



EJERCICIOS PHP

RELACIÓN II - FORMULARIOS, ESTILO Y VALIDACIÓN

1- Crea un formulario de entrada para una calculadora en PHP a partir de dos números enteros y un operador. Para la introducción de datos, utilizaremos dos campos de texto y un select que contenga como opción diferentes operadores : +, -, *, /, %. Probaremos el envío mediante los métodos GET y POST y apreciaremos las diferencias.

Repasa en w3schools el formato de un formulario en html y de los campos más habituales

PUNTOS DE INTERÉS: formularios y campos html (repaso), métodos get y post, captación de datos, controles html de validación

INTRODUCIENDO BOOTSTRAP 5

2- Vamos a introducir el uso de un framework de CSS, concretamente, de **Bootstrap 5**, para formatear el frontend. Retoma el archivo index.html de la relación 1, y aplica un estilo personalizado a tu gusto con Bootstrap 5.

3- Vamos a realizar lo propio con esta relación, y un super-index para todas las relaciones que vamos a realizar. El "super-index" tendrá una barra de navegación, una **card** por cada relación y un footer. Formatea con **Bootstrap** a tu gusto, experimentando diversas posibilidades hasta dar con tu preferida.

4- Retoma el ejercicio 4 de la relación anterior y aplícale estilos con clases de **Bootstrap 5** (experimentaremos así las listas numeradas)

5- Haz lo propio con el ejercicio 5 de la relación anterior, y el formato tabla.

Añade posteriormente, y solo a modo de prueba, algunos **components** como **buttons, spinners, lists**, etc...

6- Crea una tabla de datos, simulando un listado de personas (nombre, apellido, correo, teléfono, etc...) en PHP con formateo de **Bootstrap 5**. Experimenta con varios de ellos y quédate con el que prefieras.

Se recomienda utilizar la herramienta Mockaroo para generar los datos de prueba

7- Monta en un documento PHP un formulario para introducir 2 números formateado con el framework **Bootstrap 5**

8- Retoma el ejercicio 1 de esta relación y aplícale un formateo más estético con **Bootstrap 5**

Experimenta la de validación html. Explora sus funcionalidades, defectos y característica

9- Ahora vamos a añadir el cálculo en el propio documento, para lo que habrá que incluir un control de si ingresamos en él para rellenar los datos, o como resultado del envío del formulario.

PUNTOS DE INTERÉS: formularios y proceso en un mismo archivo de código (no recomendado), uso de `$_SERVER`, `$GET` y `$_POST`, instrucción `match` (desde la versión 8 de PHP en adelante)

10- Haz otra versión que separe en un documento html la captación de datos, y en otro, el cálculo y la edición de resultados. Hazlo así en los siguientes ejercicios hasta que se indique otra cosa.

Validaremos los datos solo con controles html en el formulario, por ahora

PUNTOS DE INTERÉS: comunicación entre archivos diferentes PHP vía `$_GET` y `$_POST`, SoC

11- Re-formatea con **Bootstrap** el ejercicio 7 de la relación anterior y experimenta validaciones de los datos de entrada con html

PUNTOS DE INTERÉS: comunicación entre archivos diferentes PHP vía `$_GET` y `$_POST`, SoC

12- Retoma el ejercicio 8 de la relación anterior y añade el nombre y el correo de la persona a la cual se le va a calcular la calificación. Cambia las notas por datos de entrada (inicial, primera, segunda y tercera).

Formatea con **Bootstrap** y valida los datos de entrada con html. Prueba con datos de entrada erróneos.¿crees que es suficiente con este control?

13- Vuelve a versionar el ejercicio 8 de la relación anterior, pero en este caso, valida utilizando técnicas de **Vanilla JavaScript**. Investiga sobre validaciones JS

PUNTOS DE INTERÉS: eventos JS (onclick, ondoubleclick, onmouseover, onmouseout, onchange, onsubmit...). Funciones de comprobación de tipos, función trim, batería de tests para validación de campos numéricos. Manipulación de estilo desde JS, Refactorización y función addEventListener

14- Haremos lo mismo que en el caso anterior pero con el ejercicio 12 de la relación 1, y acompañaremos la salida numérica con una salida gráfica utilizando una **progress bar**

PUNTOS DE INTERÉS: los mismos del ejercicio anterior, pero se manda para casa, el alumnado DEBE reproducir todo el proceso de:

- 1-creación de un formulario con bootstrap
- 2-validación de los datos de entrada con Vanilla JS
- 3-envío de datos a un programa php que realice los cálculos
- 4- utilizar una salida formateada bootstrap con el component progress bar como elemento gráfico

15- Retomamos el ejercicio 13 de la relación anterior, formateamos la entrada con **Bootstrap**, la validamos con **JavaScript**, pero utilizaremos un mismo documento para el formulario, el cálculo y la visualización de resultados (juntaremos, por tanto, funcionalidades propias del front y del back). ¿Te parece recomendable hacerlo así?

16- Lo mismo con el 15 y 16 de la relación anterior unificándolos en un único programa, pero separando la captación de datos del proceso. En el formulario de entrada se nos pedirá qué queremos, si dictaminar únicamente si el número de entrada es primo o no ó por el contrario, mostrar los divisores (solo podremos elegir una de las dos opciones)

PUNTOS DE INTERÉS: utilizamos la “plantilla” creada en el ejercicio 7 para la entrada de datos, añadiéndole campos radiobutton, y validamos los datos numéricos para que sean enteros positivos, y el radiobutton para que haya siempre una opción seleccionada

17- Lo mismo con el 17, pero incluiremos dos casillas de verificación, para que se pueda elegir si se desea calcular el cociente, el resto, o el cociente y el resto (ambas opciones)

PUNTOS DE INTERÉS: similar al ejercicio anterior, salvo que pueden estar seleccionadas como casillas de verificación las dos opciones anteriores (y al menos una de ellas) y el cálculo de MCD y división entera por Euclides pueden ser ejecutados ambos

18- Repetiremos el ejercicio 19 de la relación anterior, pero con formulario formateado en **Bootstrap**

19- Igual con el 20, pero como el programa podrá convertir a binario, octal o hexadecimal, se incluirá en el formulario un campo **select**. Utiliza **match** en lugar de **switch** para la bifurcación múltiple. ¿puedes observar alguna ventaja del uso de match?

PUNTOS DE INTERÉS: un campo de formulario SELECT cerrado, no tiene por qué ser validado, ya que se conocen todos los valores posibles para elegir. Esta forma de captar la selección, solo se adoptará cuando HAYA UNA CLARA OPCIÓN POR DEFECTO, en este caso, la conversión a binario

20- Retocamos el ejercicio anterior, incluyendo en el campo **select** una indicación de "Selecciona operación" para asegurarnos que se despliega y elige opción conscientemente. Hacer lo mismo en el ejercicio de la calculadora

PUNTOS DE INTERÉS: si la elección de una opción en un campo select no tuviera una opción clara "por defecto" (caso general), habría que incluir una falsa opción: "elige opción", asociarle un VALUE impropio (por ejemplo, 0 en este caso), y validar que no se envíe este valor, sino uno de los permitidos: 2, 8, 16. De esta forma estaremos seguros de que la opción ha sido elegida conscientemente y no por error/descuido