bolean = True/False

string = sequence of characters (text)

**front end (HTML + CSS)**

Rober Calliau - HTML - invented at Cern (hyperlinks & HTML)

Hypertext - interactive

markup language - specify atributes

web browser = html viewer

html tags - heading tags <h1>(h1 top heading, h8 lower heading, etc)

paragraph tag <p>

unordered list <ul>

ordered list <ol>

     list <li> - for each new line of list

<a> (anchor) - tags hyperlinks

<a href="www.exemle.com"> </a> - href =  html atribute

<a href="www.exemle.com" target="\_blank"> </a> - opens in new window

<a href="www.exemle.com" target="\_self"> </a> - opens in current window/tab (default location, no need to write it))

absolute vs relative path

<a href="http://www.exemle.com/legal.html"> </a> - absolute path (used for liking toother servers)

<a href="legal.html"> </a> relative path (opens from same server as current page) (should when linking to files on own website)

<body> - all visible content should be inside, doesn't display anything but keeps page organzized

<head> - code to run before loading a page

<html> - only one per page, everything is inside

<DOCTYPE .... > -tells browser which version of HTML to use (type html to use latest version)

**nesting** tags (more tags inside each together, indeting keeps things clear)

**introducing (SQL)**

databases are everywhere, database uses tables

* unique identifier = primary key of the table
* first search commumns than search rows
* database runns the code and returns result
* recipee - must end with **;** separate multiple values by **'**
  + **select** \_\_\_\_; (column)
    - select \* (displays all the data in database)
  + **from**\_\_\_; (table)
  + **where** id = w (entry) f.ex where title = donald
* sort
  + **order by** \_\_\_ - sorts ascending order
    - order by \_\_\_\_ ASC - ascending
    - order by \_\_\_DESC - descending
  + comparing where  \_\_\_ > \_\_\_\_
  + **<>**not equal sign (blacklisting a querry)
* Multiple operators -
  + **AND** into where - to ad multiple criteria
  + **OR**
  + ' ' use when searching with strings
    - exemple code
      * select title, duration
      * from movies
      * where genre='Adventure';

**Javascript roadtrip**

* propmt (the console) input
* what get returned , output
* **Numeric Operators**
  + accepted operations: +, -, \*, /,
    - modulus - % returns remainder after division ( 10%6 = 4), has same hierarchy (importance) as a multiplication/dividision
    - oder of perations PEMDAS (paranthese, etc)
  + compertors >, <, >=, <=
  + ==  is equal?
  + !=  is not equal (output true or false)
* **Text**
  + **" "** contains strings
  + **+** concatenates stings/ numbers
  + " ".leght  returns lenght of a string (even spaces, etc)
* Formating
  + \t advance to next tab stop
  + \\ to write \ inside a string
  + \n shifts to new line