

Lesson

Wednesday

Introduction to Programming

(/introduction-to-programming)

/ JavaScript and Web Browsers

(/introduction-to-programming/javascript-and-web-browsers)

/ Branching

Text

Cheat sheet

Terminology

- **Branching:** Determining the flow of your code based on certain conditions. (i.e.: *If* something is true, do one thing. If this same thing is false, do a *different* thing.)
- **Boolean:** Returns true or false. When JavaScript is attempting to discern whether a condition is true, it's looking for a boolean.
- **Comparison operators:** `===` , `>` , `<` , `>=` , `<=` .
- `=` sets a variable; `===` compares two things. Don't use `==` .

Summary

In this lesson we introduced ourselves to **branching** with `if` statements.

- There are a few ways to make branching logic. In this lesson we learned about `if` statements.
- We use comparison and equality operators to create conditions for `if/else if` statements to evaluate to `true` or `false`:
 - `if (4 > 5)`
 - `if (typeof "hello" === "string")`
 - `if (variableName === "add")`
- `=` sets a variable; `===` compares two things. Don't use `==`.
- `if` statements are made up of `if`, `else if`, and `else` statements. They must include one `if` statement, but it's not required that you use an `else if` or `else` statement in your branching. Somethings to note:
 - You can have an `if` statement all by itself (without an `else if` or an `else`).
 - You can have an `if ... else if` without an `else` statement.
 - You can have an `if ... else` without an `else if` statement.
- `if` and `else if` statements require conditions to be evaluated that are listed in parentheses.
- There is no condition (and no parentheses) for `else` statements, since they designate what to do if an `if` or `else if` condition has not evaluated to `true`.

Examples

Comparison operators return booleans:

```
3 > 2;  
// returns true
```

One branch:

```
if (age >= 21) {  
    document.querySelector('#drinks').removeAttribute("class");  
}
```

Two branches:

```
if (age >= 21) {  
    document.querySelector('#drinks').removeAttribute("class");  
} else {  
    document.querySelector('#under-21').removeAttribute("class");  
}
```

Three branches:

```
if (age > 21) {  
    drinkMenu.removeAttribute("class");  
} else if (age === 21) {  
    window.alert("Have some fun, you're just 21!");  
    drinkMenu.removeAttribute("class");  
} else {  
    under21Message.removeAttribute("class");  
}
```

Branching can use a variable whose value is a boolean:

```
const over21 = true;  
if (over21) {  
    drinkMenu.removeAttribute("class");  
}
```

[Previous \(/introduction-to-programming/javascript-and-web-browsers/practical-tips-for-researching-web-apis\)](#)

[Next \(/introduction-to-programming/javascript-and-web-browsers/practice-branching\)](#)

Lesson 59 of 75

Last updated March 24, 2023

[disable dark mode](#)



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.