

Callbacks & MySQL

How do we declare functions?

```
function hollaback() {  
    // Some code  
}
```

What is a function expression?

```
var hollaback = function() {  
    /// Some code here  
}
```

What's interesting about that?

Callbacks

- we can assign functions as values to variables
- since functions are values, we can pass them as arguments to functions
- functions passed as arguments to other functions are called callbacks



The function receiving the callback does some work, then calls the passed function with its results


```
// First, weather.find executes a search for the weather in the user's location
// ...and reports results in the degreeFormat we specify
weather.find({ search: this.location, degreeType: "F" }, function(err, result) {
    // Then, one of two things happens:
    // 1) weather.find hits an error in its search and calls the function with an error
    // 2) weather.find finds the data it wants and passes that data as an argument
    if (err) console.log(err);

    console.log(JSON.stringify(result, null, 2));
});

// In either case, weather.find does its thing, then calls us back
```

This allows us to write code that deals with values we don't have yet—in this case, the weather data—which will run later, when we do have those values.

Activity: Callbacks (10 min)

See 16-Callbacks for instructions

Review: Callbacks

- In each example, func is a value that contains a function
 - This is why we can put parentheses after it—this calls the function
- When we pass a function a callback, we can do other work before we call the callback

Homework







Introduction to SQL & MySQL

SQL

- SQL (often pronounced "Sequel") stands for "Structured Query Language" and is a powerful programming tool that was specifically designed to allow programmers with the ability to create, populate, manipulate, and access databases so as to provide them with an easy method to deal with server-side storage.

MySQL

- MySQL (often pronounced "My Sequel") is a popular type of open source software that can be placed on a server so as to allow SQL commands to affect the data stored on the server.

Tables

- Data using SQL is stored in tables on the server much like those you would create in Microsoft excel or in Google spreadsheets, making the data easy to visualize and search through.

Install MySQL Workbench

Here <https://www.mysql.com/products/workbench/>

Demo: Creating a LocalHost Connection

- This type of connection allows us to create locally stored data on our computers as if they were an external server.
- This is a much better alternative to spending hundreds to thousands of dollars on buying a server itself for the purposes of practicing on.

MySQL Workbench

- Enter "Local Instance MySQL" as your connection name
- Make sure your connection is set to "Standard (TCP/IP)"
- Enter "localhost" as the Hostname
- Enter "3306" as the port for your connection.
- Enter your MySQL username into the Username section (Default is "root")
- Click on the Store In Vault/Keychain and enter in your MySQL password
- Leave the Default Schema field empty
- Click on the "Test Connection" button
 - If successful, hit okay and your new connection should appear on the main page.
 - It is imperative that the server be started for Workbench to be able to connect to the database.
- Now double-click on that connection, enter your password if necessary, and the SQL editor will appear.

Activity: Creating a LocalHost Connection (30 min)

Connections vs. Databases

- Note the "SCHEMAS" section on the left side of their page is pretty much empty other than including the built-in sys, sakila, and world databases. These databases would not appear on external servers.
- Connecting to a server does not mean that you have created a database on that server. This means that, even if we wanted to, we could not yet populate our localhost connection with data until we create a database to house the data.
- The connection is a road which leads to an empty lot. Before we can populate the area, we first need to lay the foundations for houses (create a database), and build the houses (create a table).

Quick Note on Schemas

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

Demo: Creating a Database

- `CREATE DATABASE animals_db;`
 - This line of code will create a MySQL database on the server
 - Don't forget the semicolon!
- Click the lightning bolt
 - Note the confirmation
- Reload

Demo: Creating a Table

```
USE animals_db;
```

```
CREATE TABLE people (  
  id INTEGER(11) AUTO_INCREMENT NOT NULL,  
  name VARCHAR(30) NOT NULL,  
  has_pet BOOLEAN NOT NULL,  
  pet_name VARCHAR(30),  
  pet_age INTEGER(10),  
  PRIMARY KEY (id)  
);
```

USE animals_db;

tells MySQL that all of the code that follows will be affecting the data stored within the animals_db database

```
CREATE TABLE people (<COLUMNS>);
```

creates a table called people with the columns listed
within the parentheses

id INTEGER(11) AUTO_INCREMENT NOT NULL

creates a column id which can hold numbers with precision up to 12 places. It will not allow null fields, and will auto-increment

name VARCHAR(30) NOT NULL

creates a column of name which can hold character strings within it of up to 30 characters and will not allow null fields

has_pet BOOLEAN NOT NULL

creates a column of has_pet which can hold either true or false within it and will not allow null fields

`pet_name VARCHAR(30)`

creates a column of `pet_name` which can hold character strings within it of up to 30 characters and will allow null fields

`pet_age INTEGER(10)`

creates a column of `pet_age` which can hold whole numbers with precision up to 11 places

PRIMARY KEY (id)

Finally, we want to use the id column as our PRIMARY KEY. The primary key is used to uniquely identify each record in a database table. You can only have one - in this case we are using an INTEGER and the primary key is the id column

Lightning!

Uh Oh!

- MySQL data is persistent and therefore is not deleted or overwritten when identical commands are run
- Deleting working code is not something you want to get into the habit of doing
- Highlight the lines of code you want to run
- Then hit lightning
- Reload



Activity: 02-FavoriteDB

Review: 02-FavoriteDB



id INTEGER(11) AUTO_INCREMENT NOT NULL

creates a column id which can hold numbers with precision up to 12 places. It will not allow null fields, and will auto-increment

`five_times BOOLEAN DEFAULT FALSE,`

DEFAULT is used to insert a default value into a column. In this case, it is used to set the default BOOLEAN value for the five_time column to FALSE

PRIMARY KEY (id)

Finally, we want to use the id column as our PRIMARY KEY. The primary key is used to uniquely identify each record in a database table. You can only have one - in this case we are using an INTEGER and the primary key is the id column

Is there any data in our database?



Meanwhile, Back at the Ranch...

- `USE animals_db;`
- `SELECT * FROM people;`
- Note the empty table

Watch

```
INSERT INTO people (name, has_pet, pet_name, pet_age)  
VALUES ("Ahmed", true, "Rockington", 100);
```

```
SELECT * FROM people;
```

Yup

- Inserts data into the "people" table
- Then specifies what columns that data will be entered into
- Values then places the data contained in the parentheses into the corresponding columns listed after the INSERT INTO statement

What?

```
INSERT INTO people (name, has_pet, pet_name, pet_age)  
VALUES ("Jacob", true, "Misty", 10);
```

Unlike tables and databases, there can be identical data stored within a table

What?

```
INSERT INTO people (name, has_pet)  
VALUES ("Peter", false);
```

null

What?

```
UPDATE people
```

```
SET has_pet = true
```

```
WHERE name = "Peter";
```


Trouble?

```
SET SQL_SAFE_UPDATES = 0;
```

This code updates the data stored in the columns specified in the SET statement in the row where the data in the "name" column is equal to "Peter"

If we did not include that WHERE statement, then our UPDATE would have set the values for all columns instead of those of a specific row. That would have been a pretty major error and is why the WHERE statement is so important.

GifsRC.tumblr.com



Activity: Add Data to FavoriteDB

The Value of Unique Values

```
DELETE FROM people  
WHERE name = "Ahmed";
```

Demo: 03-animals_dbWithID

- Note how the WHERE statement uses the id column to select the row we want to affect
 - A primary key uniquely identifies a row.
- Because it auto-increments, each row's ID is guaranteed to be unique. This ensures that we don't identify and update the wrong row when we execute CRUD statements.
- Our insert statements have not changed, as we do not need to insert data specifically into the id column. MySQL automatically provides a value for this column, fulfilling the uniqueness constraint by automatically incrementing the last value used as an ID.

Removing Duplicates

```
DELETE FROM people  
WHERE id = 2;
```


Activity: Making and Using an ID Column

See 04-programmingDB for instructions

How do we create a new database?

How do we switch into a new database?

How do we create a table?

How do we create a primary key?

What is the purpose of a primary key?

Review: programmingDB.sql



What if our database has more than one
table?

Join

Joins allow MySQL to combine two or more individual tables together using a value that is shared between them.

Demo: 05-books

- We can utilize joins to combine and display data from both tables.
- These tables have what's known as a one-to-many relationship, since a book can only have one author, but an author can have many books.
 - INNER JOIN: Combines tables where the specified data-values within a column match one-another
 - LEFT JOIN: Combines tables and shows all of the values of the first table specified while only the values which match on the second side will be shown
 - RIGHT JOIN: Combines tables and shows all of the values of the second table specified while only the values which match on the first table will be shown



What is a foreign key?

<http://imgtfy.com/?q=sql+foreign+key>

A FOREIGN KEY is a key used to link
two tables together