

Running Experiments Using jsPsych and MySQL

Ruth Shaffer

Memory & Cognition Lab Meeting

30 May 2017

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Links and Tips

RECOMMENDED



LINKS

MAMP 3.5.2 <https://www.mamp.info/en/downloads/older-versions/>

Current jsPsych <https://github.com/jodeleeuw/jsPsych/releases>

jsPsych tutorial 1 <http://docs.jspsych.org/tutorials/hello-world/>

jsPsych tutorial 2 <http://docs.jspsych.org/tutorials/rt-task/>

Web path to manual experiment http://localhost/jspsych_manual/experiment2.html

jsPsych data storage <http://docs.jspsych.org/features/data/>

MTurk <https://www.mturk.com/mturk/welcome>

MTurk individual subject https://requester.mturk.com/workers/<WORKER_ID>

Links and Tips

CODE SNIPPETS

- 1 /Applications/MAMP/Library/bin/mysqladmin -u root -p password
- 2 CREATE TABLE `manual_table2` (
 `rt` int(11) NOT NULL,
 `key_press` smallint(11) NOT NULL,
 `trial_type` varchar(200) NOT NULL,
 `trial_index` smallint(6) NOT NULL,
 `time_elapsed` int(11) NOT NULL,
 `internal_node_id` varchar(300) NOT NULL,
 `subject` int(11) NOT NULL,
 `conditions` varchar(200) NOT NULL,
 `stimulus` varchar(200) DEFAULT NULL,
 `response` varchar(100) DEFAULT NULL,
 `correct` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
- 3 SHOW CREATE TABLE manual_table2;

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Set up local web development environment

MAMP: MacOS, Apache, MySQL, PHP/Python/Perl

- Apache: server
- MySQL: database management
- PHP: language we will use to interface with MySQL

Or can set it all up manually

- PowerPoint vs. manual

Set up local web development environment

Necessary for local development and testing of your experiment

Final testing on Giraffe, but initial development and testing should not be public or on the testing server

Set up local web development environment

MAMP version 3.5.2

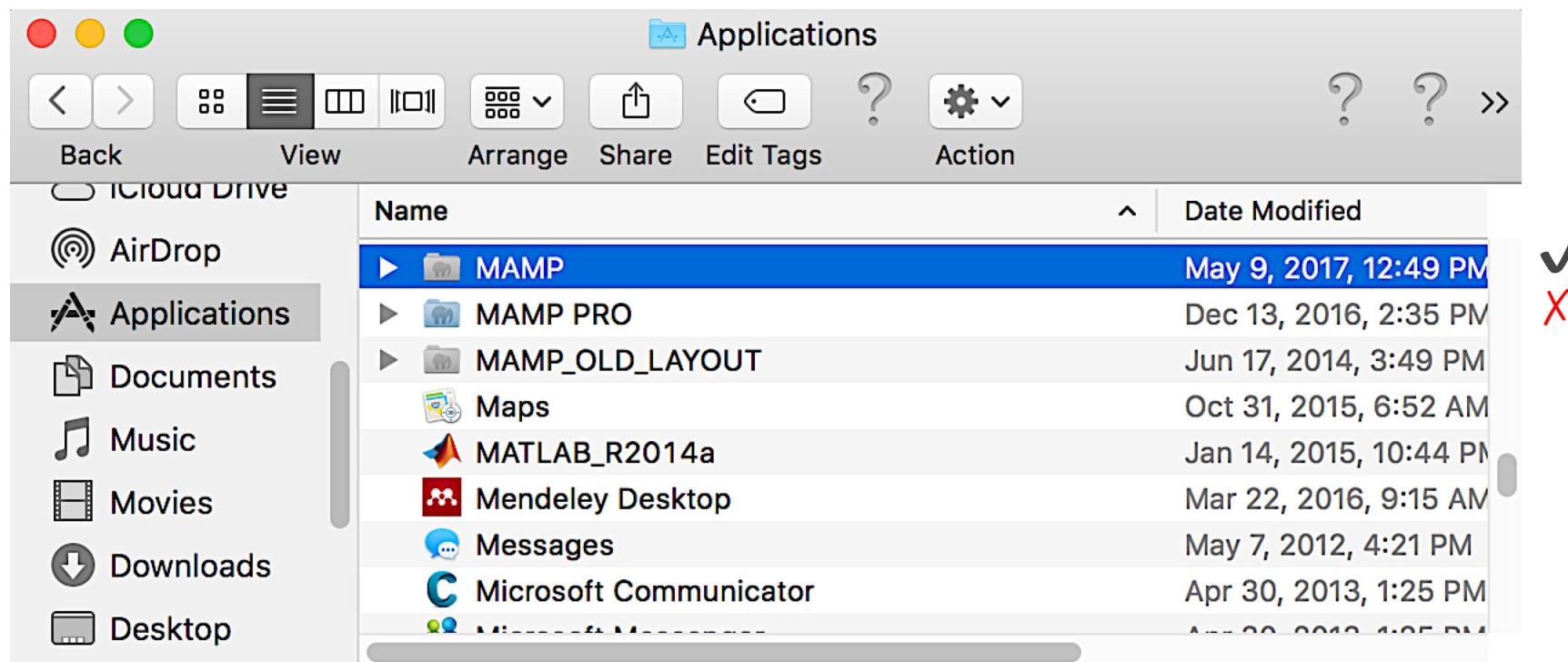
- not most recent
- how it works with jsPsych code for MySQL storage
- (we want earlier PHP version)

Download MAMP (not MAMP PRO) for Mac

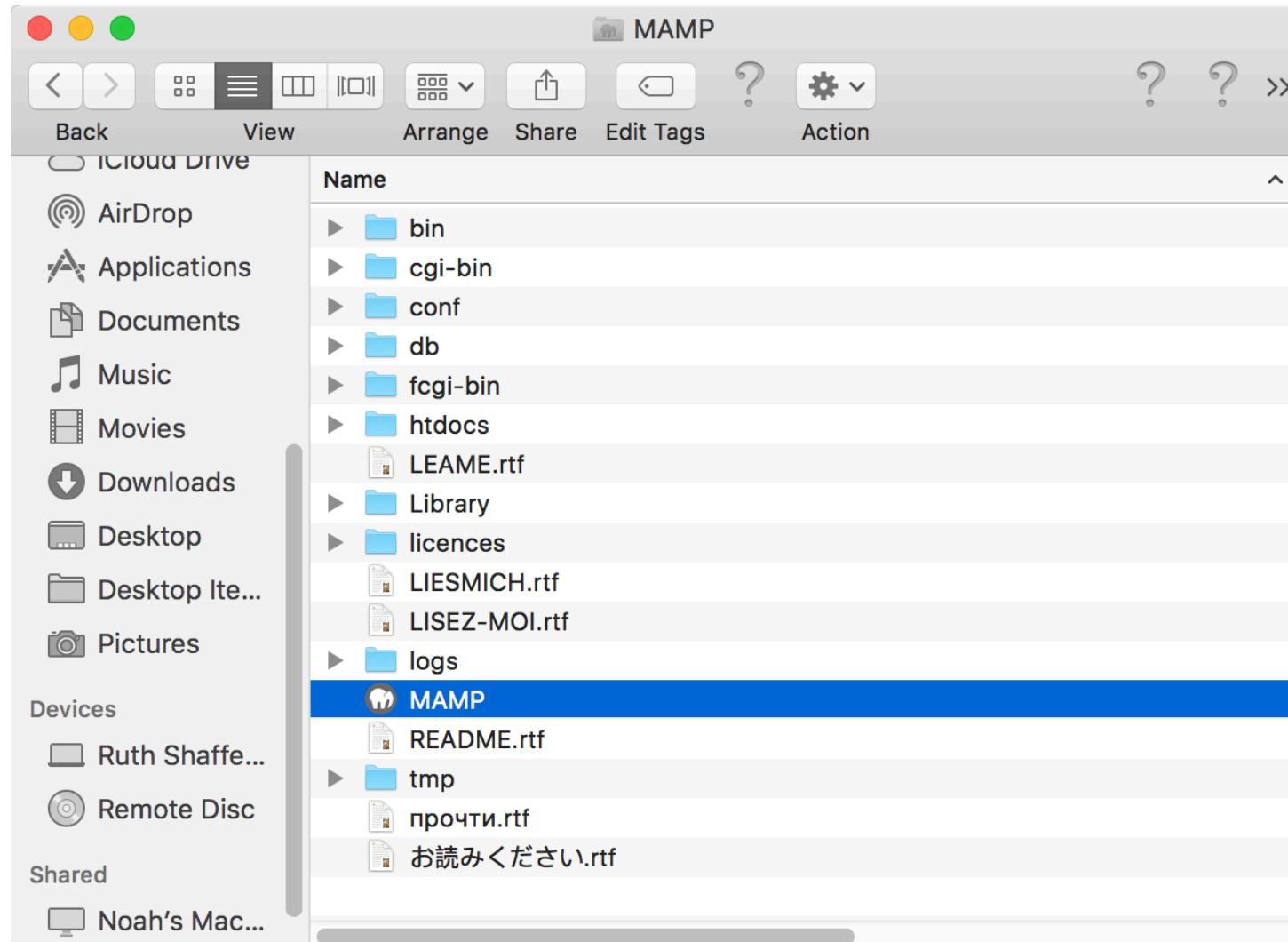
- *see MAMP download link*

Launch installer

Set up local web development environment

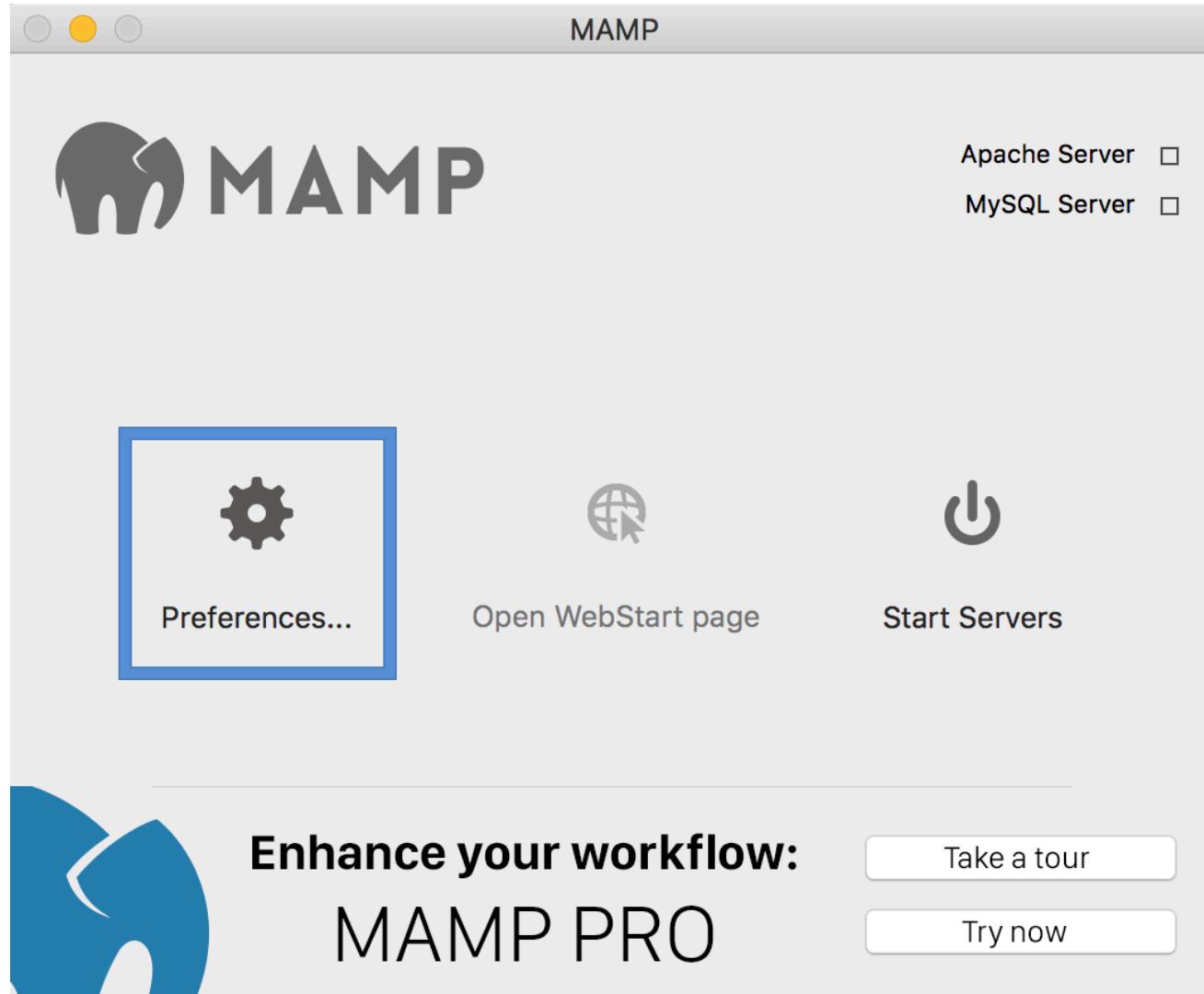


Set up local web development environment

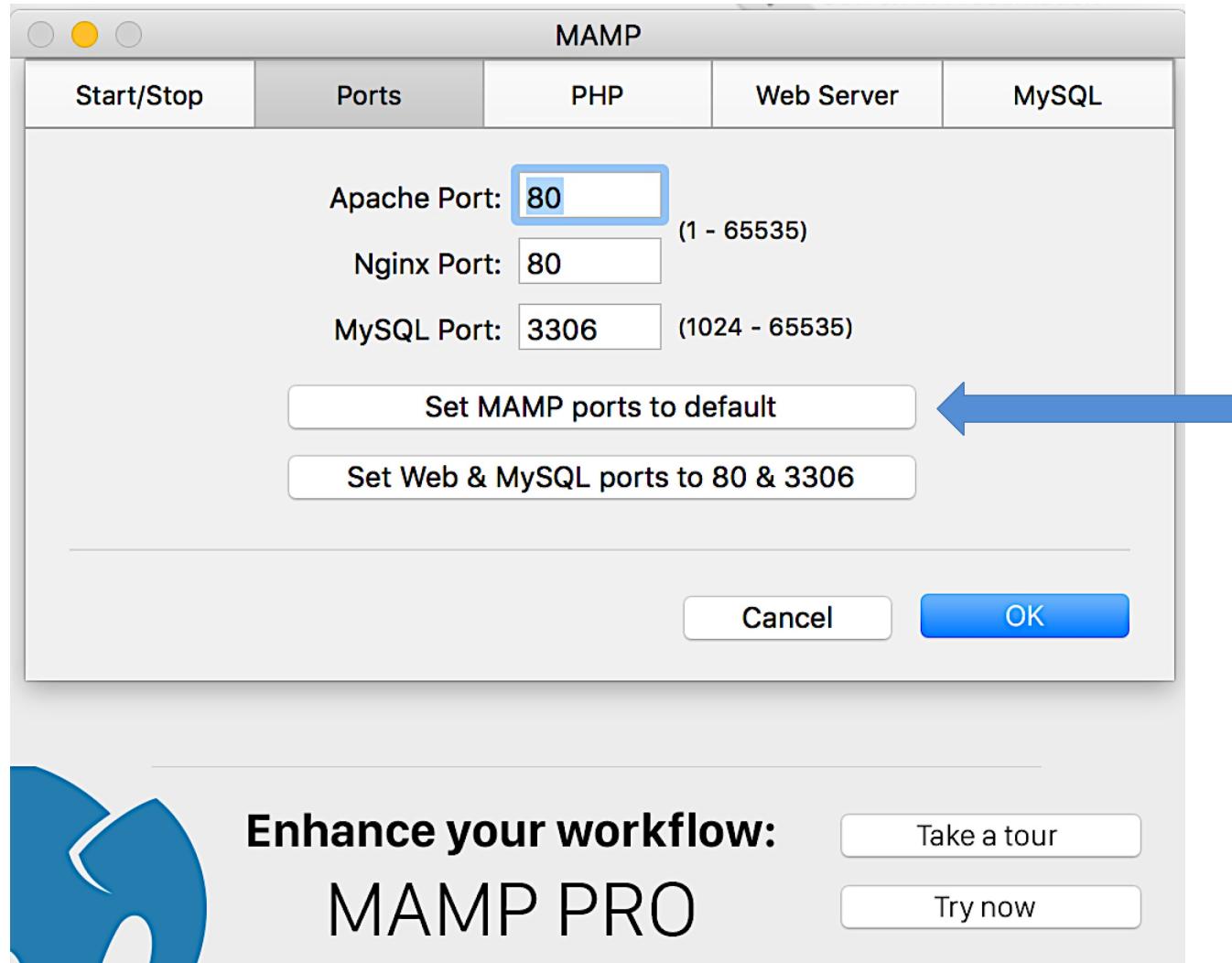


In a popup, you may have to select Launch MAMP (not launch MAMP PRO)

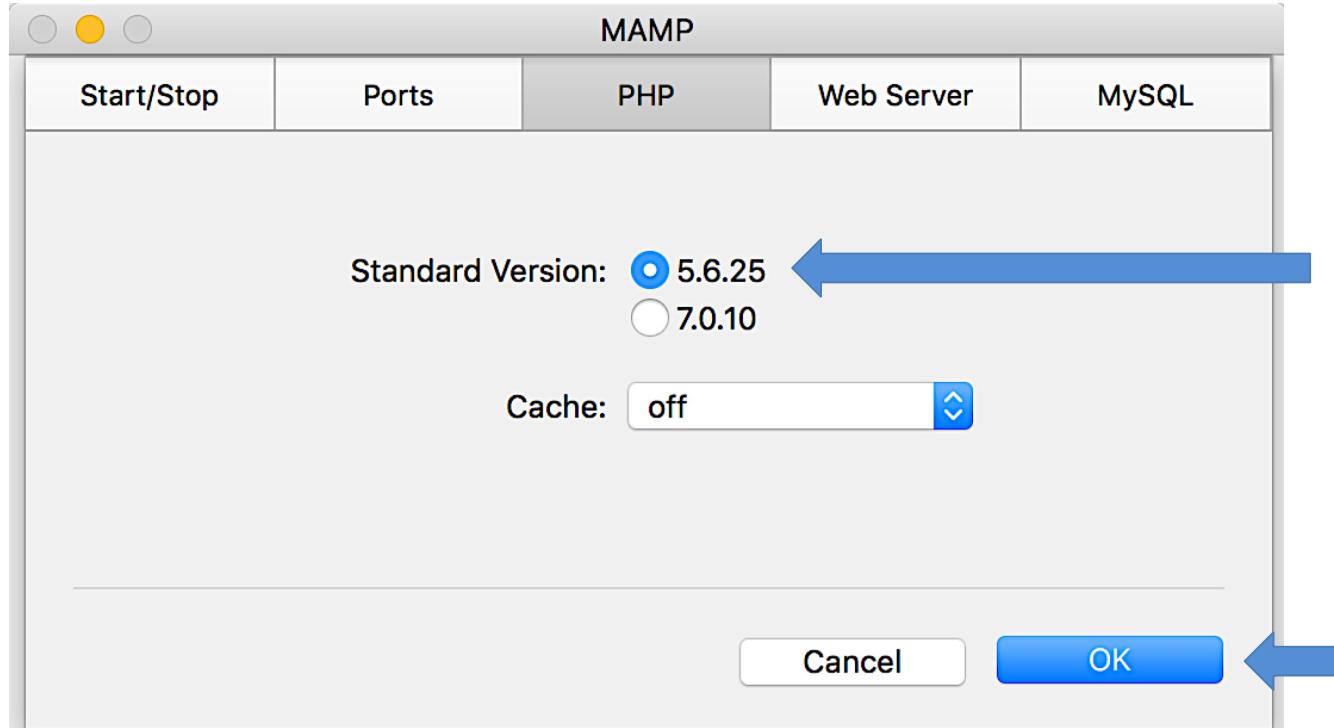
Set up local web development environment



Set up local web development environment



Set up local web development environment

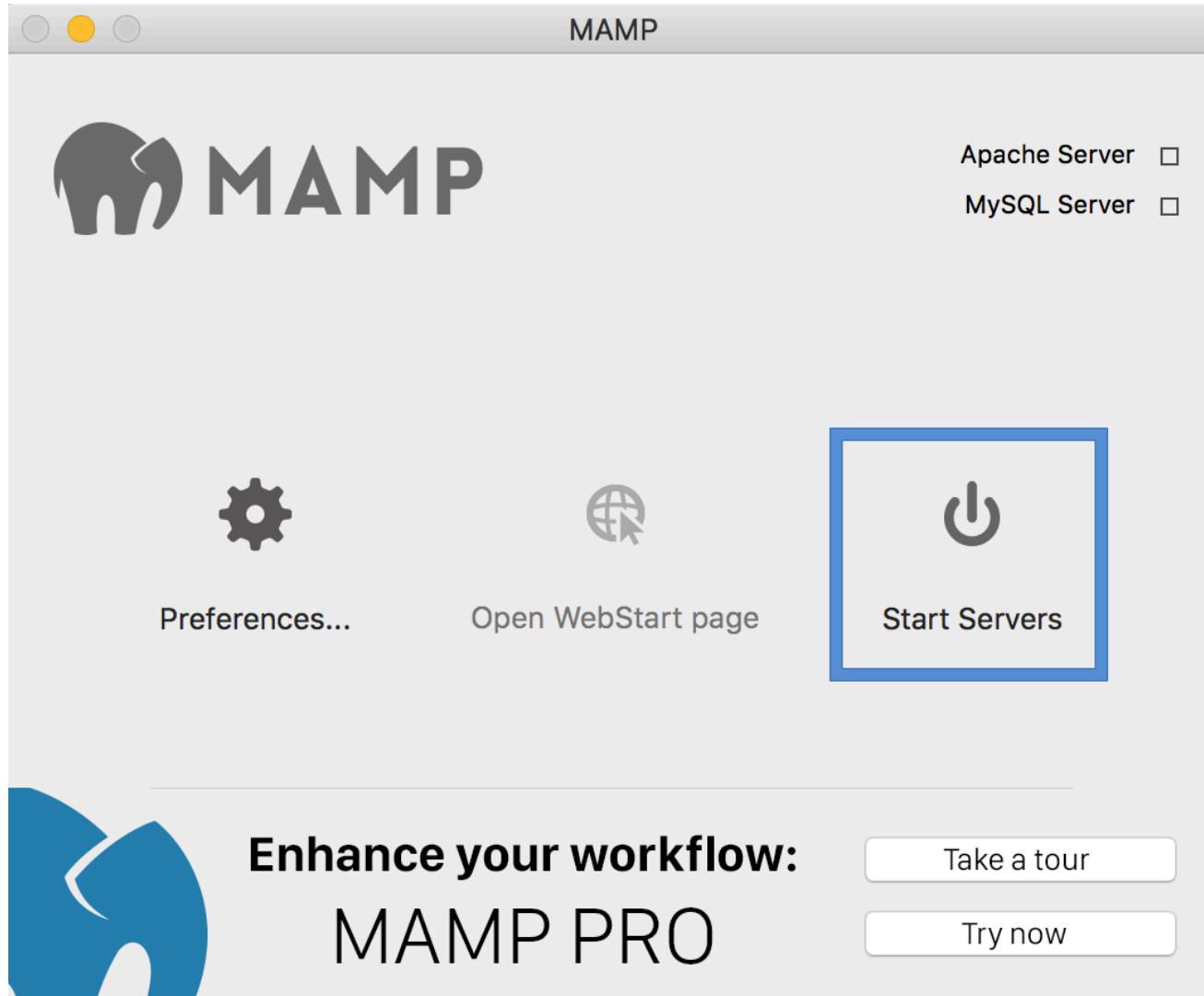


Enhance your workflow:
MAMP PRO

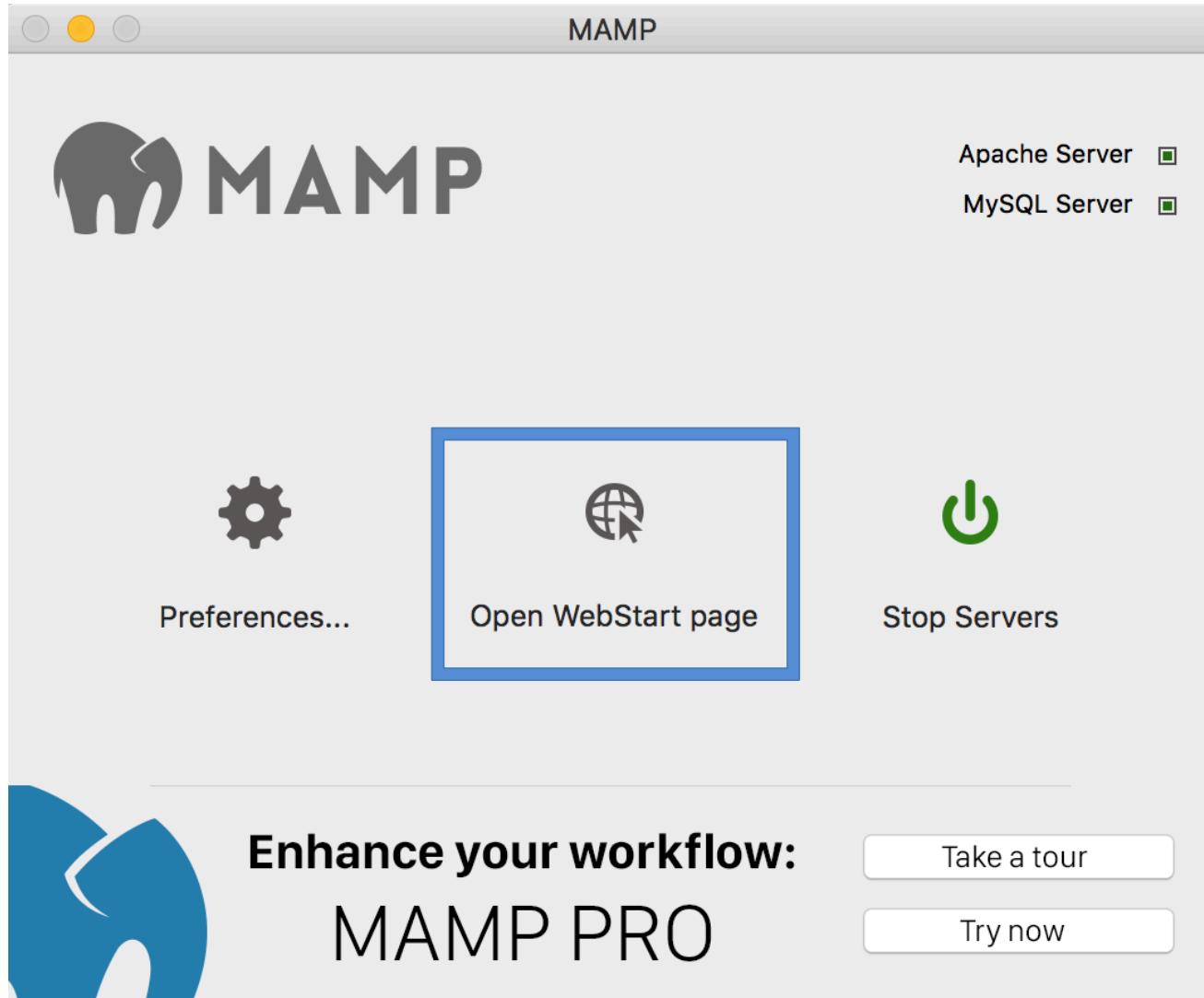
[Take a tour](#)

[Try now](#)

Set up local web development environment



Set up local web development environment



Set up local web development environment

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost/MAMP/?language=English
- Toolbar:** Includes standard browser controls (Back, Forward, Stop, Refresh) and a bookmarks bar placeholder.
- Header:** MAMP website navigation menu with links: Start, My Website, phpInfo, Tools, FAQ, MAMP Website, and a prominent orange "Buy MAMP PRO" button.
- Main Content Area:**
 - A large green section on the left features the text "Now available: MAMP & MAMP PRO 4" and an orange "Download now!" button.
 - A large white number "4" is displayed prominently.
 - A blue section on the right features the MAMP logo (a white elephant silhouette).
- Bottom Navigation:** A blue bar with the text "Host" and "localhost".
- Left Sidebar:** A box titled "PHP" with the subtext "phpinfo shows the current configuration of PHP".
- Right Sidebar:** A box titled "MAMP Version" with the subtext "4.0.5 → Update (4.1.1) available!".
- Bottom Footer:** A box titled "News" with the subtext "MAMP & MAMP PRO 4.1.1 out now" and "February, 2017 – Version 4.1.1 is out now - comes with German localization - Added german localisation".

Set up local web development environment

PHP

[phpinfo](#) shows the current configuration of PHP.



Helpful configuration summary

MySQL

MySQL can be administered with [phpMyAdmin](#).



Links to GUI to deal with MySQL database and tables.
*Click to make sure
phpMyAdmin is working*

Set up local web development environment

The screenshot shows a web browser window for the MAMP application. The address bar displays "localhost" and "MAMP". The navigation bar includes links for "Start", "My Website", "phpInfo", "Tools", "FAQ", "MAMP Website", and a prominent orange "Buy MAMP PRO" button.

The main content area is the phpMyAdmin interface, specifically the "Server: localhost:3306" configuration screen. It features several tabs at the top: Databases, SQL, Status, User accounts, Export, Import, Settings, Replication, and More. The "Settings" tab is active.

The "General settings" section contains the "Server connection collation" dropdown set to "utf8mb4_unicode_ci". The "Appearance settings" section includes "Language" (English), "Theme" (Original), and "Font size" (82%). A "More settings" link is also present.

The right side of the interface is divided into several sections:

- Database server**: Lists the MySQL server details: Localhost via UNIX socket, MySQL, 5.6.28 - MySQL Community Server (GPL), Protocol version: 10, User: root@localhost, and Server charset: UTF-8 Unicode (utf8).
- Web server**: Lists the Apache and PHP configurations: Apache/2.2.31 (Unix) mod_wsgi/3.5 Python/2.7.12 PHP/5.6.25 mod_ssl/2.2.31 OpenSSL/1.0.2h DAV/2 mod_fastcgi/2.4.6 mod_perl/2.0.9 Perl/v5.24.0, Database client version: libmysql - mysqlnd 5.0.11-dev - 20120503 - \$Id: 76b08b24596e12d4553bd41fc93cccd5bac2fe7a \$, PHP extension: mysqli curl mbstring, and PHP version: 5.6.25.
- phpMyAdmin**: Lists the software version information: Version information: 4.6.4, and links to Documentation, Official Homepage, Contribute, Get support, and List of changes.

A sidebar on the left lists recent databases: CollectiveMemoryWWII, FR_experiment_1, information_schema, MTurk_CR, mysql, performance_schema, SART_db, and Tests4Study. A "New" button is also available.

Switch to Chrome if not working properly. Once you see it's working, you can close out.

Set up local web development environment

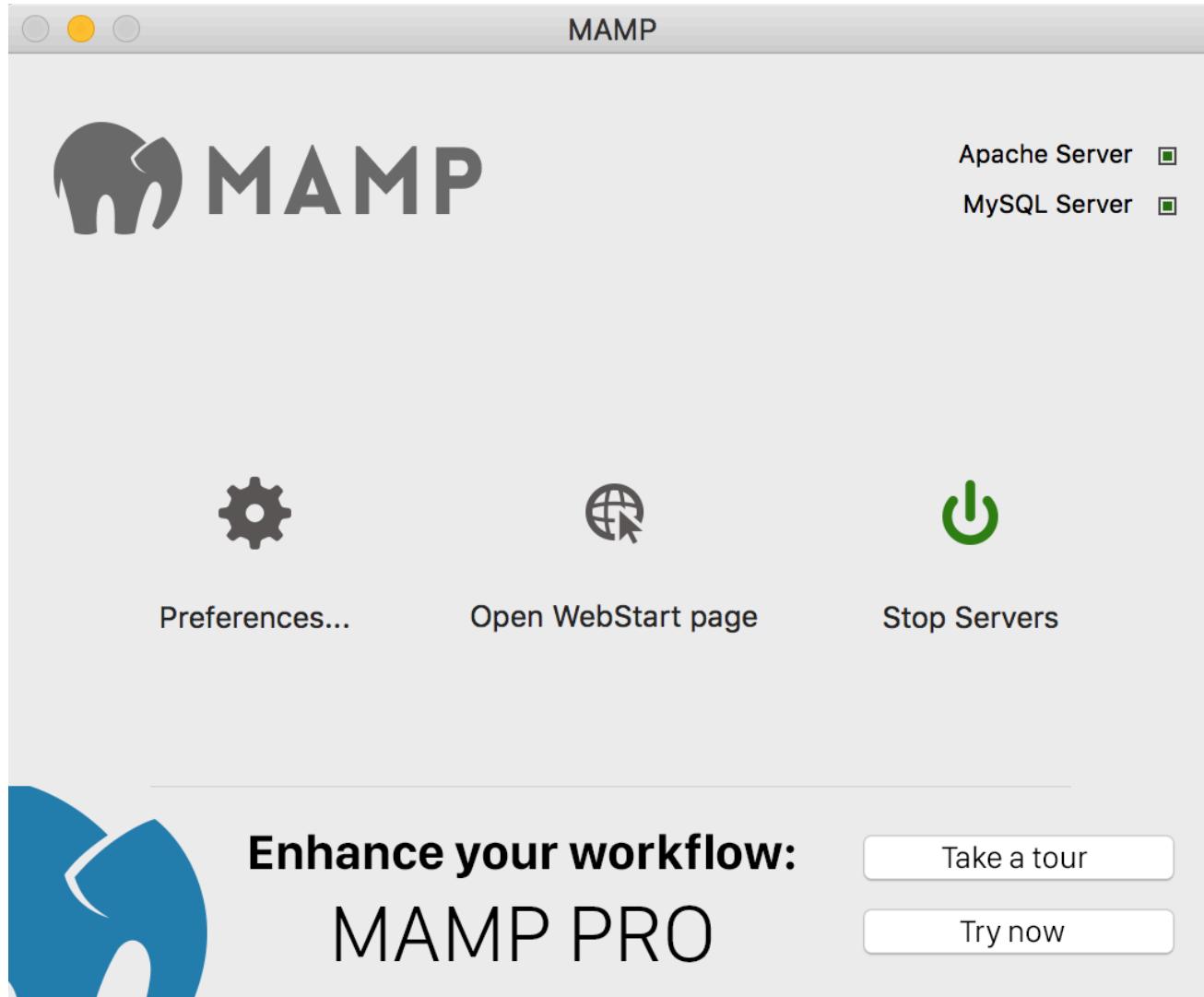
Current username/password for phpMyAdmin account:

- root/root

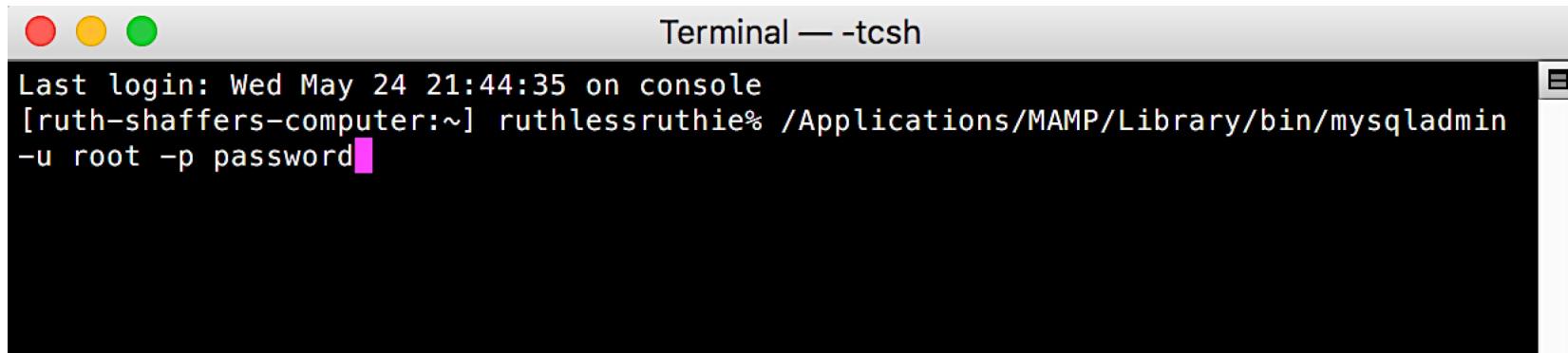
Let's change that to:

- root/<passwordOfYourChoosing>

Set up local web development environment



Set up local web development environment



```
Last login: Wed May 24 21:44:35 on console
[ruth-shaffers-computer:~] ruthlessruthie% /Applications/MAMP/Library/bin/mysqladmin
-u root -p password
```

For copying: /Applications/MAMP/Library/bin/mysqladmin -u root -p password

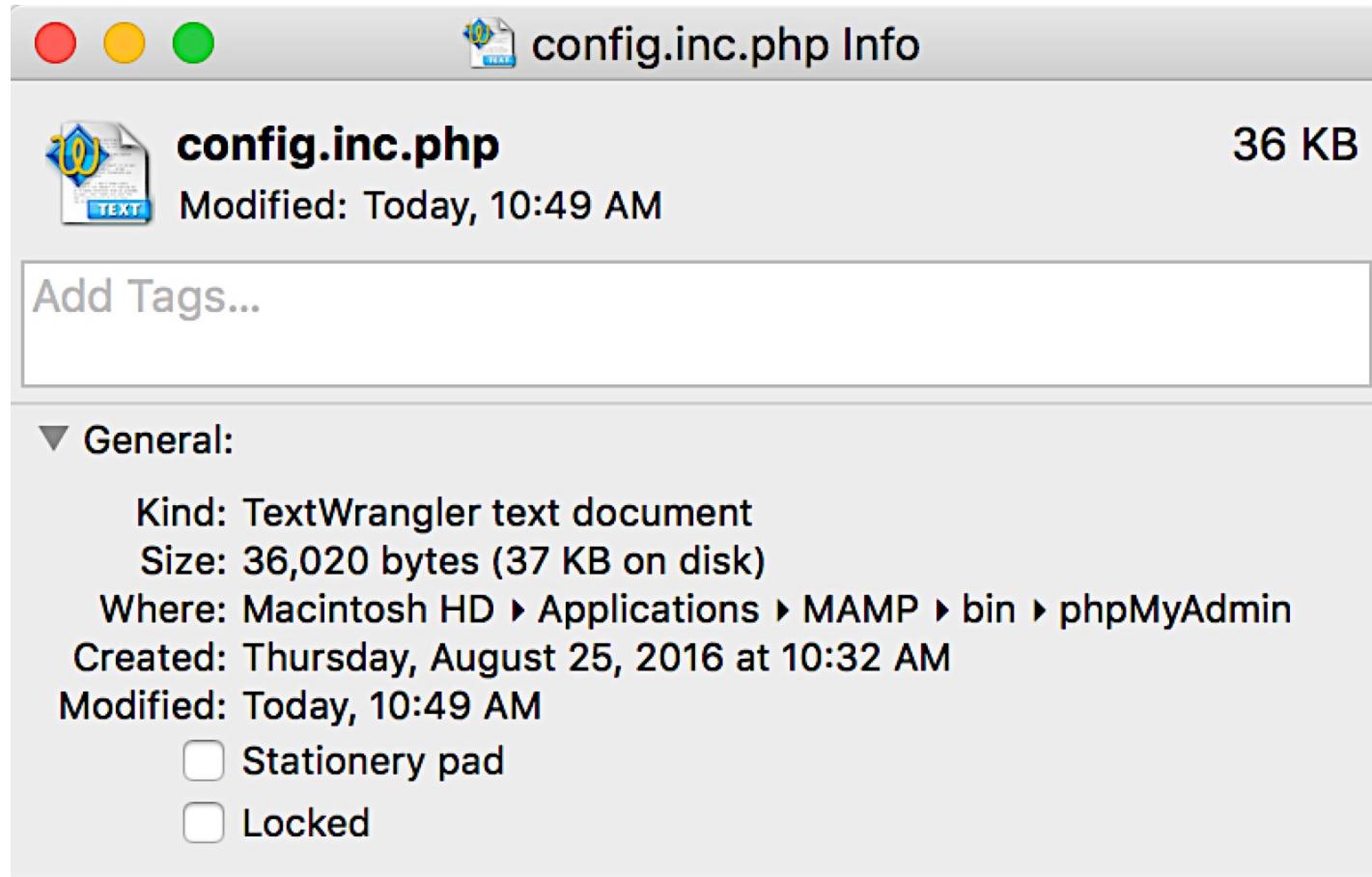
When prompted for each:

Enter current password (root)

New password (you'll need this again soon)

And again new password to verify

Set up local web development environment



Set up local web development environment

```
84 // features (pmadb)
85 $cfg['Servers'][$i]['auth_type'] = 'config'; // Authentication method (config, http or cookie based)?
86 $cfg['Servers'][$i]['user'] = 'root'; // MySQL user
87 $cfg['Servers'][$i]['password'] = 'passwordOfYourChoosing'; // MySQL password (only needed
88 // with 'config' auth_type)
```

Search “root” and this will come up
Replace password root with <passwordOfYourChoosing>

Set up local web development environment

To verify that this is all working, as before:

1. Start MAMP server (or Stop and Start it)
2. Open WebStart Page
3. Select phpMyAdmin link and make sure it opens
(no permission issues)

Outline

1. Links & starting tips
2. Set up a local web development environment
3. **Get going with jsPsych basics**
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Get going with jsPsych basics

jsPsych provides great documentation

- 2 great tutorials
- **tons** of helpful explanations of plugins / examples

For today, we'll just set up and quickly walk through their tutorial code

- *see jsPsych Tutorial 1 link*

Get going with jsPsych basics

Per jsPsych's Tutorial 1 (Hello World Tutorial)

Download jsPsych library via link they provide in Step 1

- or [*see jsPsych download link*](#)

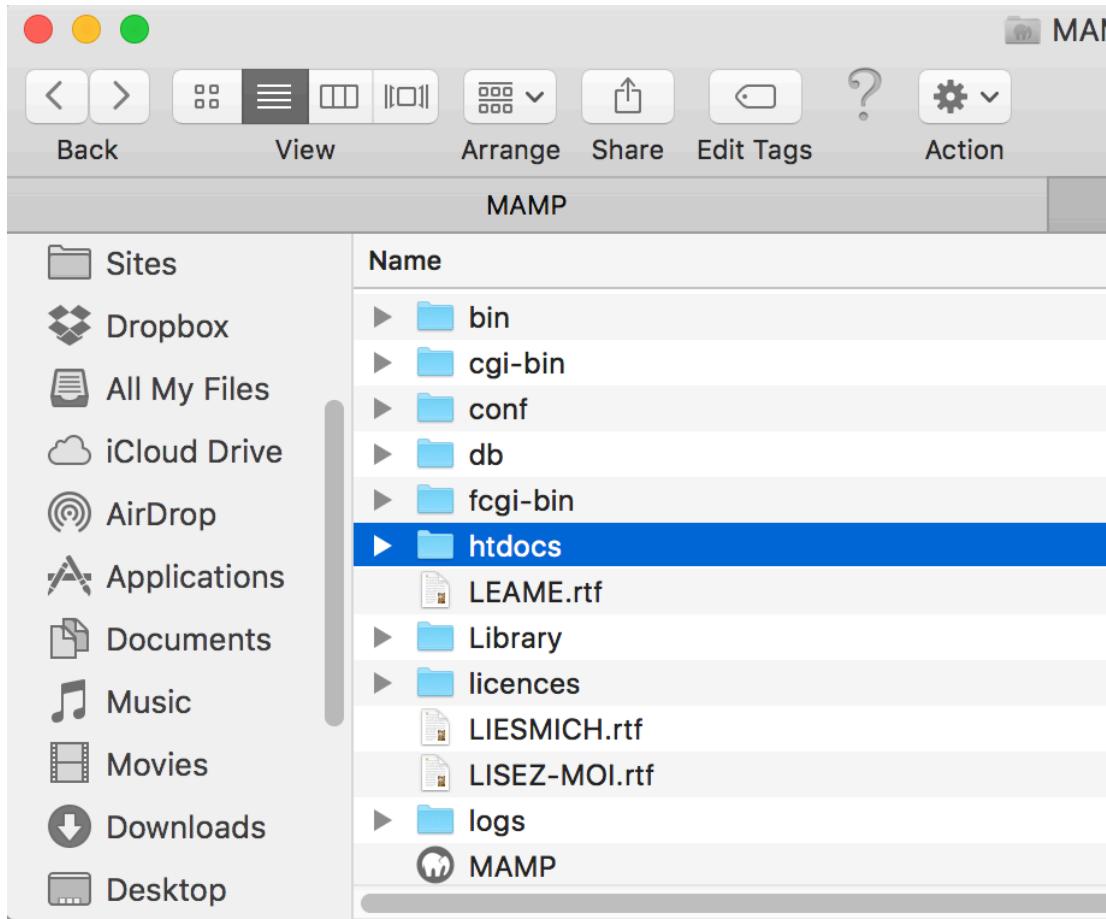
Unzip the folder

- brief walk-through of contents

Step 2 says to make a folder for the jsPsych library on your computer

- We will use MAMP's htdocs (serving) folder

Get going with jsPsych basics



So create a folder called “jspsych_manual” in the htdocs folder
Then, place the jspsych library within the “jspsych_manual” folder

Get going with jsPsych basics

For time purposes, we'll skip the rest of Tutorial 1 and go to the end of Tutorial 2:

- *see jsPsych Tutorial 2 link*

experiment2.html in manual folder

It's a copy of the code at the bottom of the tutorial page under “The final code”

Save file in the same level as the jsPsych library

Path: /Applications/MAMP/htdocs/jspysch_manual/experiment2.html

Get going with jsPsych basics

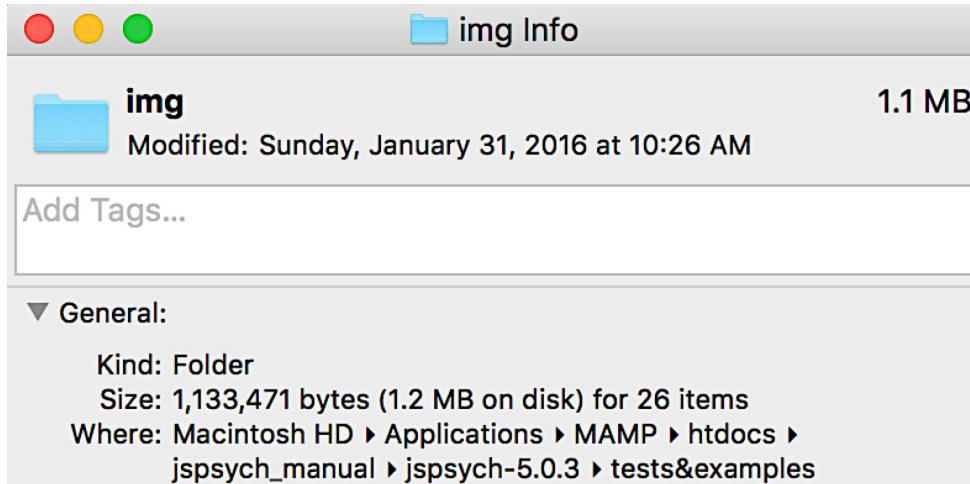
Brief walk through of code...

Changes to make:

1. src paths to correct jsPsych folder

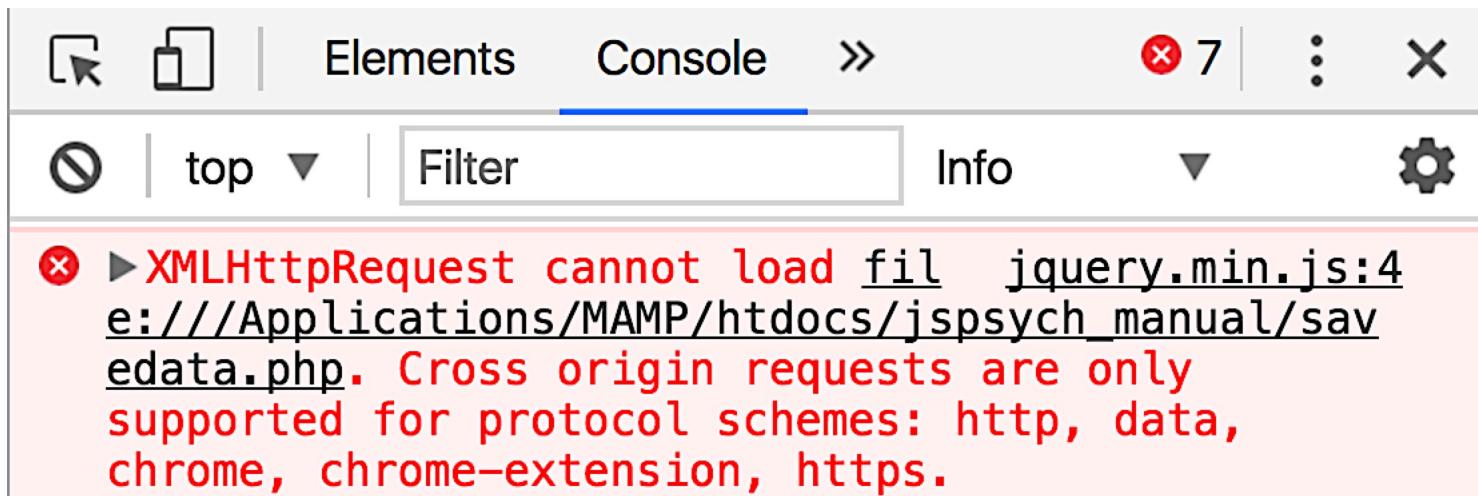
```
49 |  
50 |     var all_trials = jsPsych.randomization.repeat(test_stimuli, 2);  
51 |
```

2. copy 'img' folder to same level as experiment2.html



Get going with jsPsych basics

To run the experiment, do not click html file in folder.
Appears to work, but when it comes time to store outputs:



The path is incorrect and will look like this:

file:///Applications/MAMP/htdocs/jspysch_manual/experiment2.html

Get going with jsPsych basics

Instead, in Chrome, go to:

http://localhost/jspysch_manual/experiment2.html

- or localhost:8888/jspysch_manual/etc...
- whatever appears as the pre- “MAMP” portion of the web address when you click the MAMP Open WebStart page button

Run through the code until the end

Notice final data structure output

See next slide

Get going with jsPsych basics

```
[  
 {  
   "rt": 548,  
   "key_press": 32,  
   "trial_type": "text",  
   "trial_index": 0,  
   "time_elapsed": 549,  
   "internal_node_id": "0.0-0.0"  
 },  
 {  
   "rt": 71,  
   "key_press": 32,  
   "trial_type": "text",  
   "trial_index": 1,  
   "time_elapsed": 1624,  
   "internal_node_id": "0.0-1.0"  
 },  
 {  
   "rt": 305,  
   "stimulus": "img/blue.png",  
   "key_press": 70,  
   "response": "go",  
   "trial_type": "single-stim",  
   "trial_index": 2,  
   "time_elapsed": 3932,  
   "internal_node_id": "0.0-2.0-0.0",  
   "correct": true  
 },  
 {  
   "rt": 1054,  
   "stimulus": "img/orange.png",  
   "key_press": 70,  
   "response": "no-go",  
   "trial_type": "single-stim",  
   "trial_index": 3,  
   "time_elapsed": 6121
```

Each is a trial, and we want to store these in a way we can access later.

Get going with jsPsych basics

To store these trial outputs:

- see jsPsych data storage link

[experiment2_withExtraData.html](#) in manual folder

- experiment2.html +
 - changed src to correct jsPsych +
 - change repeat from 10 → 2 +
 - copy of the code in jsPsych's "Adding data to all trials" +
- place file in the directory with your experiment2.html file

Run your this file to see that it worked

- should see added data in output at the end

Get going with jsPsych basics

For the section: “Adding data to a particular trial or set of trials”

- Notice that the code already does this in the test_stimuli variable
- It defines “response” as either “go” or “no-go”

We can skip “Storing data permanently as a file”

But before completing “Storing data permanently in a MySQL database” we need to set up a database on our end

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. **Set up MySQL database and table**
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Set up MySQL database and tables

The screenshot shows the MAMP web interface running in a browser. The title bar says "MAMP". The address bar shows "localhost/MAMP/?language=English". The top navigation bar includes "Start", "My Website", "phpInfo", "Tools", "FAQ", "MAMP Website", and a "Buy MAMP PRO" button.

The main content area features a large green section on the left with the text "Now available: MAMP & MAMP PRO 4" and a "Download now!" button. To its right is a large white number "4". The right side of the page is blue with the MAMP logo (a stylized elephant).

Below the main content are several sections:

- PHP**: "phpinfo shows the current configuration of PHP."
- MySQL**: "MySQL can be administered with [phpMyAdmin](#). To connect to the MySQL server from your own scripts use the following connection parameters:

Host	localhost
------	-----------

"
- MAMP Version**: "4.0.5 → [Update \(4.1.1\) available!](#)
- News**: "MAMP & MAMP PRO 4.1.1 out now
February, 2017 – Version 4.1.1 is out now - comes with German localization
- Added german localisation"

Set up MySQL database and tables

The screenshot shows the MAMP web interface with the URL `localhost` in the address bar. The main navigation bar includes links for `Start`, `My Website`, `phpInfo`, `Tools`, `FAQ`, `MAMP Website`, and a prominent orange `Buy MAMP PRO` button.

The central content area displays the `phpMyAdmin` configuration interface. On the left, a sidebar lists existing databases: `CollectiveMemoryWWII`, `FR_experiment_1`, `information_schema`, `MTurk_CR`, `mysql`, `performance_schema`, `SART_db`, and `Tests4Study`. A red box highlights the `New` button.

The main configuration area contains several sections:

- General settings**: Server connection collation set to `utf8mb4_unicode_ci`.
- Appearance settings**: Language set to `English`, Theme set to `Original`, and Font size set to `82%`.
- Database server**: Lists the server configuration, including Localhost via UNIX socket, MySQL 5.6.28, and UTF-8 Unicode (utf8).
- Web server**: Lists the Apache and PHP configurations, including mod_wsgi, mod_ssl, and MySQL extensions.
- phpMyAdmin**: Lists version information (4.6.4), documentation links, and support options.

At the bottom left, there is a `Console` link.

Set up MySQL database and tables

The screenshot shows the 'Databases' section of the phpMyAdmin interface. A new database named 'manual_database' is being created. The 'Collation' dropdown is set to 'utf8_general_ci'. The 'Create' button is visible.

1

2

3

Database	Collation	Action
manual_database	utf8_general_ci	Create

Set up MySQL database and tables

1

2

 **Create table**

Name: Number of columns:

3

Set up MySQL database and tables

Specify information for the column to hold:

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected. A large number '1' is positioned above the table definition area. The 'Table name:' field contains 'manual_table1'. Below it, an 'Add' button with the value '1' and a 'column(s)' dropdown are shown. The 'Structure' section displays a single column named 'rt' of type 'INT'. The 'Collation' dropdown is set to 'None'. The 'Storage Engine' is set to 'InnoDB'. The 'PARTITION definition:' section is present but empty. At the bottom right, there are 'Preview SQL' and 'Save' buttons.

1

Name	Type	Length/Values	Default	Collation	Attributes	Nul
rt	INT		None			

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Preview SQL Save

2

For now, all else can be left as is

Set up MySQL database and tables

So now we have a database and a table with a single column in it

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. **Hook jsPsych up with MySQL database & table**
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Hook jsPsych up with MySQL database & table

Back to jsPsych's data storage page:

- see [jsPsych data storage link](#)
- in jsPsych link see “Storing data permanently in a MySQL database”

jsPsych provides code for 2 files necessary to connect to MySQL, I've sent them in the folder

- [savedata.php](#)
- [database_connect.php](#)

Hook jsPsych up with MySQL database & table

Place these in the experiment folder (same level as experiment2.html)

- will need these 2 files to be accessible to any experiment
- can always link to the same 2 files, or simply place copies of these in any experiment folder you make (this is what I do because I use separate folders for every experiment)

Hook jsPsych up with MySQL database & table

As instructed by jsPsych

See [database_connect.php](#) and change:

- username → root
- password → passwordOfYourChoosing
- database name → manual_database

Hook jsPsych up with MySQL database & table

[experiment2WithDataStorageEveryTrial.html](#) in manual folder

= experiment2.html +

changed src to correct jsPsych +

change repeat from 10 → 2 +

copy of the code in jsPsych's "Adding data to all trials" +

jsPsych function to save data in manual_table1 +

commented out opt_data: {key: value} and preceding comma +

commented out final jsPsych.data.displayData() +

code to call save_data function at the end of each trial:

```
on_finish:  function(){
    var current_node_id = jsPsych.currentTimelineNodeID();
    var currentTrialData = jsPsych.data.getDataByTimelineNode(current_node_id);
    save_data(currentTrialData);
}
```

Place in the directory with your experiment2.html file

Hook jsPsych up with MySQL database & table

The save data function calls the savedata.php file, which uses database info from the database_connect.php file

However, it needs to know what table within the database to deal with, so as instructed in the jsPsych function,
change the table to reference your table:

```
154  ...
155      function save_data(data){
156          var data_table = "manual_table1"; // change this for different experiments
157          $.ajax({
```

Hook jsPsych up with MySQL database & table

Next, rerun your code.

It should work fine, but if you check your MySQL table it will be empty (no data logged) ☹

The screenshot shows a MySQL database interface with the following elements:

- Top navigation bar with tabs: Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Triggers.
- Message bar: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)"
- SQL query input field: "SELECT * FROM `manual_table1`"
- Bottom toolbar: Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Hook jsPsych up with MySQL database & table

At this point, it is necessary to understand the MySQL table structure and how objects (each a trial of your experiment) are inserted into tables.

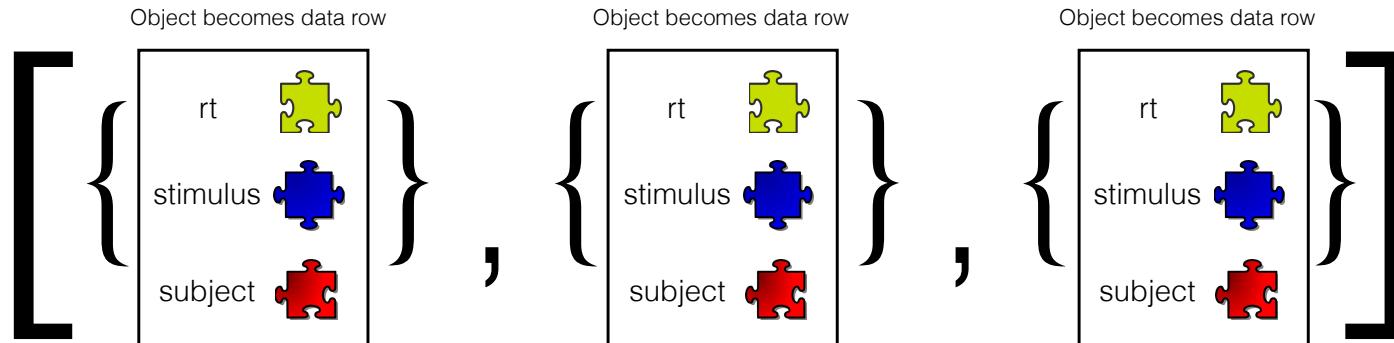
I made a few schematics on the following 4 slides that I hope will be clarifying.

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. **Understand MySQL table and data insertions**
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

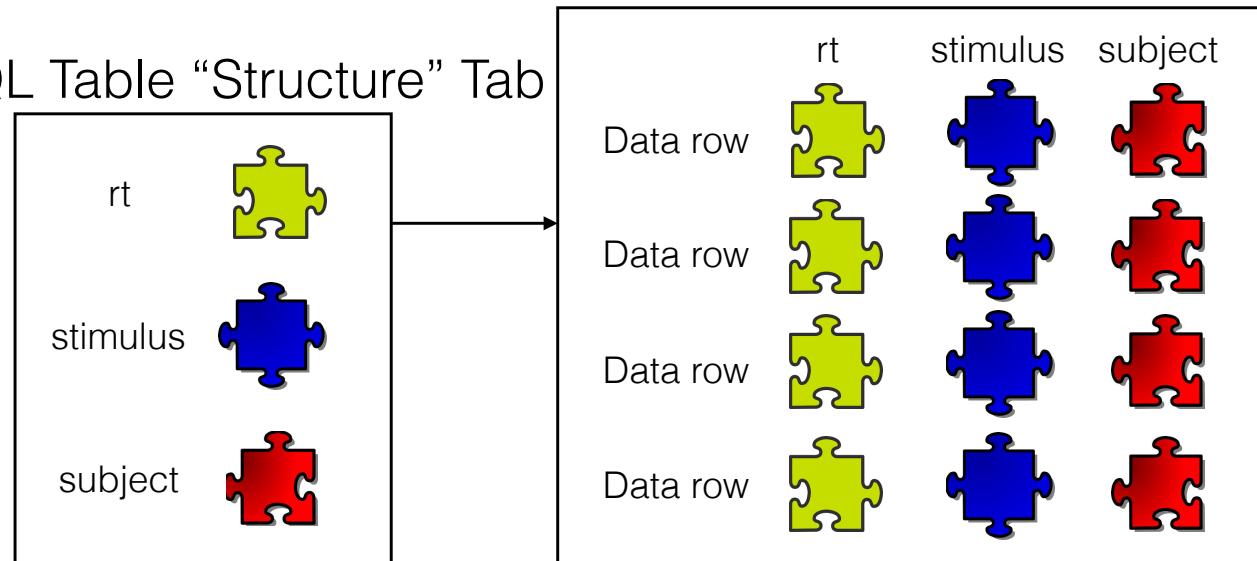
Understand MySQL table and data insertions

Array of objects to Insert into MySQL Table



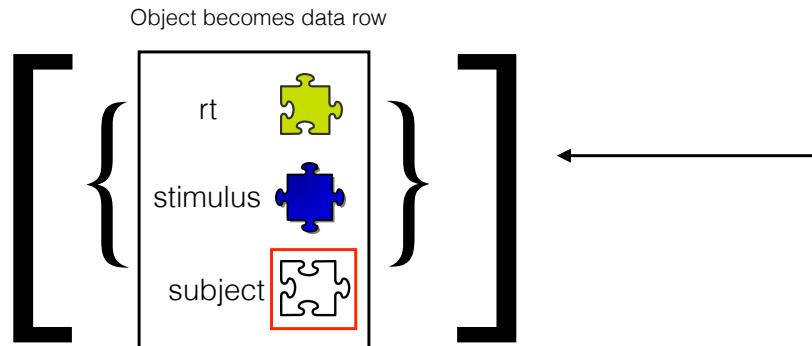
MySQL Table “Browse” Tab

MySQL Table “Structure” Tab



Understand MySQL table and data insertions

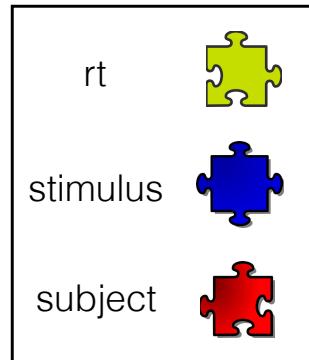
Object to Insert into MySQL Table



This object will not be inserted into the table.

Why: The subject data sent does not correspond to the subject data the table is expecting.

MySQL Table “Structure” Tab

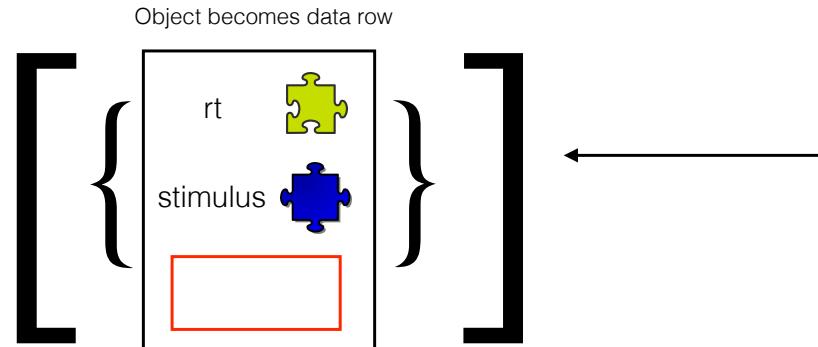


MySQL Table “Browse” Tab

	rt	stimulus	subject
Data row 1	[green]	[blue]	[red]
Data row 2	[green]	[blue]	[red]
Data row 3	[green]	[blue]	[red]
Data row 4	[green]	[blue]	[red]

Understand MySQL table and data insertions

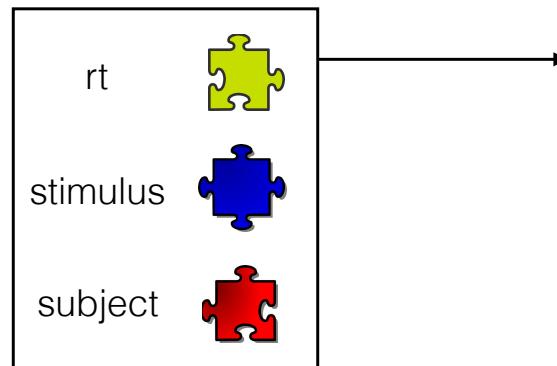
Object to Insert into MySQL Table



This object will not be inserted into the table.

Why: The table is expecting an rt, stimulus, and subject. No subject was included

MySQL Table “Structure” Tab

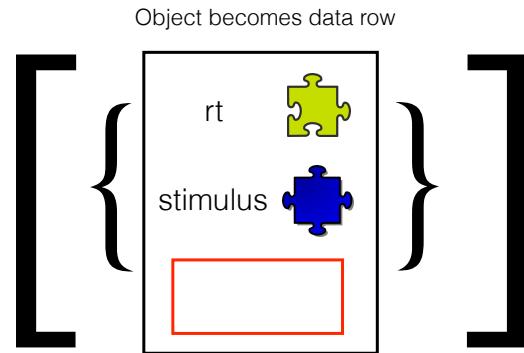


MySQL Table “Browse” Tab

	rt	stimulus	subject
Data row	Yellow puzzle piece	Blue puzzle piece	Red puzzle piece
Data row	Yellow puzzle piece	Blue puzzle piece	Red puzzle piece
Data row	Yellow puzzle piece	Blue puzzle piece	Red puzzle piece
Data row	Yellow puzzle piece	Blue puzzle piece	Red puzzle piece

Understand MySQL table and data insertions

Object to Insert into MySQL Table



So what do you do if some of your trials (objects) don't include a subject number?

Answer: Tell the subject column in your MySQL table that it's ok to be given a null result.

MySQL Table “Structure” Tab

rt	
stimulus	
subject	Default: NULL

MySQL Table “Browse” Tab

	rt	stimulus	subject
Data row			

Understand MySQL table and data insertions

Hopefully now it's clear why your MySQL table is completely empty when you click the “Browse” button to see your data.

None of the objects (trials) submitted exactly matched your table specifications

Understand MySQL table and data insertions

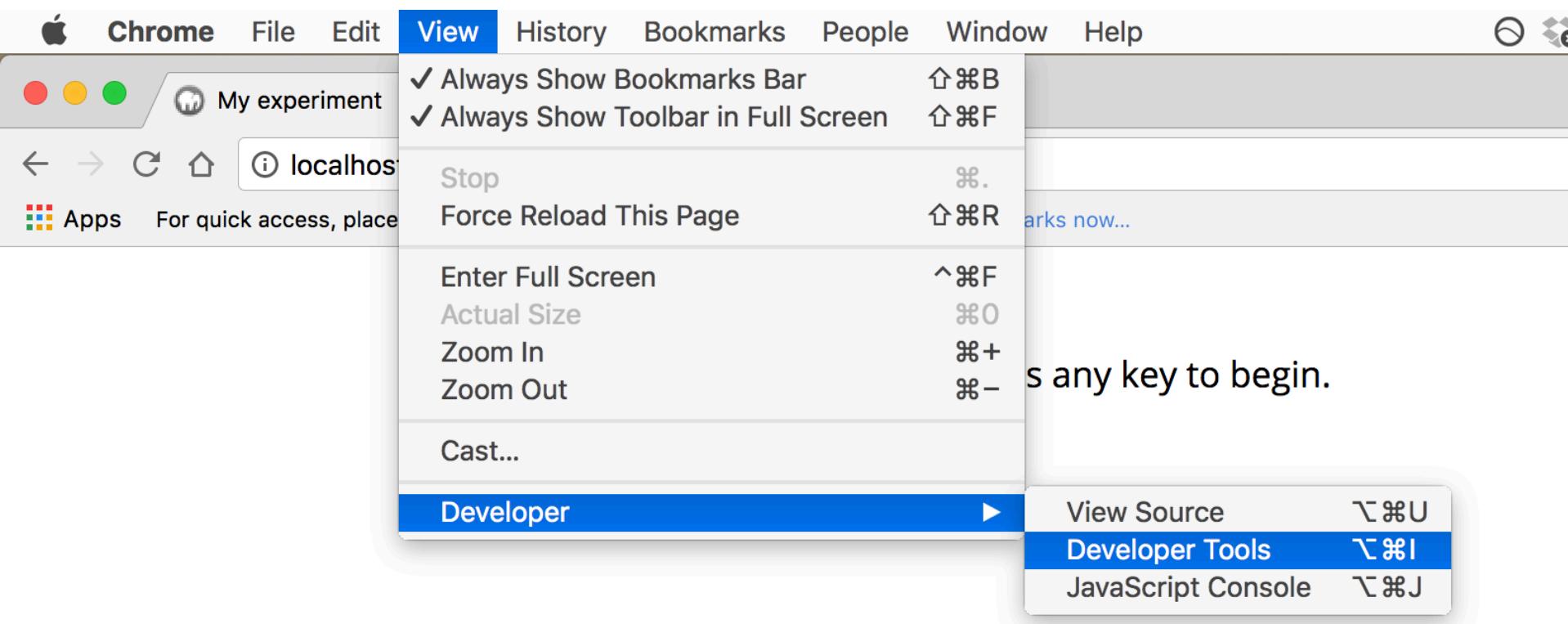
But how can you keep track of exactly what data is being sent after each trial, in order to create an appropriate table?

Outline

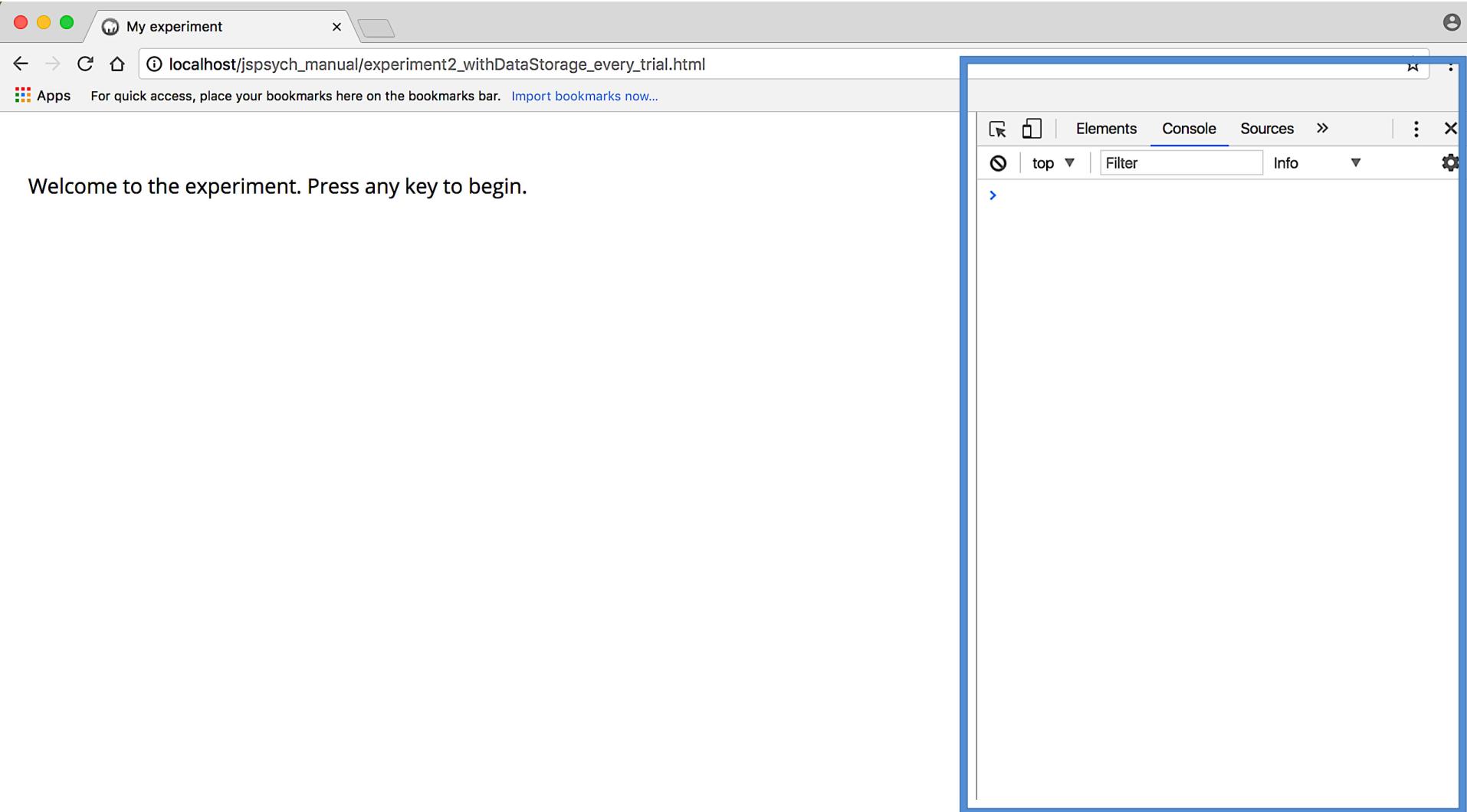
1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. **Use developer tools**
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

Use developer tools

Open the experiment2WithDataStorageEveryTrial.html file in Chrome, and then...



Use developer tools



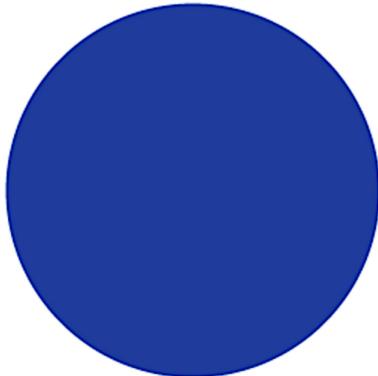
Use developer tools

If you then press a key to begin, you should see this:

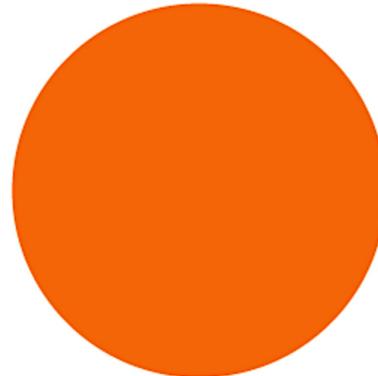
In this experiment, a circle will appear in the center of the screen.

If the circle is **blue**, press the letter F on the keyboard as fast as you can.

If the circle is **orange**, do not press any key.



Press the F key



Do not press a key

Press any key to begin.

```
Elements Console Sources Network > | : x
top ▾ Filter Info ▾
array(1) experiment2WithDataStorage every_trial.html:167
{
    [0]=>
    object(stdClass)#1 (8) {
        ["rt"]=>
        int(2033)
        ["key_press"]=>
        int(32)
        ["trial_type"]=>
        string(4) "text"
        ["trial_index"]=>
        int(0)
        ["time_elapsed"]=>
        int(2036)
        ["internal_node_id"]=>
        string(7) "0.0-0.0"
        ["subject"]=>
        int(46637)
        ["conditions"]=>
        string(10) "conditionC"
    }
}
Invalid query: Unknown column 'key_press' in 'field list'
```

This text corresponds
to the attempt to
insert data into your
MySQL table

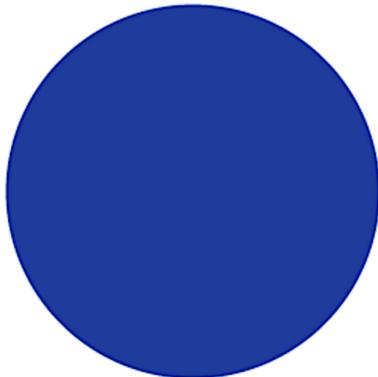
Use developer tools

If you then press a key to begin, you should see this:

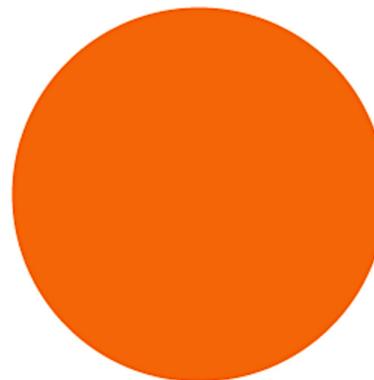
In this experiment, a circle will appear in the center of the screen.

If the circle is **blue**, press the letter F on the keyboard as fast as you can.

If the circle is **orange**, do not press any key.



Press the F key



Do not press a key

Press any key to begin.

```
Elements Console Sources Network > | : x
top ▾ Filter Info ▾
array(1) experiment2WithDataStorage every_trial.html:167
{
  [0]=>
  object(stdClass)#1 (8) {
    ["rt"]=>
    int(2033)
    ["key_press"]=>
    int(32)
    ["trial_type"]=>
    string(4) "text"
    ["trial_index"]=>
    int(0)
    ["time_elapsed"]=>
    int(2036)
    ["internal_node_id"]=>
    string(7) "0.0-0.0"
    ["subject"]=>
    int(46637)
    ["conditions"]=>
    string(10) "conditionC"
  }
}
Invalid query: Unknown column 'key_press' in 'field list'
```

Trial data sent



The first issue with data insertion to table

Use developer tools

If you run through the experiment, you can see each trial data output, as well as its insertion status

Then you know exactly what your table will need, including:

1. The name of each key
2. The type of data it holds
3. Whether or not it will always be sent with the rest of the trial data

Use developer tools

If you run through the experiment, you can see each trial data output, as well as its insertion status

Then you know exactly what your table will need, including:

1. The name of each key
2. The type of data it holds
3. Whether or not it will always be sent with the rest of the trial data

This info lets you create a table to accept / store your data

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. **Get your data into your MySQL table**
9. Miscellaneous tips
10. MTurk basics

Get data into your MySQL table

You can manually add the necessary columns to your table via the structure tab

The screenshot shows the MySQL Workbench interface with the 'Structure' tab selected. The main area displays a table structure with one column named 'rt'. The 'Action' column for this row contains links to 'Change', 'Drop', 'Primary', 'Unique', and 'Index'. Below the table, there are buttons for 'Check all', 'With selected:', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'. At the bottom, there are buttons for 'Print', 'Propose table structure', 'Move columns', and 'Improve table structure'. A search bar at the bottom left contains the text 'column(s) after rt'. A blue arrow points to the 'Go' button.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	rt	int(11)			No	None			Change Drop Primary Unique Index

With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#)

[Print](#) [Propose table structure](#) [Move columns](#) [Improve table structure](#)

[Add](#) 1 [column\(s\)](#) after rt [Go](#) ←

+ Indexes

Get data into your MySQL table

This way can take a really long time if you have many keys sent per trial

Screenshot of the MySQL Workbench interface showing the 'Structure' tab for a table named 'rt'. The table has one column 'rt' of type int(11). The 'Action' column shows options: Change, Drop, Primary, Unique, and Index. A blue arrow points to the 'Add' button at the bottom left.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	rt	int(11)			No	None			Change Drop Primary Unique Index

With selected: Browse Change Drop Primary Unique Index

Print Propose table structure Move columns Improve table structure

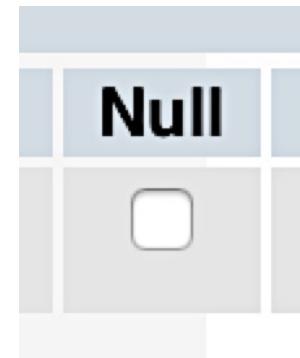
Add 1 column(s) after rt Go ←

+ Indexes

Get data into your MySQL table

So a brief explanation of the important inputs:

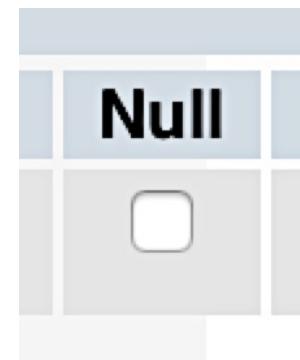
Type ?	Length/Values ?
INT	



Get data into your MySQL table

So a brief explanation of the important inputs:

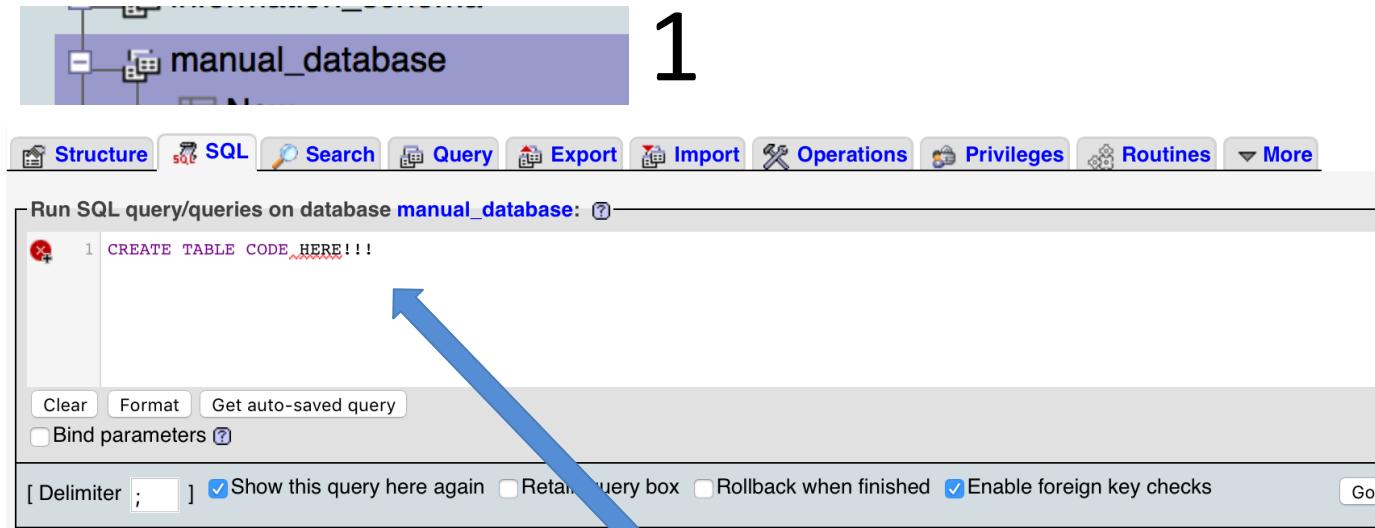
Type ?	Length/Values ?
INT	



But now a quicker way to do this

Get data into your MySQL table

To create a table and columns in one swoop:



COPY
CODE:

2

CREATE TABLE `manual_table2` (
 `rt` int(11) NOT NULL,
 `key_press` smallint(11) NOT NULL,
 `trial_type` varchar(200) NOT NULL,
 `trial_index` smallint(6) NOT NULL,
 `time_elapsed` int(11) NOT NULL,
 `internal_node_id` varchar(300) NOT NULL,
 `subject` int(11) NOT NULL,
 `conditions` varchar(200) NOT NULL,
 `stimulus` varchar(200) DEFAULT NULL,
 `response` varchar(100) DEFAULT NULL,
 `correct` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8

3

Get data into your MySQL table

What do the specifications mean:

```
CREATE TABLE `manual_table2` (
  `rt` int(11) NOT NULL,
  `key_press` smallint(11) NOT NULL,
  `trial_type` varchar(200) NOT NULL,
  `trial_index` smallint(6) NOT NULL,
  `time_elapsed` int(11) NOT NULL,
  `internal_node_id` varchar(300) NOT NULL,
  `subject` int(11) NOT NULL,
  `conditions` varchar(200) NOT NULL,
  `stimulus` varchar(200) DEFAULT NULL,
  `response` varchar(100) DEFAULT NULL,
  `correct` varchar(100) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

Get data into your MySQL table

Make sure to change the table name in your experiment file to manual_table2 (from manual_table1)

Then rerun your code, using developer tools

You should see:

```
array(1 experiment2 withDataStorage every trial2.html:167
) {
    [0]=>
    object(stdClass)#1 (8) {
        ["rt"]=>
        int(913)
        ["key_press"]=>
        int(32)
        ["trial_type"]=>
        string(4) "text"
        ["trial_index"]=>
        int(0)
        ["time_elapsed"]=>
        int(915)
        ["internal_node_id"]=>
        string(7) "0.0-0.0"
        ["subject"]=>
        int(69014)
        ["conditions"]=>
        string(10) "conditionC"
    }
}
successful insert! ←
```

Get data into your MySQL table

Screenshot of the MySQL Workbench interface showing a table named 'manual_table2'.

The top navigation bar includes: Browse (highlighted with a red box), Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers.

A message bar indicates: "Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available." (Yellow background)

A success message bar shows: "Showing rows 0 - 6 (7 total, Query took 0.0004 seconds.)" (Green background)

The SQL query entered is: `SELECT * FROM `manual_table2``

Below the query are several action buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh.

Table filtering options include: Show all (unchecked), Number of rows: 25, and Filter rows: Search this table.

The table has the following columns: rt, key_press, trial_type, trial_index, time_elapsed, internal_node_id, subject, conditions, stimulus, response, and correct.

rt	key_press	trial_type	trial_index	time_elapsed	internal_node_id	subject	conditions	stimulus	response	correct
913	32	text	0	915	0.0-0.0	69014	conditionC	NULL	NULL	NULL
96356	70	text	1	98278	0.0-1.0	69014	conditionC	NULL	NULL	NULL
354	70	single-stim	2	100637	0.0-2.0-0.0	69014	conditionC	img/orange.png	no-go	
1039	70	single-stim	3	102485	0.0-2.0-1.0	69014	conditionC	img/blue.png	go	1
570	70	single-stim	4	104043	0.0-2.0-2.0	69014	conditionC	img/blue.png	go	1
281	70	single-stim	5	106517	0.0-2.0-3.0	69014	conditionC	img/orange.png	no-go	
142	68	text	6	108614	0.0-3.0	69014	conditionC	NULL	NULL	NULL

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. **Miscellaneous tips**
10. MTurk basics

Miscellaneous tips

Developer tools

Sometimes if code isn't working you'll simply see a blank experiment screen

- the developer tools console however often automatically displays errors

In JavaScript you can display things to the console by adding: `console.log(<stuff to display>)`

- This is really helpful for debugging
- You can place this code in the `<script>` tags

Miscellaneous tips

MySQL: Get the code you need to recreate a table you already have

Have one table created and need another one or a similar one?

Miscellaneous tips

The screenshot shows a MySQL Workbench interface. At the top left, there's a tree view showing a single database named "manual_database". Below the tree is a toolbar with several tabs: Structure, SQL (which is highlighted with a red box), Search, Query, Export, Import, Operations, Privileges, Routines, and More. The main area is titled "Run SQL query/queries on database manual_database". Inside this area, there's a code editor containing the following SQL command:

```
1 SHOW CREATE TABLE manual_table2;
```

This command is labeled with a large number "2" below it. At the bottom of the SQL editor, there are several buttons: Clear, Format, Get auto-saved query, Bind parameters (with a question mark icon), Delimiter (set to ";"), Show this query here again (checked), Retain query box (unchecked), Rollback when finished (unchecked), Enable foreign key checks (checked), and a Go button.

Large numbers 1, 2, and 3 are overlaid on the interface to indicate specific points of interest: 1 is over the database tree, 2 is over the SQL tab, and 3 is over the SQL editor area.

Miscellaneous tips

- Options

Partial texts Show binary contents Hide browser transformation Geometry
 Full texts Show BLOB contents Well Known Text
 Well Known Binary

1 Go

Table	Create Table
manual_table2	CREATE TABLE `manual_table2` (`rt` int(11) NOT ...

2

Miscellaneous tips

+ Options

Table	Create Table
manual_table2	<pre>CREATE TABLE `manual_table2` (`rt` int(11) NOT NULL, `key_press` smallint(11) NOT NULL, `trial_type` varchar(200) NOT NULL, `trial_index` smallint(6) NOT NULL, `time_elapsed` int(11) NOT NULL, `internal_node_id` varchar(300) NOT NULL, `subject` int(11) NOT NULL, `conditions` varchar(200) NOT NULL, `stimulus` varchar(200) DEFAULT NULL, `response` varchar(100) DEFAULT NULL, `correct` varchar(100) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>

Miscellaneous tips

MySQL: Outputting the contents of your table to an excel file

1. Select the table you want to export, then select the following tab and options:

The screenshot shows the MySQL Workbench interface with the 'Export' tab highlighted by a red box. The 'Export' tab is located in the top navigation bar, which also includes 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Import', 'Privileges', 'Operations', and 'Triggers'. Below the navigation bar, the main content area displays the following configuration options for exporting data from the 'manual_table2' table:

Export method:

- Quick - display only the minimal options
- Custom - display all possible options

Format:

CSV for MS Excel

Rows:

- Dump some row(s)
 - Number of rows: 7
 - Row to begin at: 0
- Dump all rows

Output:

Rename exported databases/tables/columns

Miscellaneous tips

MySQL: Outputting the contents of your table to an excel file

Output:

- Rename exported databases/tables/columns
- Use `LOCK TABLES` statement
- Save output to a file

File name template: 

manual_table2_data



use this for future exports

Character set of the file:

utf-8

Compression:

None

- View output as text

Skip tables larger than MiB

Miscellaneous tips

MySQL: Outputting the contents of your table to an excel file

Format-specific options:

Replace NULL with:

Remove carriage return/line feed characters within columns

Put columns names in the first row

Excel edition: 

Miscellaneous tips

MySQL: Outputting the contents of your table to an excel file

Example file

Miscellaneous

Current experiment folder for MTurk study

FOLDER

[`mturk_study_example_RAS`](#)

THINGS TO CHANGE

- Change [`database_connect.php`](#) file to match your requirements
- Set the desired MySQL table in the experiment files
 - [`cr_mturk_session1.php`](#)
 - [`cr_mturk_session2.php`](#)
- These are old consent & debriefing forms

Miscellaneous

Current experiment folder for MTurk study

Includes

Checking for browser (Chrome / Mozilla)

Consenting in an external html form (via html plugin)

Obtaining and checking MTurk ID

Creating stimuli / html formatting

Online response scoring

Time stamps (UTC)

Final demographic and other questions we typically ask on MTurk

Some plugins I've changed (formatting/data logging/presentation changes)

- any plugin with RAS in it means it's a modified jsPsych plugin
- so it may differ from online documentation

Note that the css has been altered (although I left name as is)

DISCLAIMER: if statements and exclamation points

Outline

1. Links & starting tips
2. Set up a local web development environment
3. Get going with jsPsych basics
4. Set up MySQL database and table
5. Hook jsPsych up with MySQL database & table
6. Understand MySQL table and data insertions
7. Use developer tools
8. Get your data into your MySQL table
9. Miscellaneous tips
10. MTurk basics

MTurk basics

- see MTurk link
- Requester login info
- see *SubjectPaymentGuidelines_McDermottLab.docx*, *Sample_subjects_spreadsheet.xlsx*,
MTurk_subject_spreadsheet_inUse.xlsx (used for current study/based on Jeff's spreadsheets)
 - purchasing HITs; keeping track of MTurk participation and payment
 - See section “For studies conducted on MTurk:”

MTurk basics

Create Tab

- walk through (copy previous for starting point)
- max 9 assignments per batch
- what an MTurk Worker sees/does

MTurk basics

Manage Tab

Results

- example batch (Results / Info)

MTurk basics

Manage Tab

Results

- example batch (Results / Info)

Workers

- can download full workers csv
 - can help with checking qualifications were given correctly

MTurk basics

Manage Tab

Results

- example batch (Results / Info)

Workers

- can download full workers csv
 - can help with checking qualifications were given correctly

Qualification Types

Why:

- Avoid repeat subjects (quals for given session + stimulus set)
- Allow specific subjects to complete follow-ups

How:

- Assign manually on individual worker page
 - *see MTurk subject page link example (replace <WORKER_ID>)*
- Assign in groups via R (must have a HIT accepted by MCL first)
 - *see R code (contains MCL account info)*

MTurk basics

R Code

Must have “MTurkR” in library to use

MTurk basics

R Code

Once a worker has had at least 1 HIT accepted by MCL, you can:

- give qualifications (previous slide)
- send emails to invite workers to complete future sessions (wait ~10min)
 - *see R code (contains MCL account info)*

MTurk basics

R Code

Once a worker has had at least 1 HIT accepted by MCL, you can:

- give qualifications (previous slide)
- send emails to invite workers to complete future sessions (wait ~10min)
 - *see R code (contains MCL account info)*

To test R code on yourself:

You cannot do this unless you have a HIT accepted from MCL account:

1. Create an Mturk worker account for yourself
2. Via the MCL account, create a HIT with 1 assignment worth \$0.00
(Mturk will take a small %, which ends up at a cent or so)
3. Submit this hit with your Mturk worker account
4. Accept the HIT with the MCL account

Now furthermore (because you have at least 1 HIT accepted from MCL) you can test the R code qualifications and emails on yourself

MTurk basics

We get rated on lots of things, which then workers may use to determine if they will participate

Here are some things to try to always do when running a study on Mturk:

- check if HITs are valid quickly/ record HIT info, and then accept or reject HITs quickly
- have MCL's email open (within reason) when HITs are available and try to respond quickly
 - try to leave plenty of time for workers to finish the task
 - if experiment takes 15min, allow ~40min to complete HIT on MTurk

Questions?