



<details>

Document: Developer Test
Date: 02/01/2018

</details>

Developer Test

List Grouping

Arrange a list of relatives from the oldest generation to the youngest.

Each generation must be on their own level of list. Each level must be in alphabetical order.

First generation: James and Mary

Second generation: Tom, Morris and Maria

Third generation: Taylor, Timothy, Alexander, John, James and Juliette

Fourth generation: Dmitry, Denis, Delore and Samuel

If new generation is added then it should be processed as described above. Code should be flexible enough to handle N number of generations.

1. Use native PHP / JavaScript
2. Arrange the names
3. Create a html list of the names
4. Functionality should support N number of generations

Calculator

Write a simple car insurance calculator which will output price of the policy using vanilla PHP and JavaScript:

1. Create HTML form with fields:



- Estimated value of the car (100 - 100 000 EUR)
- Tax percentage (0 - 100%)
- Number of instalments (count of payments in which client wants to pay for the policy (1 – 12))
- Calculate button

2. Build calculator logic in PHP using OOP:

- Base price of policy is 11% from entered car value, except every Friday 15-20 o'clock (user time) when it is 13%
- Commission is added to base price (17%)
- Tax is added to base price (user entered)
- Calculate different payments separately (if number of payments are larger than 1)
- Installment sums must match total policy sum- pay attention to cases where sum does not divide equally
- Output is rounded to two decimal places

3. Final output (price matrix):

- Base price
- Price with commission and tax (every instalment separately)
- Tax amount (separately with every instalment)
- Grand totals (sum of all instalments): Price with commission and tax, total tax sum.
- Example with 2 instalments:

	Policy	1 instalment	2 instalment
Value	10000.00		
Base premium (11%)	1100.00	550.00	550.00
Commission (17%)	187.00	93.50	93.50
Tax (10%)	110.00	55.00	55.00
Total cost	1397.00	698.50	698.50

Store employee data

1. Create a database structure to store employee information. The information stored is as follows:

- Name
- Birthdate
- ID code / SSN
- Is a current employee (yes/no)
- Contact information (e-mail, phone, address)

The following information in English, Spanish and French:

- Introduction



- Previous work experience
- Education information

Log info:

- Who and when created the entry
 - Who and when last modified the data
2. The information should be presented as an .sql file which can be imported into a MySQL database without errors.
 3. Write example query to get 1-person data in all languages