Создайте html-форму. Сделайте на форме два поля ввода Date 1 и Date 2. Через поля ввода будут передаваться даты в формате 'год-месяц-день'.

Добавьте на форму два элемента типа radio. Задайте элементы label для этих кнопок: dd.mm.YY и YY.mm.dd соответственно. Радио кнопки будут отвечать за формат даты, в котором будут выводится даты пользователю на экран.

Данные, которые передаются формой либо через адресную строку, необходимо проверять на корректность (валидировать) на сервере. Валидацию можно проводить единожды и затем использовать для решения задач.

По клику на кнопку sumbit вам необходимо выполнить следующие задачи:

1. Пользователь вводит число, а скрипт определяет високосный ли год. Сделать проверку на формат и количество введенных значений. Если есть ошибка - уведомить об этом пользователя.
2. Пользователь передает две даты. Первую дату запишите в переменную $date1, а вторую в $date2. Сравните, какая из введенных дат больше. С помощью функций explode и mktime переведите большую дату в формат timestamp. По этому timestamp узнайте день недели (словом, латиницей) и выведите его на экран.
3. В поле вводится дата. Прибавьте к этой дате 1 год 2 месяца и 3 дня. Отнимите от этой даты 5 дней. Результат преобразуйте ее в выбранный формат и отобразите на экране.
4. С помощью адресной строки передайте GET-параметр date, который хранит год (4 числа). Посчитайте сколько воскресений в введенном году приходится на первое число месяца.
5. С помощью адресной строки передайте GET-параметр date, который хранит год (4 числа). Найдите все пятницы 13-е в этом году. Результат выведите в виде списка дат.
6. Дан GET-параметр date, в который вводится год. Узнайте, какой это будет год по восточному календарю. Результат выведите на экран.
7. Пользователь в форму вводит дату, например, 2017-01-01. Найдите количество дней, часов, минут, секунд, прошедших с 2017-01-01 23:59:59 до настоящего момента.
8. Пользователь в форму вводит дату. Узнайте какой день недели был 100 лет назад.
9. Создайте html-форму. Добавьте на форму 4 элемента с типом checkbox, сгруппированных с помощью элемента fieldset. Чекбоксы должны иметь названия: html, css, php, javascript. Названия чекбоксам задаются с помощью элемента label. Спросите у пользователя, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь. Если пользователь не отметил ни один язык — выведите на экран сообщение об этом.
10. Создайте html-форму. Спросите у пользователя знает ли он PHP с помощью двух radio-кнопок. Выведите результат на экран. Сделайте так, чтобы по умолчанию один из вариантов был уже отмечен.
11. Создайте html-форму. Спросите у пользователя его возраст с помощью нескольких radio-кнопок, сгруппированных элементом fieldset. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30. Результат выдайте на экран в видет “Ваш возраст в диапазоне <n> лет”.
12. Создайте html-форму. Спросите у пользователя его возраст с помощью select. Варианты ответа сделайте такими: менее 20 лет, 20-25, 26-30, более 30.
13. Создайте html-форму. Спросите у пользователя с помощью элемента multiselect, какие из языков он знает: html, css, php, javascript. Выведите на экран те языки, которые знает пользователь.
14. Сделайте функцию, которая создаёт html элемент. Функция должна иметь следующие параметры: type, name, value, placeholder. В функцию на вход может попасть только input или textarea. В любом другом случае необходимо вывести предупреждение об ошибке.
15. Сделайте функцию, которая будет создавать селект. Функция должна принимать многомерный массив, например:

$arr = array(

0=>array('value'=>'php', 'text'=>'Язык PHP'),

1=>array('value'=>'html', 'text'=>'Язык HTML'),

)