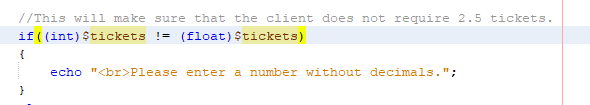
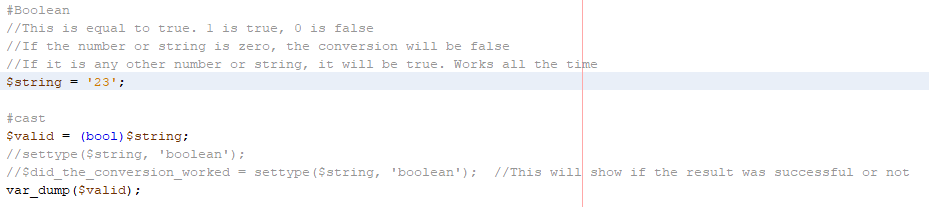
# PHP Journal

## Variables

### Check if the user typed a decimal input



### Boolean



* In this case the result will be true. If you convert any number/string that is not zero to Boolean, it will be converted to true. Only zero will be converted to false.

### String

#### Length

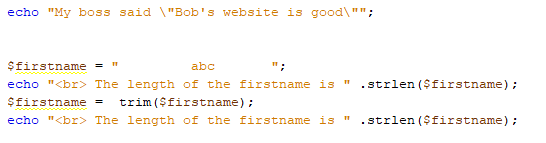


Two ways of finding the length a string

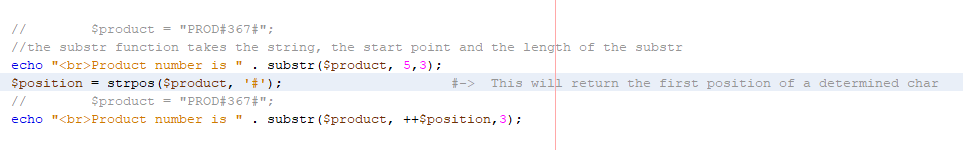
* strlen()
* mb\_strlen()

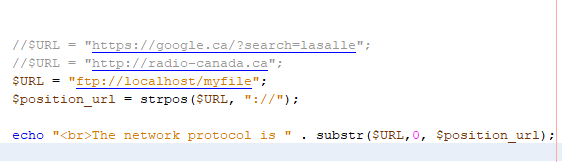
The difference between the two of them is that if we use strlen(), words like ‘’Montréal’’ would have one character more because of the accent and this is not what we want when validating an input. That is why it is better to always use mb\_strlen()

#### Trim



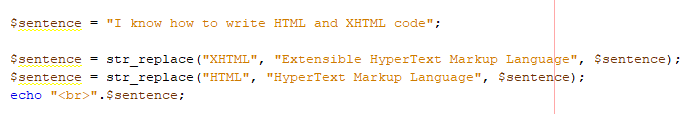
#### Substring and Strpos





#### Str\_replace

Modify and change a part of a string for another

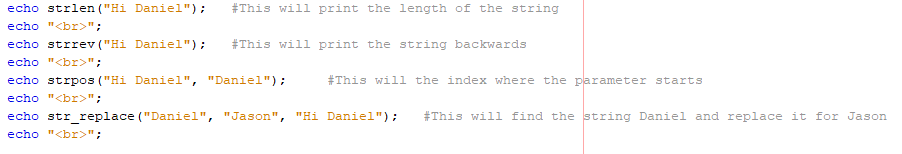


Explode and implode

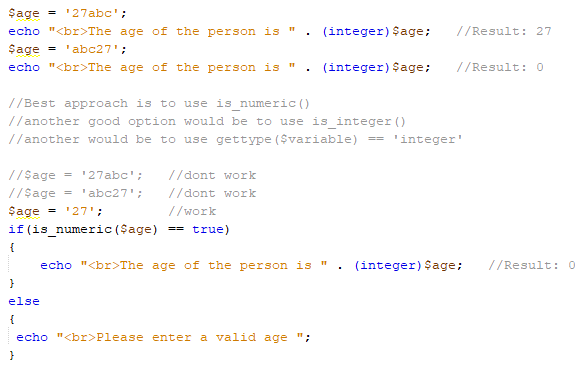
Separate a string by a certain char and then unite it by a specific char.

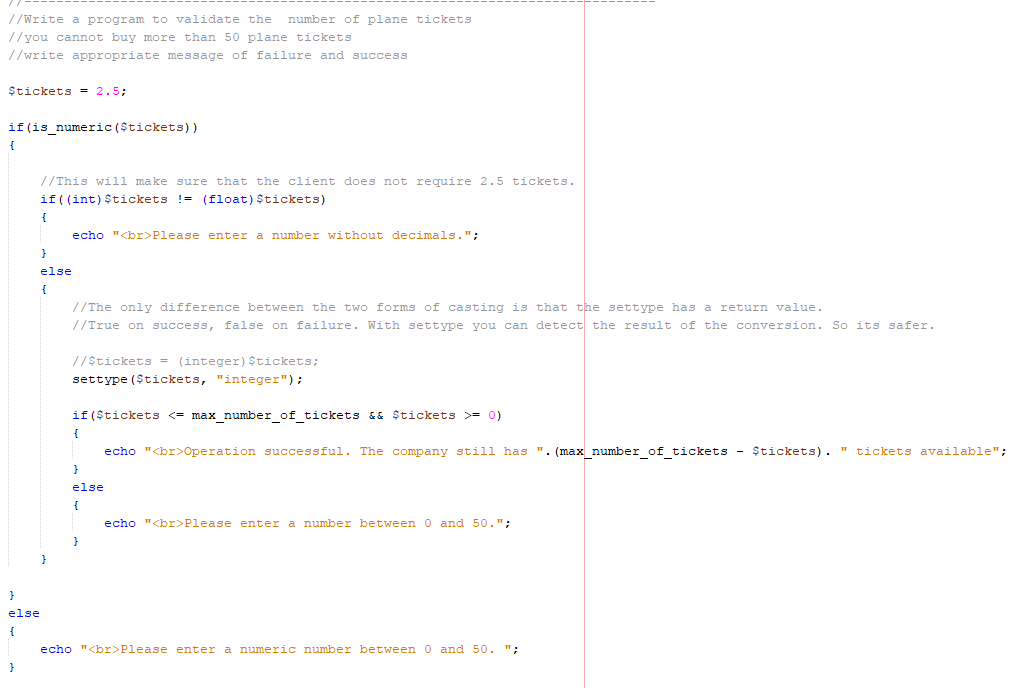


#### Summary of functions



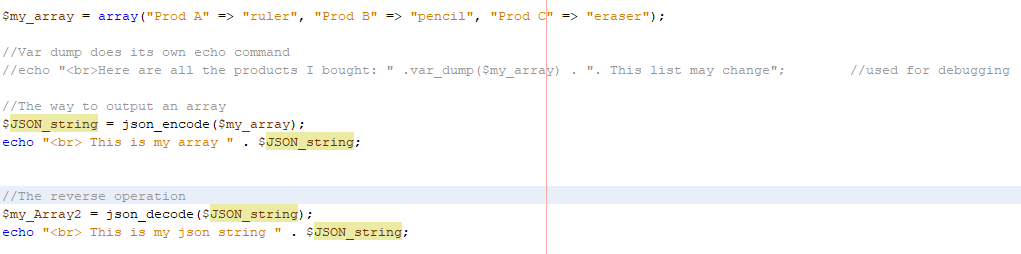
## Casting



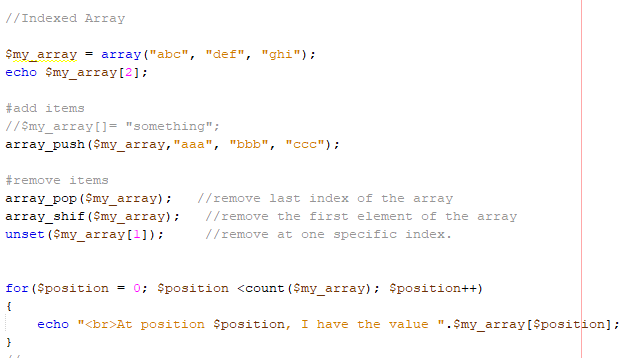


## Arrays

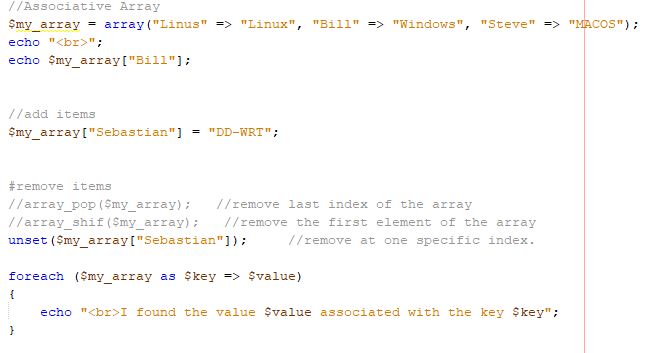
### How to output an array



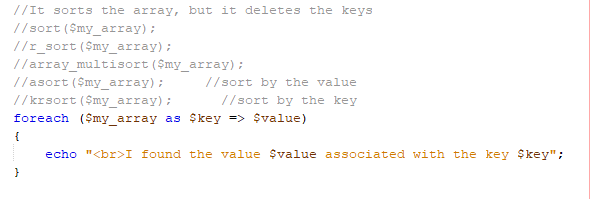
### Indexed Array



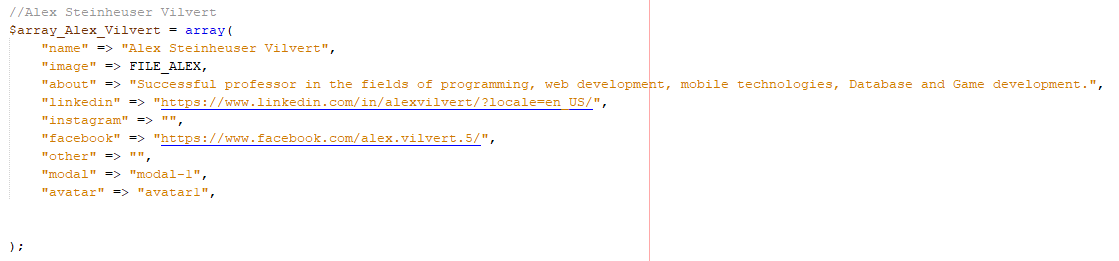
### Associative Array

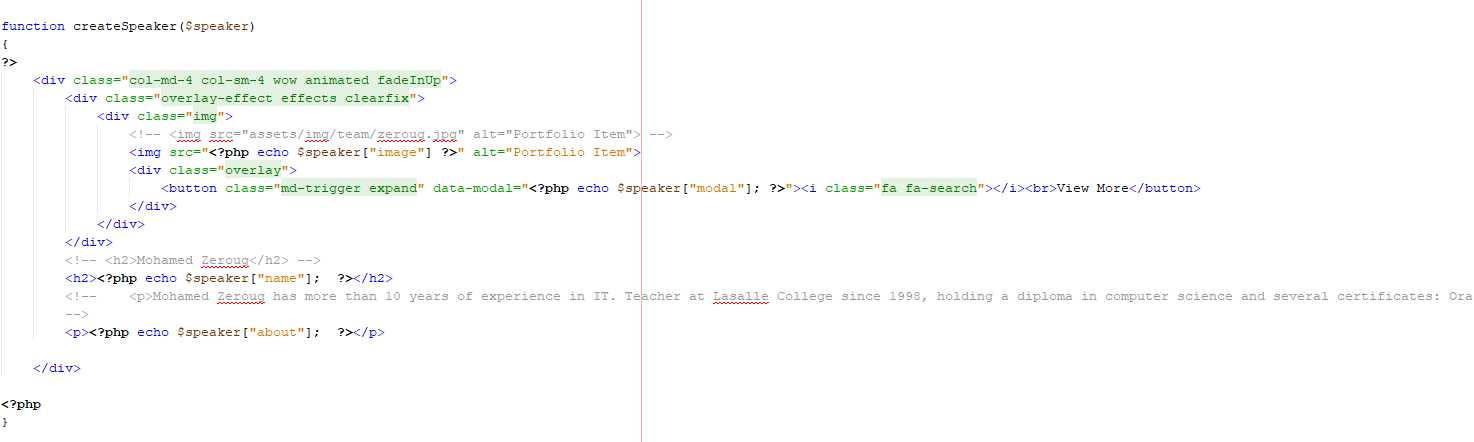


### Sorting

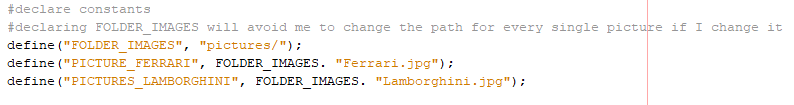


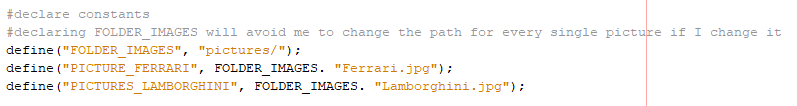
### Array applied





## Defining a constant

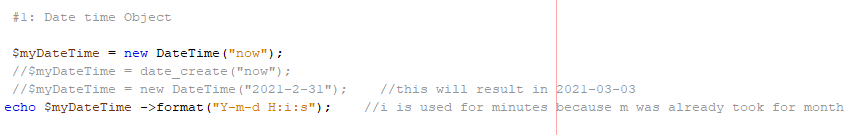




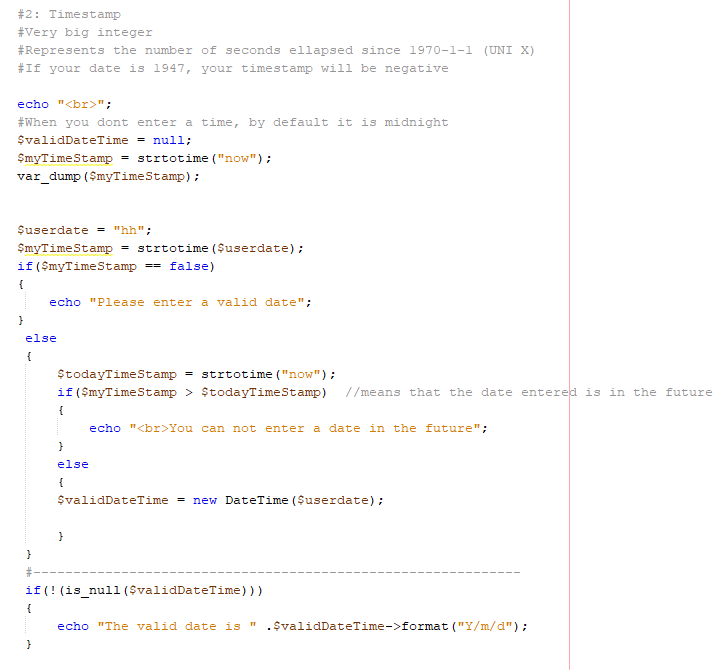
## DateTime

Different ways to manage Date the date

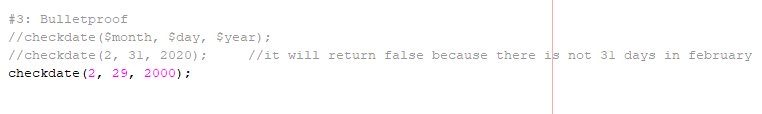
### 1: Date time Object



### 2: Timestamp



### 3: Bulletproof



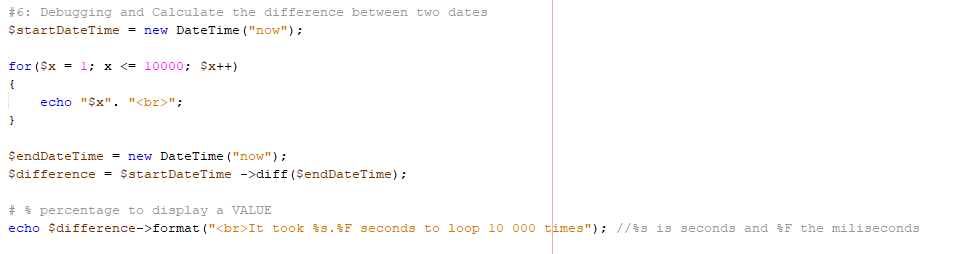
### 4: Array



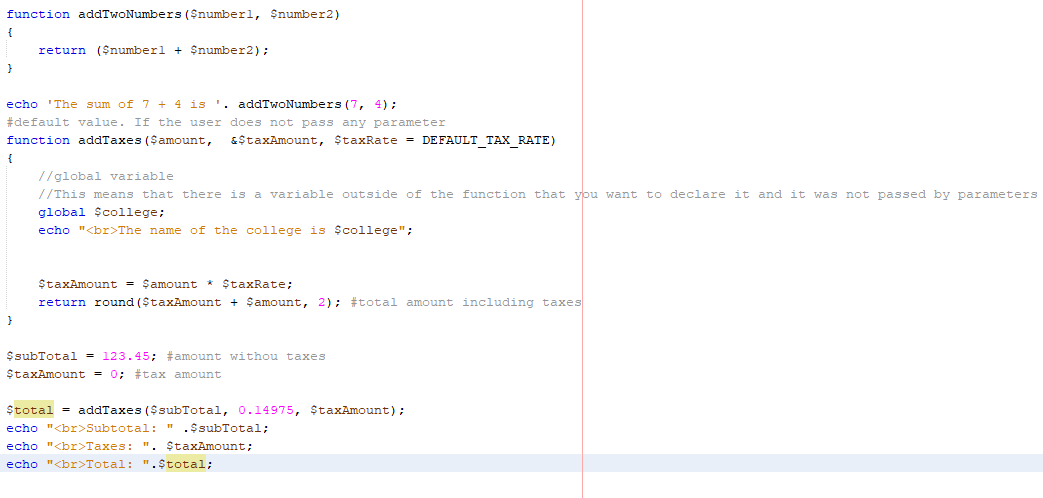
### 5: Shortcut to get a quick string representation



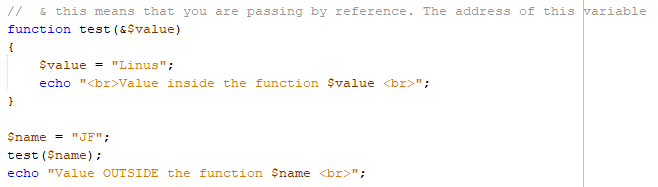
### 6: Debugging and Calculate the difference between the two dates



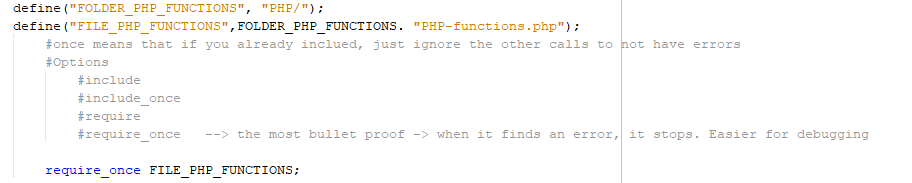
## Functions



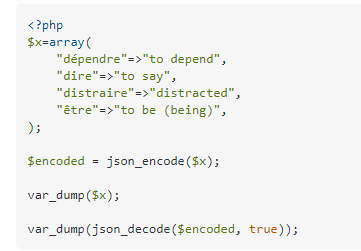
### Passing by reference



## Import one file to another

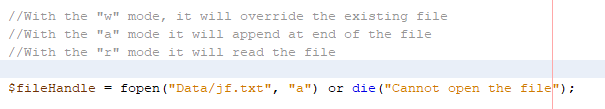


## Json\_encode



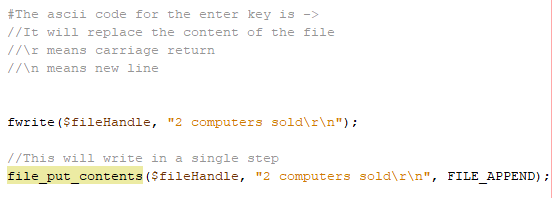
## Text File

### Modes



* It is important to notice that this will only create a text file in the htdocs, not on the NetBeans folder

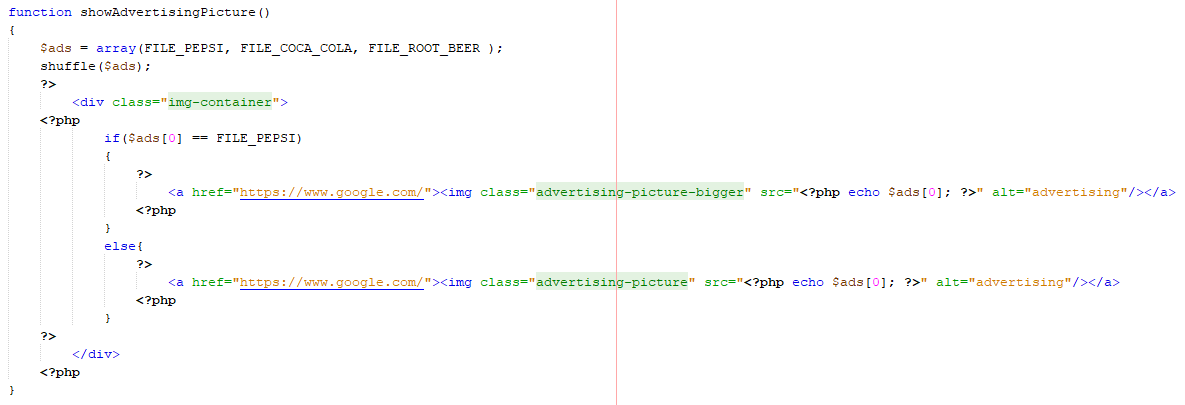
### Write



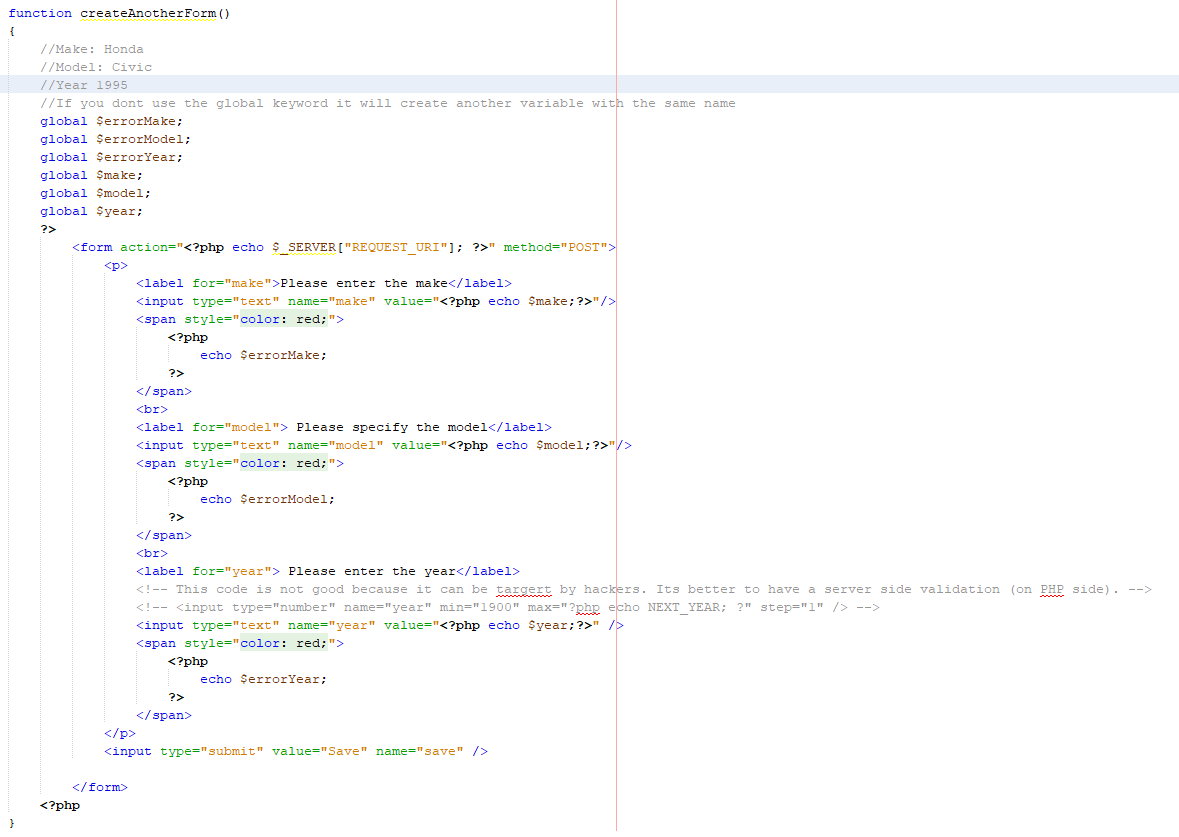
### Read



## Function to generate handle advertising picture



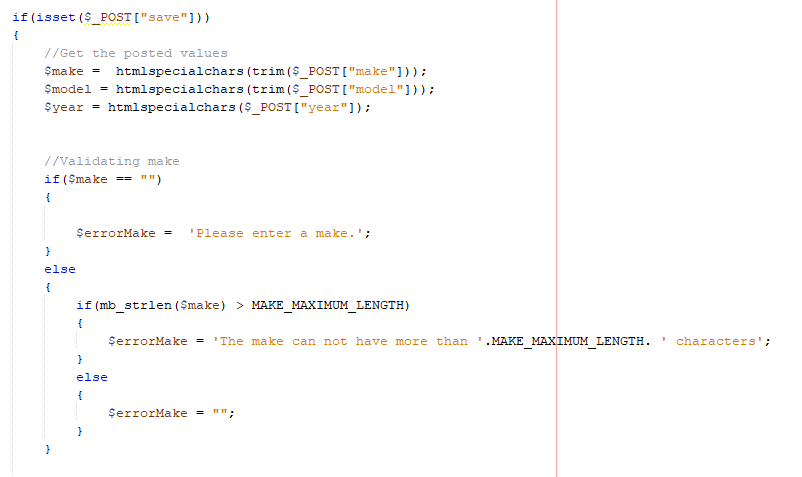
## Create Form



* Notes
  + It is important to note here the value for each input, by putting the own variable inside will avoid that the input is erased on the textbox if the validation is not successful, allowing the user to modify and submit again.

### Reading and validating input against injection

* Types of functions to validate data and protect against injection
  + Filter 1
    - Removes special chars and gets string. Would not be the best because we don’t want to remove input from the user
    - Would not get accents properly on the page source as well
    - Montréal would have a length of 8 instead of 7, because of the accent
      * echo "<br><br>Filter #1 SANITIZE STRING: <br>";
      * echo filter\_input(INPUT\_POST, "firstname", FILTER\_SANITIZE\_STRING);
  + Filter 2
    - Remains the special chars. It kept the script but it replace the '<' for codes in the page source
    - It would not be the best choice becaause it dont get the accent properly
      * echo "<br><br>Filter #2 SANITIZE FULL: <br>";
      * echo filter\_input(INPUT\_POST, "firstname", FILTER\_SANITIZE\_FULL\_SPECIAL\_CHARS);
  + Filter 3
    - Remains the special chars. It kept the script but it replace the '<' for codes in the page source
    - This is the one recomended because it gets the accent properly on page source and it blocks injection as well.
      * echo "<br><br>Filter #3 htmlspecialchars: <br>";
      * echo htmlspecialchars($\_POST["firstname"]);
  + Filter 4
    - Remains the special chars. It kept the script but it replace the '<' for codes in the page source
      * echo "<br><br>Filter #4 htmlentities <br>";
      * echo htmlentities($\_POST["firstname"]);



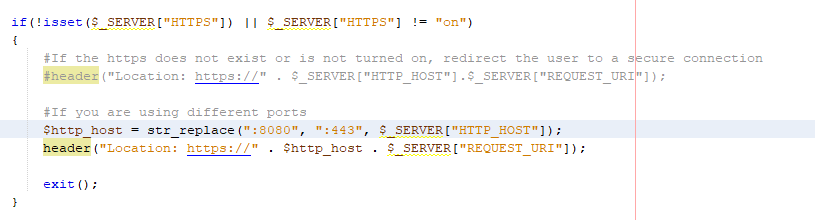
* Notes
  + The save is the name given to the button

## Password Hash

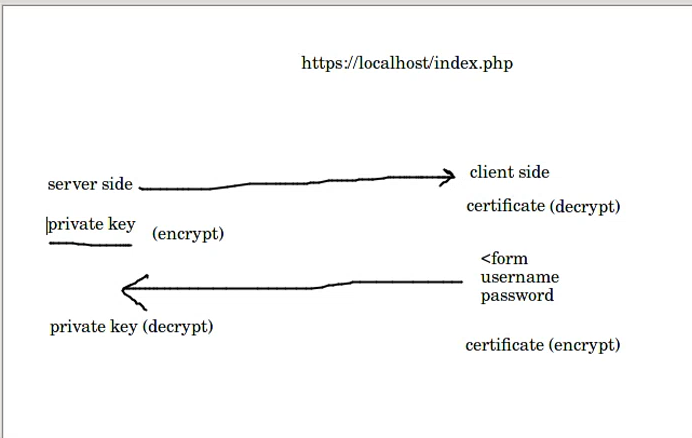


## Encryption and Decryption

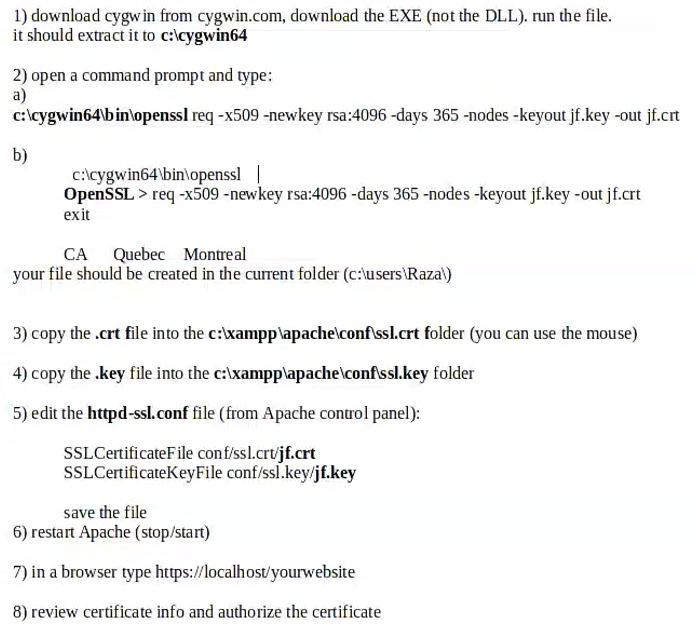
* You can encrypt with the private key and decrypt with the certificate (public key)
* You can encrypt with the public key (certificate) and decrypt with the private key
* Default port with HTTP is 80
* Default port with HTTPS is 443
* URL (Uniform Resource locator) is divided into some parts
  + Networkd protocol
    - ://
    - Is optional
  + Domain
    - Mandatory
    - Computer reads the domain from right to left
  + URI (Uniform resource identifier)
    - /
    - Folders
    - Files
    - Parameters

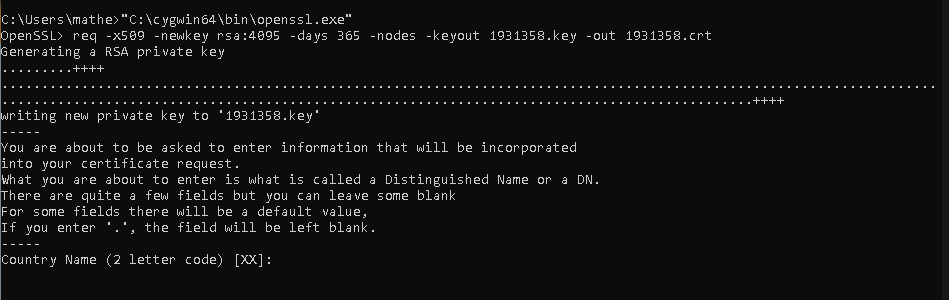


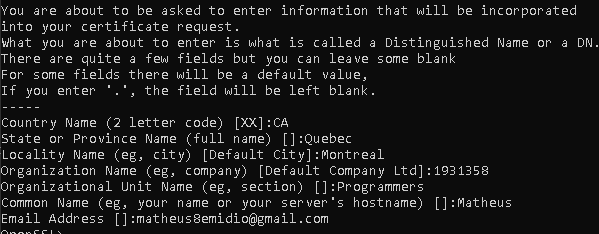
## Create key and certificate



* Open cmd prompt
* Don’t change directories when typing the command so that your file can be created at your currend directory







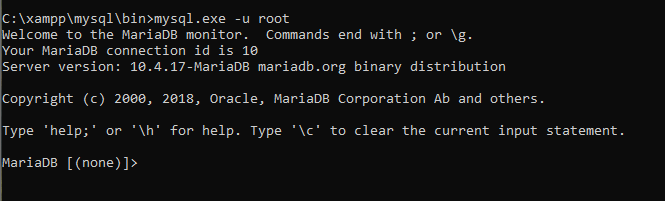
## Database

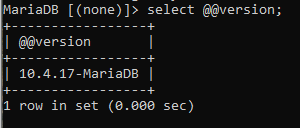
* Administrator
  + Root: Linux user
* Database
  + Root: MariaDB user
* Make sure the root password is not empty
* Text-based command will be used to connect to mySQL

Guide:

1. Open a command prompt
2. C:\xampp\mysql\bin\mysqladmin.exe -u root
   1. I did not specify a password
   2. Root as an empty password
   3. Replace it for
      1. C:\xampp\mysql\bin\mysqladmin.exe -u root -p
         1. This means that you will specify a password
3. If unable to connect
   1. Select @@version:
   2. Exit;
4. If root password is empty
   1. C:\xampp@mysql\bin\mysqladmin.exe -u root password NEWPASSWORD
5. In case you really don’t know the password
   1. Edit my.ini file, search the section called [mysqld] and below it add:
      1. skip-grant-tables
      2. then stop and restart mysql and redo the steps
6. Remove the skip-grant-tables after solving it
7. If you want to modify existing password
   1. C:\xampp\mysql\bin\mysqladmin.exe -u root -p password (your new password)
8. My password will be
   1. **matheusemidio**

* Install HeidiSQL

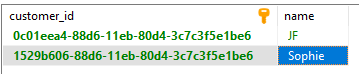




## HeidiSQL

* Right click root
* Create new database
* Allowed characters for name
  + 0-9
  + a-z
  + $
  + \_
* Collation setting on a database
  + Ci means if there is a lot of data from a-z is not going to care about if you used lower case or not. -> case insensitive
  + Ai -> accent insensitive
  + Query would return both Montreal or mineapolis, even knowing that they have different case
  + The best option for the project would be choosing
    - Uft8mb4\_general\_ci
    - Or uft8\_general\_ci (but its not stable)
    - utf8
    - mb4
      * every byte will take 4 bytes
      * database will take 4 times more space
    - Table names should be plural
    - Field name should be singular
    - When adding a column, remember to maybe set a default value
    - Right click column number and create new index (primary key)
    - Go to data, right click the main screen and insert row
    - The professional way to set up a primary key is using UUID
      * SELECT UUID() will prompt an UUID in the screen
      * Datatype should be char
      * Length 36
      * Drawback is that it will be slow for the computer





* When we need some information typed by the user, always user VARCHAR (length is variable)
* Remember to use the correct INTEGER to hold the value. Not too big, not too small.
* For decimals, USE DECIMAL type
* Use date for dates
* Datetime for handle dates and times
  + Default value is now()
* UNIQUE key is similar to the primary key, can not have duplicate entry
* KEY is for you to be able to search it later in WHERE conditions, INNER JOINS, foregin key
  + Don’t use too much of them

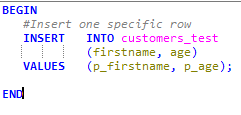
## Create an user and give permissions

* Click the two faces “Manage user authentication and privileges”
* Add a new one
* Fill the fields
* The second line you don’t need to change, but is vital, means that you can only connect using the same machine.
* Don’t check the global privileges box
* Click add object
* Go to the current database
* Don’t select the tables, only the views and stored procedures.
* Click Ok
* For all the other ones, check the first green checkbox meaning that this user only has the right to execute the stored procedure and select the data from the view.

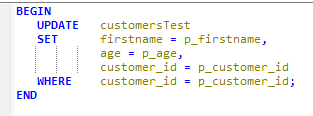
## Stored Prodedure

* Always write all the data you need on the queries. Avoid using \*
* Right click the database and put create new -> stored procedure
* Procedure returns rows, not a result
* Use procedure to deal with SQL data and function to deal with variables
* TableName\_action of the procedure
* Remember to order by
* If its not saving, try debuging it by commenting some parts of the query to check which one is not valid
* Right click the procedure and click run routine

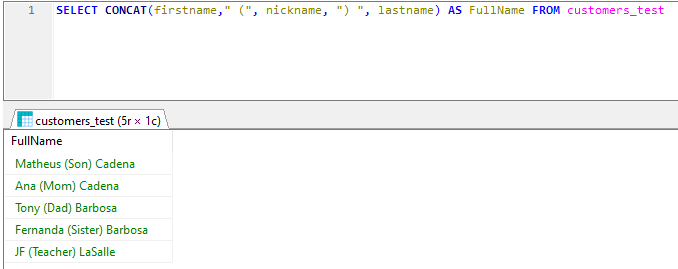
INSERT



UPDATE

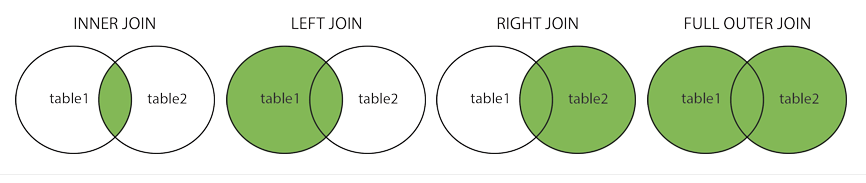


## View

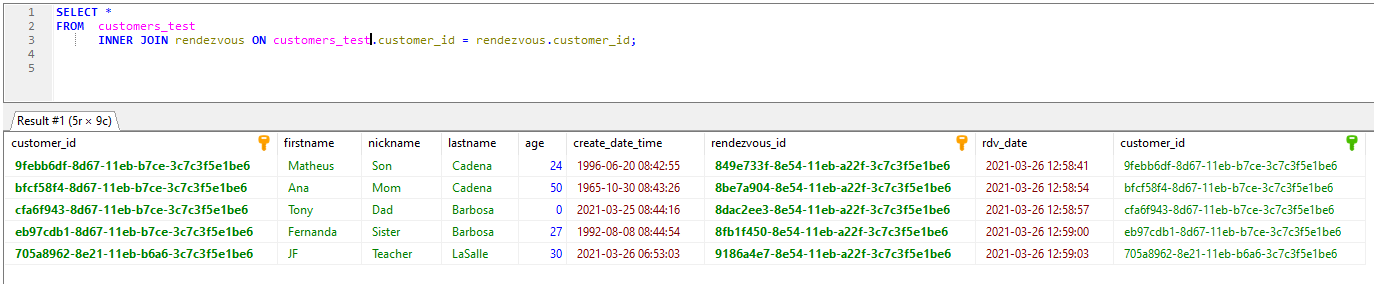
* You create a view because you want to view the fields
* You can then select \* from your view
* AS keyword just means to change the display column name
* CONCAT function will put one string after another
  + CONCAT(firstname, nickname, lastname) AS Full Name;
  + 

## Joins

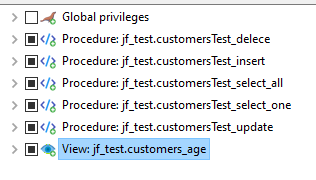
* It’s preferable to use INNER JOIN instead of WHERE conditions
  + Easier to debug
  + JOIN
    - Default one.
    - It’s like the inner join
    - Bad practice
  + FULL OUTER JOIN
    - I want to see the customers without rendezvous
    - I want to see the rendezvous without customers
    - I want to see all the data
  + INNER JOIN
    - Shows rows when data exist in both tables
    - If I have an emp linked to a rendezvous, I see it
    - If I have an emp that does not have a rendezvous, I don’t see it
  + LEFT JOIN
    - The data to the left of the join is the most important data.
    - If customer is to the left, I will see all the customers, even those that don’t have a rendezvous.
  + RIGHT JOIN
    - The data to the right of the join is the most important data
    - In the case rendezvous is to the right I will see all the rendezvous, even the ones that don’t have a customer.



* For foreign key
  + Use FK\_nameoftable\_nameofothertable



* Don’t give permission to the table, just to the views
* Click manage user authentication and permissions
  + Add a new one
  + The icon is the 8 on the top menu
  + Don’t check the global privileges checkbox
    - Add object
    - Go to the current database and select the views. Use CRTL to select more than one.
    - Just have the right to execute the stored procedure and Select the view



* Leave HeidiSQL
* On the session manager, create a new one
  + Name it
  + Put the users
  + Password
  + Save and open it

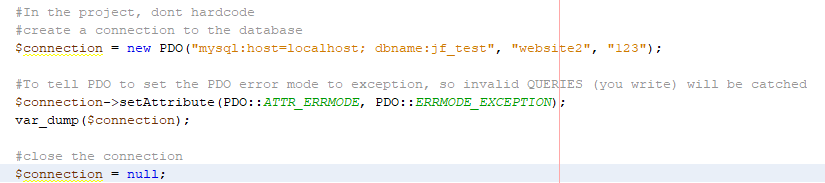
## How to back up a database?

* + Comments should be between the begin and the end, not at the top, otherwise it will be excluded.
  + Right click the database
  + Select export database as SQL
  + Don’t choose the binary file
  + Choose SQL text file
  + Make sure the two CREATE checkboxes are checked
  + On the data field, it has to be INSERT
  + Click the yellow folder, select your folder of desire
  + Name it like
    - Jf\_test\_2021-03-26.sql
  + Click export
* The missing thing when you run again is the user and the permissions
  + Go to a query and type
    - SHOW GRANTS FOR website2@localhost
    - Press CRTL A to select everything, right click and press COPY SELECTED ROWS
    - Save it to another file and run it.
    - Don’t put at the end of the backup, it will generate errors.

## Connection

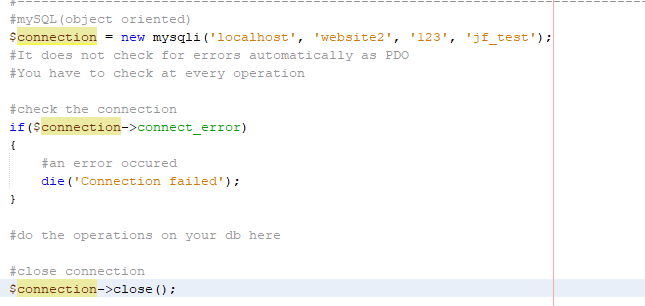
### First approach

PDO



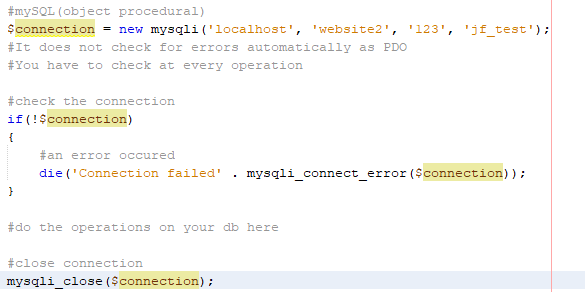
### Second approach.

Using MySQL object oriented



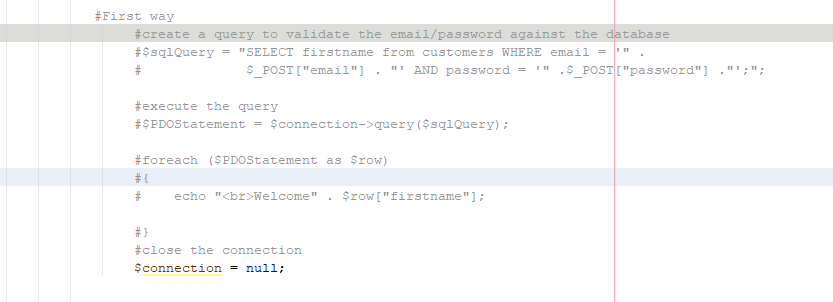
### Third approach

Using MySQL procedural



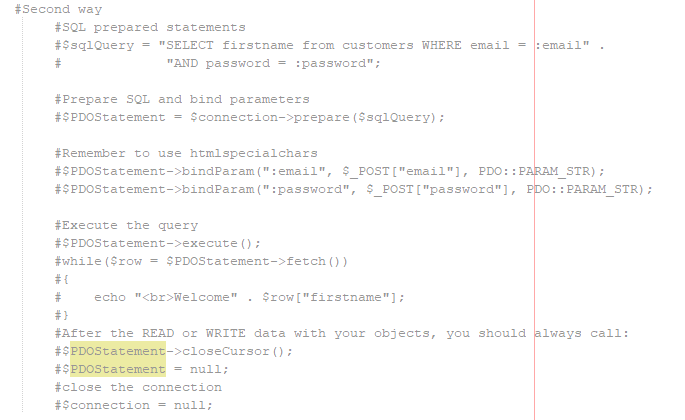
### Bad approach

Vulnerable to SQL injection

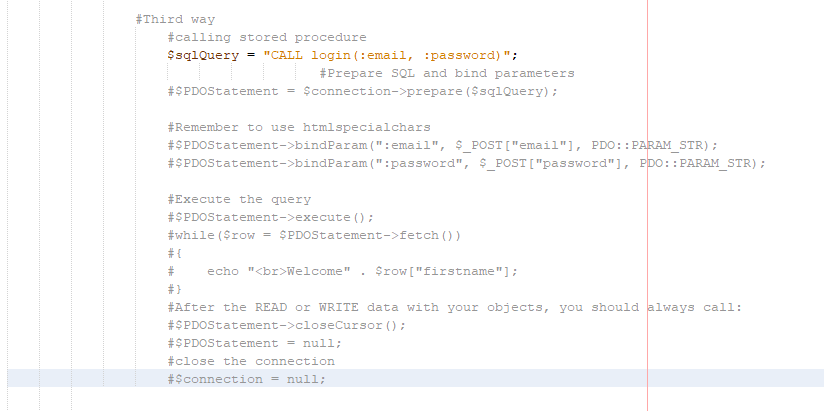


Never write sql queries inside PHP

### Another approach



### Good approach

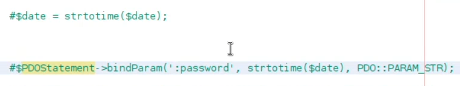


* In the project we need SQL Prepared Statements to call stored procedures, this way we will have the code protected against SQL injection and we will have our queries separated from php commands

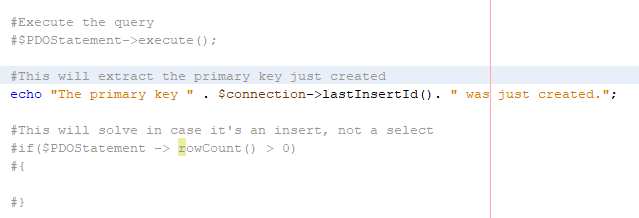
## Passing by reference

* You can pass by reference using the & before the variable

## Dates in the project



## How to prove you can read data when you are writing

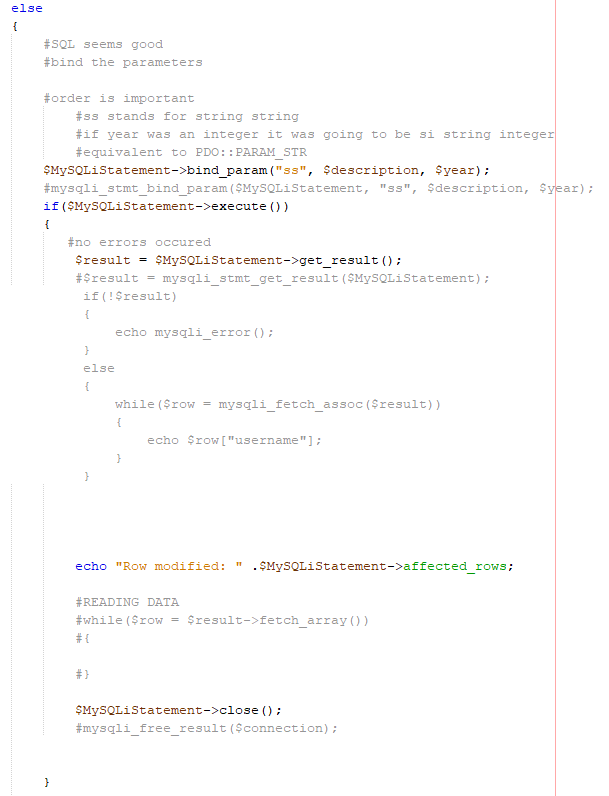


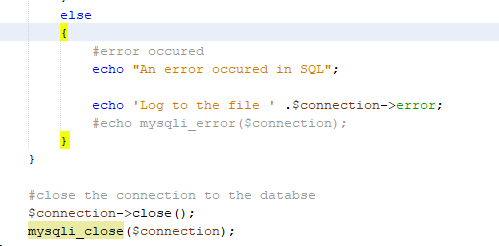
## Array of database content



## Using SQL

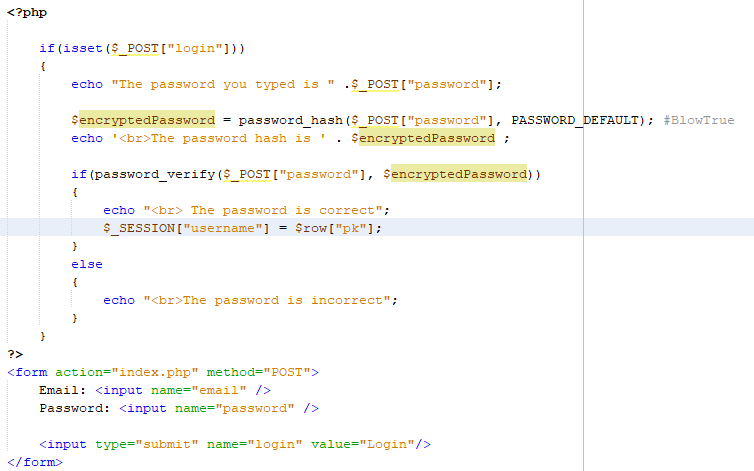
* In the code below we have two ways to do the same thing
* The one in comment is the mysqli procedural, where we call functions to perform the job



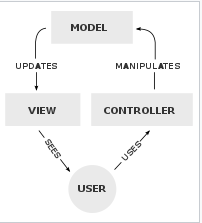
## Login

* You should encrypt the password when comparing for security purposes (create a password hash)
* Encryption: the data is different from the password
* By definition you can not decrypt a password hash
* You can compare a password hash with a password (ghi)
* The encrypted data is the same as the password
* Password hash is a one-way encryption
* Even if every customer used the same password, the hash would be different for every single one
* Use a single quote when comparing the hash, because if you use double quote, php will try to solve the dollar sign inside the hash as a variable
* Use varchar 60 or 255
* Encryption is handled by php not by the database
* The login stored procedure will just have email as parameter, but will return the primary key and hash password
* After login is successful, save a session variable.
* On the database, we will not know the password for sure, we will know the hash for the user password.
  + That’s why in secure websites, they can never tell you your password, because they don’t know it. They can allow you to change it, and when you do, they will store the hash for the new password, but not literally the password.
* When you receive the data from the user when registering, you will save the data as usual, validate it and then you will apply the hash on the password before saving on the database. Never save the password in clear text on the database.

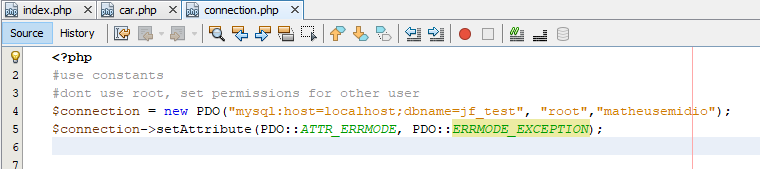


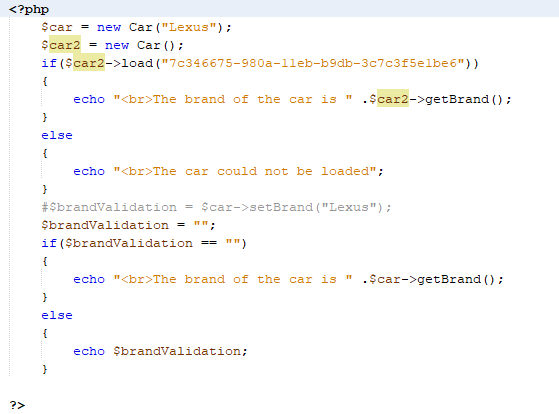
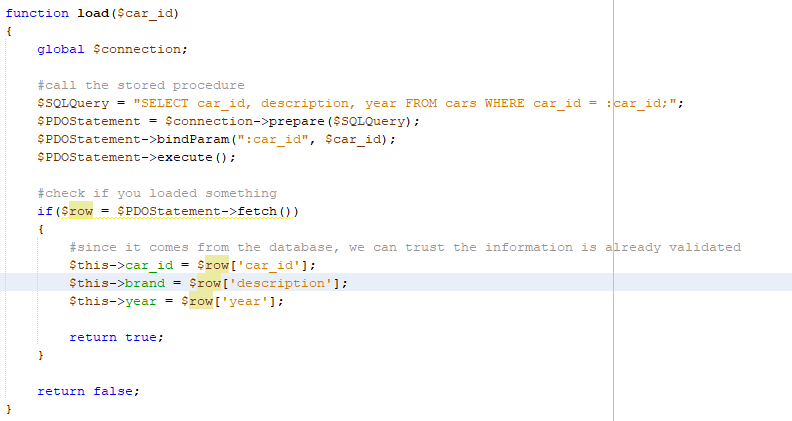
## Objects

* MVC (Model View Controller)
  + A way to build your application/layout
    - Model
      * Classes/objects
      * Business rules
    - View
      * Output
      * What the user sees
      * Receives the data from the model
    - Controller
      * Input, user entry, commands
* Plural will be a list of singular products
* When you create a class, you can
  + Create variables
  + Create functions
  + You can not echo something
* The model only receives information and give information, neve communicates with the user
* Remember to initialize your properties at creation time

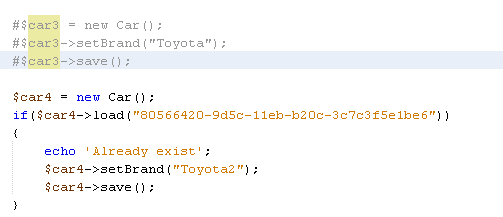


* On your project, move the validation to the setters
* Use self::constant\_name when accessing inside an object or class (access static data)
* Constructor will only be use for data that are already on the database.
  + If you wanted, you could use the setters inside the constructor for validation purposes
* But the approach used is going to be using setters for client input and constructor when data is already verified.

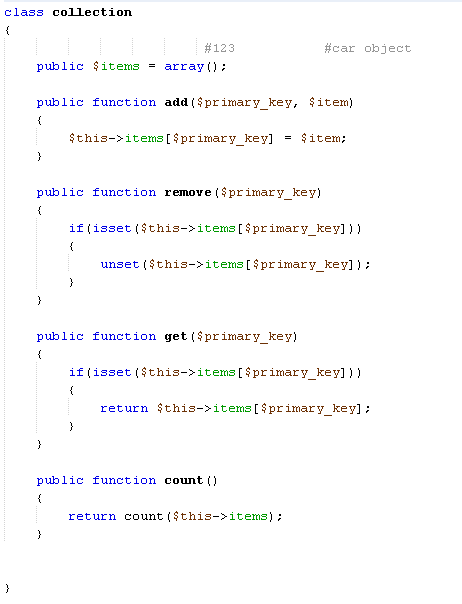




## Save



## Collection

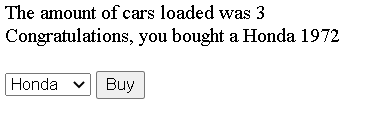




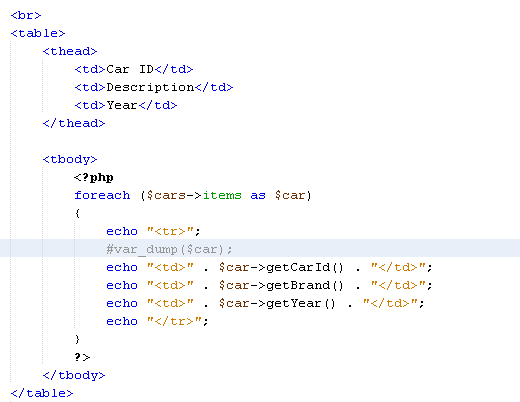
## Interactions with HTML

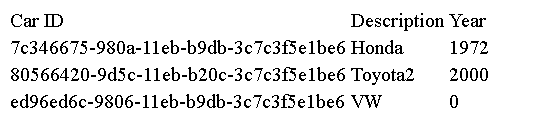
### Form





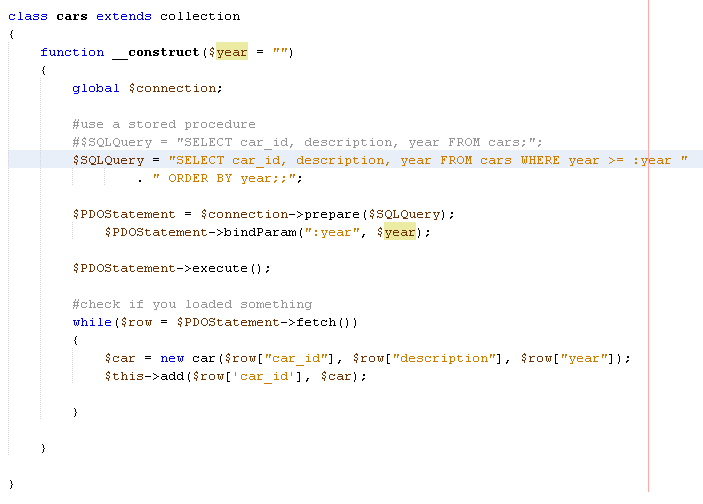
### Table

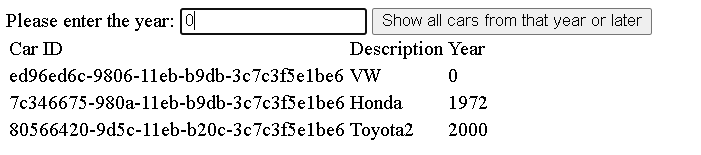


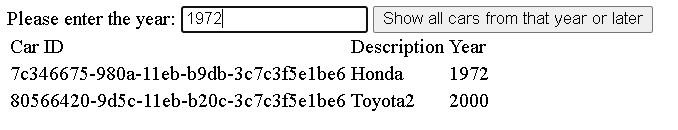


### Searching









# AJAX

* Asynchronous JavaScript and XML
* Synchronous means that the previous task must be completed before you start a new one
* Asynchronous is the opposite. Multi-tasking
* When debugging with Ajax, take a look at the inspector, not the page source

