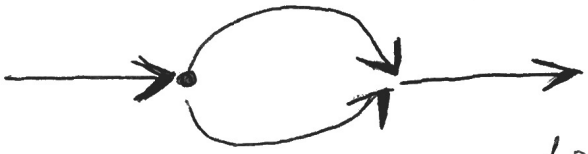
JavaScript: Conditionals, Control Flow, & Logical Operators

The Complete Web Developer in 2019

The Complete Web Developer in 2019
Zero to Mastery
Andrei Neagoie
Lecture Notes by Stephanie

CONTROL FLOW

execute code in straight line



conditional execution

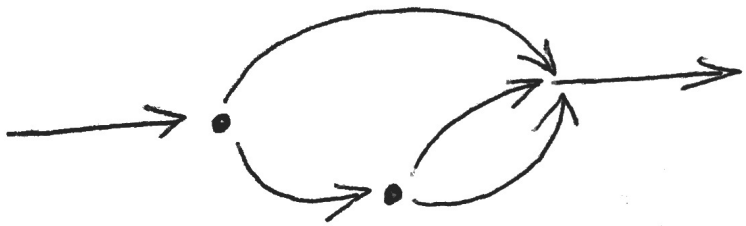
JAVASCRIPT CONDITIONALS

aka "conditional statements"

- if
- else
- else if
- ternary operator
- switch

expressions return values

note: an if statement is not an expression, so it does NOT require a semicolon



conditional execution can be complex

ex: if ( ) {
    ~~;
} else if ( ) {
    ~~;
} else {
    ~~;
}

if - else statement

```
> if (name === "Billy") {
      alert("hi Billy!");
  } else {
      alert("hmmm I dont know you");
  }
```

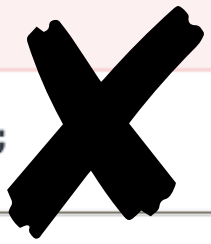
Note: these are conditional statements, do NOT need semicolon

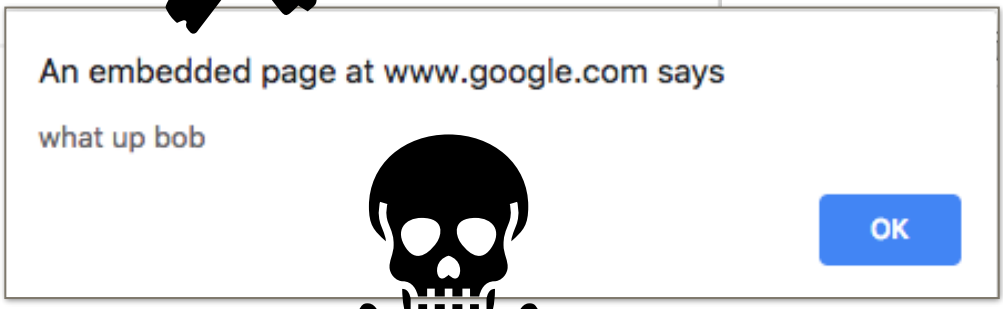
else if statement

```
> if (name === "Billy") {
      alert("hi Billy!");
  } else if (name === "Susy") {
      alert("hi Susy!");
  } else {
      alert("I dont know you");
  }
```

Note: if you use = instead of ===, you will create a new variable instead of checking the condition!!! Be very careful!!!

```
> firstName5
▶ Uncaught ReferenceError: firstName5 is not defined    VM1709:1
      at <anonymous>:1:1
> if (firstName5="Bob") {
      alert("what up bob");
  }
< undefined
> firstName5
< "Bob"
```





ternary operator

For this function:

```
> function isUserValid(bool) {
      return bool;
  }
```

Instead of using this:

```
> function conditionFunxn(boolEntry) {
      if (isUserValid(boolEntry)) {
          return "Enter here";
      } else {
          return "Access denied";
      }
  }
```

To get:

```
> conditionFunxn(true)
<- "Enter here"
> conditionFunxn(false)
<- "Access denied"
```

We can use a ternary operator!

(*condition*) ? (*expression1*) : (*expression2*);

if *condition* = true, then *expression1*

if *condition* = false, then *expression2*

```
> function ternaryOperatorFunxn(boolEntry) {
      return (isUserValid(boolEntry) ?
              "Enter here" : "Access denied");
  }
```

```
> ternaryOperatorFunxn(true)
<- "Enter here"
> ternaryOperatorFunxn(false)
<- "Access denied"
```

.

Ternary Operator: Daisy Chained Ternary Statememts

```
H = (C == 0)          // Is C zero?
    ? null
    : (V == r)        // Is V equal to r?
    ? (g - b) / C
    : (V == g)        // Is V equal to g?
    ? (b - r) / C + 2
    : (r - g) / C + 4; // Fallback (default) value
```

switch

Instead of using if - else if - else if - else if - else statement:

```
> function moveCommand(direction) {
      var whatHappens;
      switch (direction) {
          case "forward":
              whatHappens = "encounter monster!";
              break; // exit switch statement
          case "back":
              whatHappens = "arrive at home";
              break;
          case "right":
              whatHappens = "find river";
              break;
          case "left":
              whatHappens = "see troll";
              break;
          default:
              whatHappens = "please enter valid direction";
      }
      return whatHappens;
  }
```

To get:

```
> moveCommand("back")
<· "arrive at home"
> moveCommand("forward")
<· "encounter monster!"
> moveCommand("lalalalal")
<· "please enter valid direction"
```

Exercise: Advanced Control Flow

Section 13, Lecture 134

It's time to code some javascript! Get your sublime text ready for this exercise, and use Google Chrome javascript console to test your code. You can find the exercise file and the solution file attached. Good luck!

Resources for this lecture

exercise2-SOLUTIONS.js
exercise2.js

```
> //#1 change this function into a ternary and assign it
  // to variable called experiencePoints
  function experiencePoints() {
      if (winBattle()) {
          return 10;
      } else {
          return 1;
      }
  }

  //solution:
  function expPointsTernary() {
      return (winBattle() ? 10 : 1);
  }
```

```
> //Using this function, answer the questions below:
  function moveCommand(direction) {
      var whatHappens;
      switch (direction) {
          case "forward":
              break;
              whatHappens = "you encounter a monster";
          case "back":
              whatHappens = "you arrived home";
              break;
              break;
          case "right":
              return whatHappens = "you found a river";
              break;
          case "left":
              break;
              whatHappens = "you run into a troll";
              break;
          default:
              whatHappens = "please enter a valid direction";
      }
      return whatHappens;
  }
```

```
  //#2 return value when moveCommand("forward");
  undefined

  //#3 return value when moveCommand("back");
  "you arrived home"

  //#4 return value when moveCommand("right");
  "you found a river"

  //#5 return value when moveCommand("left");
  undefined
```

Javascript Logical Operators

&& and

|| or

! not

AND operator: &&

```
> if (firstName === "Bob" && lastName === "Smith") {
      alert("Hi Bob Smith");
  }
```

OR operator: ||

```
> if (name === "Billy" || name === "Ann") {
      alert("Hi Billy or Ann!");
  }
```

NOT operator: !

```
> !false
<· true
> !true
<· false
```

```
> name = "Barry"
<· "Barry"
> if (!(name === "Malia")) {
      "You're not Malia";
  }
<· "You're not Malia"
```

// Exercise 4
// Make a keyless car!
// This car will only let you drive if you are over 18. Make it do the following:

//using prompt() and alert(), ask a user for their age.
// IF they say they are below 18, respond with:
// "Sorry, you are too young to drive this car. Powering off"

// IF they say they are 18, respond with:
// "Congratulations on your first year of driving. Enjoy the ride!"

// IF they say they are over 18, respond with:
// "Powering On. Enjoy the ride!"

My solution:

```
> var age = Number(prompt("How old are you
  bro?"));
  if (age < 18) {
      alert("Sorry, too young kiddo");
  } else if (age === 18) {
      alert("Congrats, 18!!");
  } else {                        // age > 18
      alert("you old");
  }
```