Ejercicios UT2 – Arrays

Normas para todos los ejercicios que se hagan

- todos los proyectos incluirán una hoja de estilos (la que se da u otra personal)
- las páginas web mantendrán una división lógica a través de elementos div
- el código HTML generado será válido
- el código PHP será legible (nombres de variables descriptivos, indentación correcta, ….)

Objetivos

- Saber trabajar con arrays asociativos.
- Utilizar los arrays para realizar validaciones.
- Saber realizar validaciones de campos a través de las reglas de validación establecidas en un array asociativo.

Ejercicio 1.

Abre el proyecto arraysReglas1_AL. Incluye un nuevo archivo php registro.php. Incluye la cabecera y el pie. Crea un array asociativo $personas con el nombre, apellidos y email de 4 personas tal que al crearlo y visualizarlo con la función verArray el resultado sea el siguiente:

```
Array
(
    [0] => Array
        (
            [nombre] => Pedro
            [apellidos] => Huici
            [mail] => pe.hu@gmail.com
        )

    [1] => Array
        (
            [nombre] => Ana
            [apellidos] =>
            [mail] => an.be@gmail.com
        )
```

Van a representar registros de personas. Para las otras dos personas incluye tus datos y el de un compañero. Un campo vacío se representa con el valor **null.**

El ejercicio simulará validaciones de campos, por ejemplo de un formulario, para ello se debe cambiar en cada ejecución los valores del array. Haciendo que estos en alguna ejecución no tengan valor o estén erróneos para probar el programa.

El programa validará que los campos nombre, apellidos y email tengan valor y que el email contenga la @. Para ello se llamará a una función validar1 con cada una de las personas y retornará un array de errores con la información de los errores que se han producido.

```
)
```

CORRECTA

```
Array
(
    [nombre] => Pedro
    [apellidos] => Huici
    [mail] => pe.hu@gmail.com
)
```

INCORRECTA

```
Array
(
    [apellido] => Introduzca los apellidos
)
```

Ejercicio 2.

Abre el proyecto arraysReglas1_AL. Analiza el código. Los comentarios del proyecto te indicarán los pasos a seguir.
En el array $reglasValidacion se incluyen reglas del tipo valores requeridos, en formato (required=>true)
En el array $camposValidar se va a dar en cada ejecución valores a los campos nombre, apellidos y edad.
En el caso de querer dar el valor vacío se debe utilizar null.

El programa:

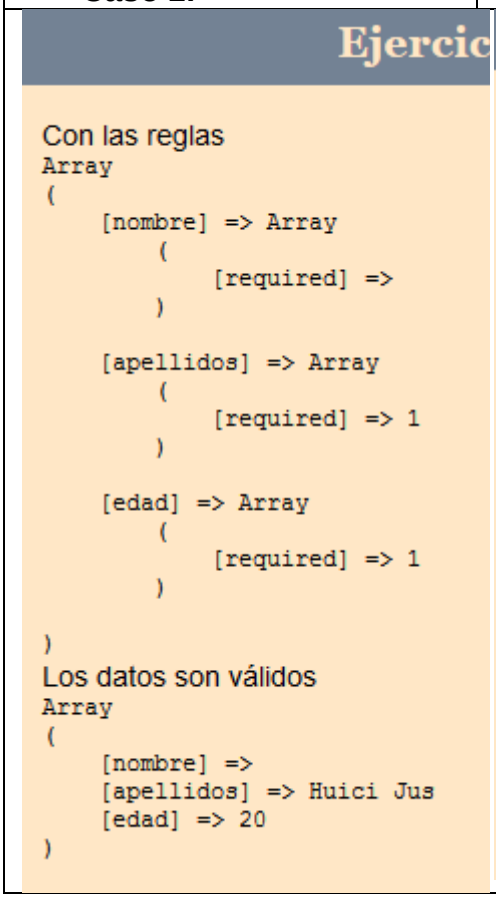
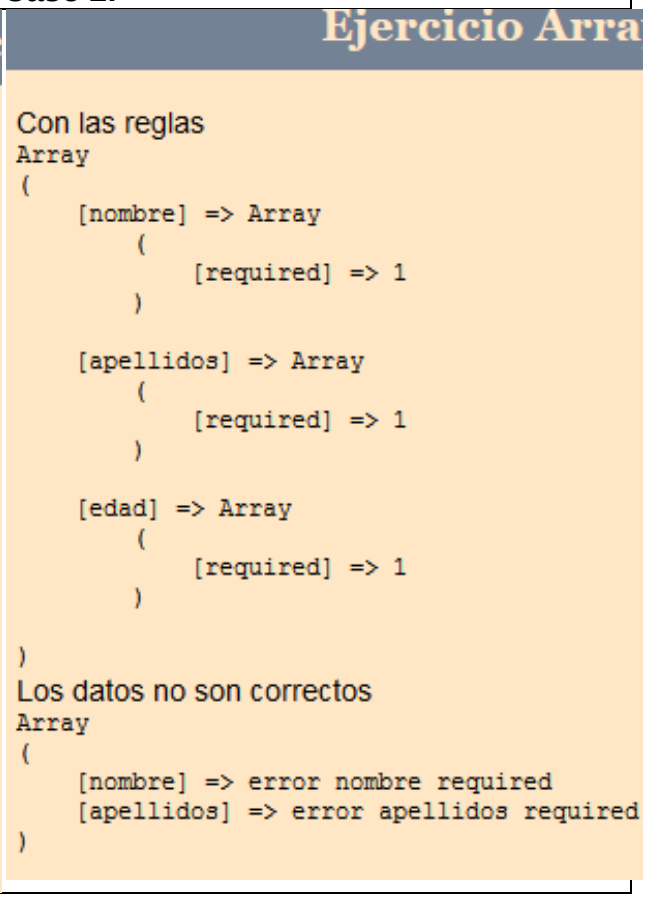
1. Crea las reglas.
2. Da valores a los campos.
3. Se Llama a la función validar2.
4. La función validar2 devuelve en este caso los errores que se produzcan según las reglas establecidas. Visualiza los errores según el ejemplo. Utiliza la función verArray()

Caso 1: Datos válidos según las reglas

Reglas	Valores
nombre opcional	nombre=null
apellidos required	apellidos= Huici Jus
edad required	edad=20

Caso 2: Datos incorrectos según las reglas

Reglas	Valores
nombre required	nombre=null
apellidos required	apellidos= null
edad required	edad=20

Caso 1:	Caso 2:
	

Ejercicio 2.

Incluye una nueva regla de validación en la edad. La edad debe ser mayor que un tope (en el ejemplo 20).

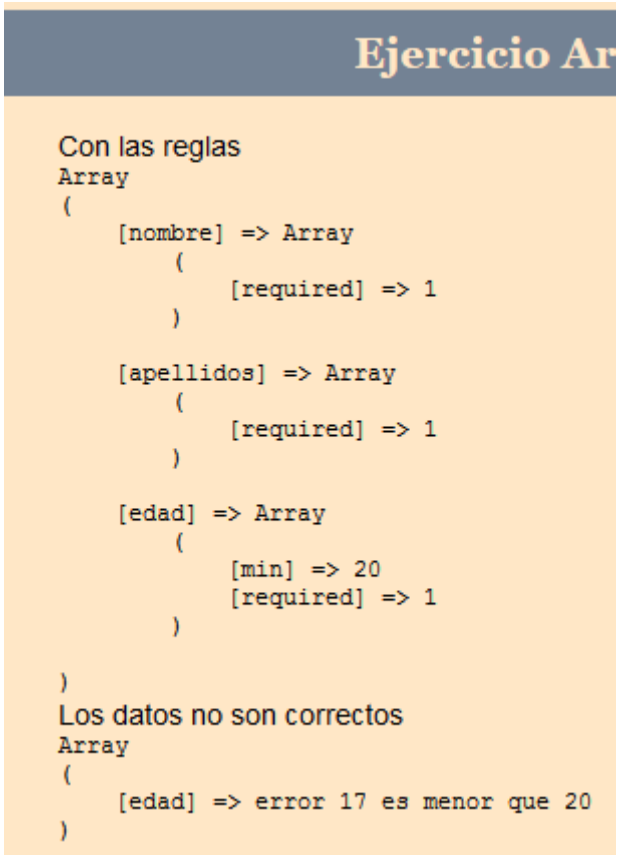
Caso 1: Datos válidos según las reglas

Reglas	Valores
nombre opcional	nombre=null
apellidos required	apellidos= Huici Jus
edad required	edad=21

Caso 2: Datos incorrectos según las reglas

Reglas	Valores
nombre required	nombre=Ane
apellidos required	apellidos= Zazpi Lore
edad required	edad=17

Caso2:


Ejercicio Ar
Con las reglas
Array
(
[nombre] => Array
(
[required] => 1
)
[apellidos] => Array
(
[required] => 1
)
[edad] => Array
(
[min] => 20
[required] => 1
)
)
Los datos no son correctos
Array
(
[edad] => error 17 es menor que 20
)