   true
false true    true
false false   false
```

- In both tables, there are specific scenarios where regardless of the value of B, the value of A is enough to satisfy the condition.
- This behavior is called short-circuiting because it describes the event when later arguments in a logical expression are not considered because the first argument already satisfies the condition.

12. Quiz: Checking your Balance (3-5)


Using the ATM to check your balance.
Check your balance?
Yes
No
"Thank you. Have a nice day!"
Is account active and balance > 0?
Yes
"Your balance is $" + balance + "."
No
Is account not active?
Yes
"Your account is no longer active."
No
Is balance = 0?
Yes
"Your account is empty."
No
"Your balance is negative. Please contact bank."

- I had to really think this one through and sketched some things out on my whiteboard. Even with the flow diagram, I couldn't really tell how to frame the conditions or which cases meant I could end the tests.
- I think "Is account not active?" and of threw me, because it's a boolean on a negative statement.

```
var balance = 325;
var checkBalance = true;
var isActive = true;

if (checkBalance === true) {
    if (isActive === true && balance > 0) {
        console.log("Your balance is $" + balance.toFixed(2) + ".");
    } else if (isActive === true && balance === 0) {
        console.log("Your account is empty.");
    } else if (isActive === true && balance < 0) {
        console.log("Your balance is negative. Please contact bank.");
    } else {
        console.log("Your account is no longer active.");
    }
} else {
    console.log("Thank you. Have a nice day!");
}
```

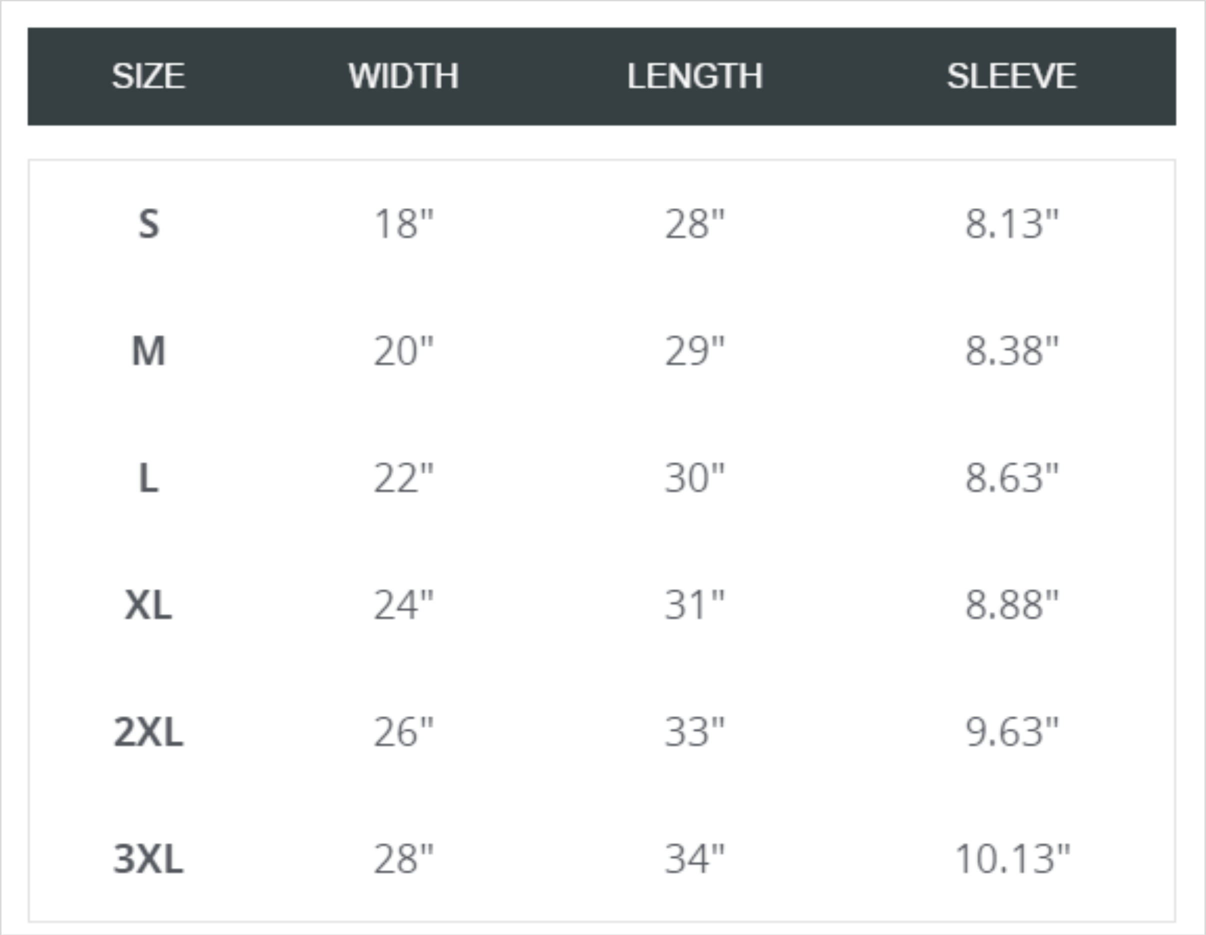
13. Quiz: Ice Cream (3-6)

- I drew out my own flow diagram. Things went much faster.

```
var flavor = "vanilla";
var vessel = "cone";
var toppings = "sprinkles";

if ((flavor === "vanilla" || flavor === "chocolate") && (vessel === "cone" ||
vessel ==="bowl") && (toppings === "sprinkles" || toppings === "peanuts")) {
    console.log("I'd like two scoops of " + flavor + " ice cream in a " +
vessel + " with " + toppings + ".");
}
```

14. What do I Wear? (3-7)



SIZE	WIDTH	LENGTH	SLEEVE
S	18"	28"	8.13"
M	20"	29"	8.38"
L	22"	30"	8.63"
XL	24"	31"	8.88"
2XL	26"	33"	9.63"
3XL	28"	34"	10.13"

- I struggled with this one for a long time. My original solutions failed the grader, passing tests with different data, but my final solution passed the grader, failing tests with different data.
- I realized that the reason the final solution works is because it's strict. All 3 variables need

to be within range of the same size, or the console should yield a `N/A` result.

```
var shirtWidth = 19;
var shirtLength = 28;
var shirtSleeve = 8.21;

if ((shirtWidth >= 18 && shirtWidth < 20) && (shirtLength >= 28 && shirtLength
< 29) && (shirtSleeve >= 8.13 && shirtSleeve < 8.38)) {
  console.log("S");
} else if ((shirtWidth >= 20 && shirtWidth < 22) && (shirtLength >= 29 &&
shirtLength < 30) && (shirtSleeve >= 8.38 && shirtSleeve < 8.63)) {
  console.log("M");
} else if ((shirtWidth >= 22  && shirtWidth < 24) && (shirtLength >= 30 &&
shirtLength < 31) && (shirtSleeve >= 8.63 && shirtSleeve < 8.88)) {
  console.log("L");
} else if ((shirtWidth >= 24 && shirtWidth < 26) && (shirtLength >= 31 &&
shirtLength < 33) && (shirtSleeve >= 8.88 && shirtSleeve < 9.63)) {
  console.log("XL");
} else if ((shirtWidth >= 26 && shirtWidth < 28) && (shirtLength >= 33 &&
shirtLength < 34) && (shirtSleeve >= 9.63 && shirtSleeve < 10.13)) {
  console.log("2XL");
} else if ((shirtWidth > 26 && shirtWidth <= 28) && (shirtLength > 33 &&
shirtLength <= 34) && (shirtSleeve > 9.63 && shirtSleeve <= 10.13)) {
  console.log("3XL");
} else {
    console.log("N/A");
} // S
```

15. Advanced Conditionals

- Truthy and Falsy Values
- Ternary Operator
- Switch Statement

16. Truthy and Falsy

- Every value in JavaScript has an inherent boolean value.
- When that value is evaluated in the context of a boolean expression, the value will be transformed into that inherent boolean value.

- A value is falsy if it converts to false when evaluated in a boolean context.
  1. the Boolean value `false`
  2. the `null` type
  3. the `undefined` type
  4. the number `0`
  5. the empty string `””`
  6. the odd value `NaN`

     MDN NaN (Not-A-Number)
- A value is truthy if it converts to true when evaluated in a boolean context.
  - true
  - 42
  - "pizza"
  - "0"
  - "null"
  - "undefined"
  - {}
  - []
- Anything that is not false is truthy.

17. Ternary Operator

- Test whether something is true or false in a conditional with shorthand.
  - `if(isActive)` === `if(isActive === true)`
  - `if(!isActive)` === `if(isActive === false)`
- The ternary operator `?` is a shortcut for simple conditionals.
  - `conditional ? (if condition is true) : (if condition is false)`
  - The ternary operator is a stand-in for `if` while the colon is a stand-in for `else`.
  - “Ternary” is an adjective meaning “composed of 3 parts.” These parts are the conditional, the code run if it is true, and the code run if it is false.

```
var isGoing = true;
var color;

if (isGoing) {
  color = "green";
} else {
  color = "red";
}

console.log(color);
```

- With the ternary operator:

```
var isGoing = true;
var color = isGoing ? "green" : "red";
console.log(color);
```

18. Quiz: Navigating the Food Chain (3-8)

- This one took me a little effort to wrap my head around. I started with the if else statement version.

```
var eatsPlants = true;
var eatsAnimals = false;

if (eatsPlants && eatsAnimals) {
  console.log("omnivore");
} else if (eatsPlants && !eatsAnimals) {
  console.log("herbivore");
} else if (!eatsPlants && eatsAnimals) {
  console.log("carnivore");
} else {
  console.log(undefined);
}
```

- Then I did the ternary version. I didn't get how they should be nested at first, but the optional parentheses help a great deal.

```
var eatsPlants = true;
var eatsAnimals = false;

var category = eatsPlants ? (eatsAnimals ? "omnivore": "herbivore") :
(eatsAnimals ? "carnivore" : undefined);

console.log(category);
```

- Resources
  - MDN Conditional (Ternary) Operator
  - DNA Web Agency Ternary Operator: Usage and Examples

  - ESLint Multiline Ternary Expressions

19. Switch Statement

- Switch statements are best used when else if conditions are repeated based on the same value.
- By default, fall-through occurs once the correct case is identified.
- Use breaks to prevent fall-through, though sometimes it is useful.

```
// Conditionals

if (option === 1) {
  console.log("You selected option 1.");
} else if (option === 2) {
  console.log("You selected option 2.");
} else if (option === 3) {
  console.log("You selected option 3.");
} else if (option === 4) {
  console.log("You selected option 4.");
} else if (option === 5) {
  console.log("You selected option 5.");
} else if (option === 6) {
  console.log("You selected option 6.");
}

// Switch
var option = 3;

switch (option) {
  case 1:
    console.log("You selected option 1.");
    break;
  case 2:
    console.log("You selected option 2.");
    break;
  case 3:
    console.log("You selected option 3.");
    break;
  case 4:
    console.log("You selected option 4.");
    break;
  case 5:
    console.log("You selected option 5.");
    break;
  case 6:
    console.log("You selected option 6.");
    break; // Technically not needed, since there are no additional statements to fall-through.
}
```

- 30 days hath September, April, June, and November; February has 28 alone; All the rest have 31; Except in leap year, that’s the time, when February’s days are 29.

```
// There is no case included for Leap Year.

var month = 2;

switch (month) {
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12:
    days = 31;
    break;
  case 4:
  case 6:
  case 9:
  case 11:
    days = 30;
    break;
  case 2:
    days = 28;
}

console.log("There are " + days + " days in this month.");
```

20. Falling-Through

- Use falling-through when code follows a hierarchical structure.
- Add a default case when none of the values match the other cases.
- Use breaks to avoid unexpected behavior for dissimilar cases.

```
var tier = "nsfw deck";
var output = "You’ll receive "

switch (tier) {
  case "deck of legends":
    output += "a custom card, ";
  case "collector's deck":
    output += "a signed version of the Exploding Kittens deck, ";
  case "nsfw deck":
    output += "one copy of the NSFW (Not Safe for Work) Exploding Kittens card game and ";
  default:
    output += "one copy of the Exploding Kittens card game.";
}

console.log(output);
```

21. Quiz: Back to School (3-9)

```
var education = "a Doctoral degree";

var salary;

switch (education) {
    case "no high school diploma":
        salary = 25636;
        break;
    case "a high school diploma":
        salary = 35256;
        break;
    case "an Associate\'s degree":
        salary = 41496;
        break;
    case "a Bachelor\'s degree":
        salary = 59124;
        break;
    case "a Master\'s degree":
        salary = 69732;
        break;
    case "a Professional degree":
        salary = 89960;
        break;
    case "a Doctoral degree":
        salary = 84396;
}

console.log("In 2015, a person with " + education + " earned an average of $" +
salary.toLocaleString("en-US") + "/year.");
```

22. Lesson 3 Summary

- Break down problems into steps.
- Use conditional statements and logical operators.
- Remember advanced strategies to simplify code and decrease file size.