**Road Trip Part 1**

* 1. **BOOLEANS, THE TRUE PATH**
* JS is used for Interactive web pages

*But...*

* In this lesson I’ll learn fund. Building blocks of JS lang.
* Then engage in JQuery
* Manipulate values with operators
* Create/store those values inside variables
* How to find, create, and run JS Source files

**THE PROMPT AND BASIC NUMBERS**

>symbol The JavaScript Prompt, aka “the Console

-> What gets returned from the code

- JS automatically recognizes numbers

*Example:*

>24

-> 24

**COMMON OPERATORS USED IN JS SYNTAX:**

Addition, Subtraction, Multiplication, Division, Modulus (return remainder after division)

Order of Operations: *PEMDAS*

**COMPARATORS**

Common Number Comparators used in JavaScript Syntax

**> greater than, < less than, == equals, != not equal, >= greater or equal, >= less or equal**

**Boolean – true or false**

**1.7 STRINGS ON SIGNS**

How JS stores and processes flat text

**CONCATENATION**

>“Raindrops on roses” + “And “ + “Whiskers on kittens”

->Raindrops on roses And Whiskers on kittens

* Works with numbers and their expressions, too.
* Does not provide a space

>”The meaning of life is “ + 42

->The meaning of life is 42

* Works with numbers and their expressions, too

>”Platform “ + 9 + “ and “ + ¾

->”Platform 9 and 0.75”

- Expressions get evaluated!

>”Platform “ + 9 + “ and ¾”

->”Platform 9 and 3/4”

**SPECIAL CHARACTERS INSIDE STRINGS**

Some characters need backslash *notation* in JS Strings

- Advances to the next “tab stop” (\t) (Tab space)

**>**”Flight #:**\t**921**\t\t**Seat**:\t**21C”

**->**”Flight #: 921 Seat: 21C”

- Adds a quotation mark without ending the string too early

- Console sees “” as ending/starting a string **– need a \ at the begin&end** *\”string\”*

**>**”Login Password:**\t\t\”**C3P0R2D2**\”**”

**->**”Login Password: “C3P0R2D2””

Some characters need backslash notation in JS Strings

* Places a backslash itself in the String
* Double backslash to produce *one* backslash

>”Origin**\\**Destination:**\t**Orlando(MCO)[**\\**London(LHR)](file:///\\London(LHR))”

->”Origin\Destination: Orlando(MCO)\London(LHR)”

**\n** shifts the printout to a “new line”

>”Departure:**\t**09:55A**\n**Arrival:**\t**14:55P”

->”Departure: 09:55A

->Arrival: 14:55P”

**STRING COMPARISONS**

Checking for matching strings and alphabetical ordering

>”The Wright Brothers” == “The Wright Brothers”

->true

>”The Wright Brothers” == “Super Mario Brothers”

->false

>”The Wright Brothers” != “the wright brothers”

->true

(case sensitive)

The length of strings can be accessed with the **.length** property

>”antidisestablishmentarianism”**.length**

-> 28

Spaces and any non-alphabetic characters are counted too!

>”red blue”

->8