

BLOQUE Nº3.

programación Web del lado del cliente

Ejercicio 2

Para este ejercicio hay que añadir CSS a la web app realizada en el anterior bloque.

Primero es necesario quitar todas las etiquetas de la aplicación, en mi caso no tenía ninguna.

Lo segundo es añadir estilo para los textos, mensajes de error, etc.

Vamos a empezar por la página de login:

```
body {  
  background: -webkit-linear-gradient(bottom, #8932f6, #b37bf7);  
  background-repeat: no-repeat;  
  height: 100% !important;  
}
```

Primero tenemos el body, en el cual de fondo he puesto un gradiente que no se repite.

Como div principal tenemos #div:

```
<div id="card">
```

```
#card {  
  background: #fbfbfb;  
  border-radius: 8px;  
  box-shadow: 1px 2px 8px rgba(0, 0, 0, 0.65);  
  height: auto;  
  margin: 6rem auto 8.1rem auto;  
  width: 329px;  
}
```

Donde ponemos como color de fondo blanco, un radio al borde para que tenga los bordes redondeados y una sombra negra. También ponemos que la altura sea automática, con un poco de margen para que haya espacio entre el contenido de #card y el borde. Además de ponerle una anchura predeterminada

La siguiente div que tenemos es #card-tittle donde se puede ver el título de la página y tenemos lo siguiente:

```

<div id="card-title">
  <h2>LOGIN</h2>
  <div class="underline-title"></div>
</div>

```

La div con clase underline-tittle se usa para poner una línea bajo el titulo

```

#card-title {
  font-family: "Raleway Thin", sans-serif;
  letter-spacing: 4px;
  padding-bottom: 23px;
  padding-top: 13px;
  text-align: center;
}

```

Donde cambiamos la Font del texto, hacemos que haya un poco de espacio entre las letras y que haya espacio arriba y debajo de este. Además, lo centramos en #card.

Ahora toca el formulario de inicio de sesión:

```

<form action="{pageContext.request.contextPath}/login" method="post" class="form" id="formulario">
  <label for="username" style="text-align: center;">Usuario </label>
  <input type="text" id="username" name="username" class="form-content" required>
  <div class="form-border"></div>
  <label for="password">Contraseña </label>
  <input type="password" id="password" name="password" class="form-content" required>
  <div class="form-border"></div>
  <a href="{pageContext.request.contextPath}/recuperarPassword"><legend id="forgot-pass">Recuperar contraseña</legend>
  </a>
  <input id="submit-btn" type="submit" name="login" id="login" value="LOGIN" />
</form>

```

Donde tenemos el típico formulario, la única diferencia es que bajo cada input tenemos div como con el título para mostrar una línea bajo estos.

```

.form {
  align-items: left;
  display: flex;
  flex-direction: column;
}

```

En el formulario aplicamos que los objetos se peguen a la izquierda. Además, usamos como display flex en dirección de columna. Flex nos permite mostrar los elementos de dos maneras, en forma de columna o de fila, para este caso usamos el de columna. Además, usando flex nos permite que se coloquen automáticamente a la misma altura o definir que si hay varios objetos, estos se vayan distribuyendo a lo largos de columnas o filas necesarias.

```

.form-content {
  background: #fbfbfb;
  border: none;
  outline: none;
  padding-top: 14px;
}

```

A cada input le ponemos eso, para que la caja para introducir texto sea invisible. Le ponemos fondo blanco y le quitamos tanto el borde como la línea que aparece al hacerle focus.

```

.form-border {
  background: -webkit-linear-gradient(right, #a6f77b, #2ec06f);
  height: 1px;
  width: 100%;
}

```

A las líneas que hay debajo de cada input le aplicamos ese estilo. Con ese hacemos que tenga un gradiente, la altura de 1px y toda la anchura que pueda. De este manera nos queda una línea fina.

```

#submit-btn {
  background: -webkit-linear-gradient(right, #a6f77b, #2dbd6e);
  border: none;
  border-radius: 21px;
  box-shadow: 0px 1px 8px #24c64f;
  cursor: pointer;
  color: white;
  font-family: "Raleway SemiBold", sans-serif;
  height: 42.3px;
  margin: 0 auto;
  margin-top: 25px;
  transition: 0.25s;
  width: 153px;
}

#submit-btn:hover {
  box-shadow: 0px 1px 18px #24c64f;
}

```

Por último tenemos el botón de submit. Primero, le aplicamos un gradiente igual al resto de cosas, le quitamos el borde y le ponemos un tamaño al radio del borde para que sea redondo.

Lo siguiente es darle un poco de sombra y hacer que la letra sea blanca con una nueva tipografía. Además, le damos cierto tamaño y márgenes para que quede mejor. También tenemos transition, que nos deja poner cuanto tarda en cambiar entre estados.

Con :hover, podemos darle estilo cuando pasemos el ratón sobre este. Hacemos una box-shadow, para que cuando pasemos el ratón sobre el botón se vea una sombra alrededor.

Lo último que tenemos es un link a la página de recuperar contraseña con el siguiente estilo:

```
#forgot-pass {  
    color: #2dbd6e;  
    font-family: "Raleway", sans-serif;  
    font-size: 10pt;  
    margin-top: 3px;  
    text-align: right;  
}
```

Con el que simplemente lo cambiamos de color, su tamaño y Font, y lo pegamos a la derecha.

Lo siguiente que tenemos en login es:

```
<hr>  
<div id="links">  
    <button class="boton" id="boton1" name="boton1" type="button">Olvidé mi contraseña</button>  
    <button class="boton" id="boton2" name="boton2" type="button">Registrarme</button>  
</div>
```

Aquí tenemos un par de botones para ir a la página de registro y acceder sin cuenta con el siguiente estilo:

```

#links .boton {
    background: -webkit-linear-gradient(right, #a6f77b, #2dbd6e);
    border: none;
    border-radius: 21px;
    box-shadow: 0px 1px 8px #24c64f;
    cursor: pointer;
    color: white;
    font-family: "Raleway SemiBold", sans-serif;
    height: 42.3px;
    margin: 0 auto;
    margin-top: 25px;
    transition: 0.25s;
    width: 153px;
}

#links .boton:hover {
    box-shadow: 0px 1px 18px #24c64f;
}

#links{
    justify-content: center;
    align-items: center;
    text-align: center;
}

```

El estilo de los botones es igual al del submit del formulario, solo se ha cambiado un poco el tamaño. Al final tenemos un selector para el div completo, con el que alineamos todo el contenido al centro.

Al final de cada página hay que poner un footer:

```

<footer class="autor">
    <p>David Navarro Jimena (Universidad de Malaga)</p>
    <p>Copyright 2023 Todos los derechos reservados</p>
</footer>

```

El cual tiene el siguiente estilo:

```

.autor {
  z-index: 1;
  position: fixed;
  left: 0px;
  bottom: 0px;
  height: auto;
  width: 100%;
  background: #ff0202;
}

```

Con z-índex podemos hacer que este sobre el resto de los elementos y con position fixed hacemos que nunca se mueva. Con left y bottom lo colocamos en la parte de abajo. Además, le damos una altura automática, una anchura del 100% de la página y un fondo rojo para que se vea.

También es necesario darles estilo a los links, he usado esto:

```

a {
  color: #24c64f;
}

a:link{
  color: red;
}

a:visited{
  color:purple;
}

a:hover{
  color:pink;
  text-decoration: none;
}

a:active{
  color: brown;
}

```

De esta manera, modificamos todo. Con el primero modificamos el color base. Con el segundo, modificamos el enlace cuando no lo hemos visitado. Con el tercero en caso de haberlo visitado. Con el cuarto cuando pasamos el ratón sobre él. Y en el último caso cuando está activo.

Lo siguiente sería explicar las páginas de registro y recuperar contraseñas. Estas páginas son iguales a la de login, el único cambio es un poco en el tamaño de un par de cosas. Así que voy a explicar directamente la página principal.

```
<header>
  
<nav>
  <%if((String)session.getAttribute("loginName")!=null){%>
    <pre>Hola, ${loginName} <a href="${pageContext.request.contextPath}/logout">Logout</a></pre>
  <%}%>
</nav>
```

Lo primero que hay es un header donde tenemos un logo, y además aparece el nombre de usuario y un botón para cerrar sesión si el usuario entra con su usuario.

```
header{
  font-size: 25px;
  display: flex;
  flex-direction: row;
  align-items: center;
  font-family: "Raleway", sans-serif;
  letter-spacing: 4px;
  padding-bottom: 23px;
  padding-top: 13px;
}
```

A ese header le aplicamos ese estilo, con el cual cambiamos el tamaño de la letra. Usamos el flex en este caso para mostrar tanto el logo como el nombre en la misma línea. También cambiamos el espacio entre las letras y su Font para que quede mejor. Por último, añadimos un poco de espacio al final para separarlo un poco del resto de la página.

```
<div class="buscar">
  <form action="index.jsp?pag=1">
    <input type="text" id="titulo" name="titulo" placeholder="Titulo" class="form-content">
    <input type="text" id="autor" name="autor" placeholder="Autor" class="form-content">
    <button type="submit">Search</button>
  </form>
</div>
```

Lo siguiente que nos encontramos es el formulario de búsqueda de libros que tiene el siguiente estilo:

```
.buscar{
  padding-left: 10px;
  font-size: 25px;
  padding-bottom: 20px;
}
```

Primero colocamos la div un poco para que quede bien

```

7. buscar .form-content {
    --color-debajo : -webkit-linear-gradient(right, #a6f77b, #2ec06f);
    background: transparent;
    border: none;
    border-bottom: 1px solid;
    outline: none;
    border-image-slice: 1;
    border-image-source: -webkit-linear-gradient(right, #a6f77b, #2ec06f);
    width: 200px;
}

7. buscar .form-border {
    background: -webkit-linear-gradient(right, #a6f77b, #2ec06f);
    height: 1px;
    width: 200px;
}

```

Tras esto modificamos los input para que queden como en el login. Colocar la línea como antes era más complicado, así que opto por usar border-image, gracias al cual podemos poner un gradiente de borde. Además, tenemos que colocar que solo haya borde abajo y que el fondo sea transparente para que no haya ningún tipo de caja de input.

Y el botón de input le doy el mismo estilo que a los de login.

Lo siguiente que tenemos en la página es una tabla que muestra todos los libros, a este le daremos el siguiente estilo:

```

table{
    border-radius: 10px;
    background-color: white;
    text-align: left;
    border-collapse: collapse;
    width: 100%;
}

```

Primero colocamos el texto a la izquierda y le ponemos un borde

```

th, td{
    padding: 20px;
}

```

A cada fila y columna le metemos un padding para que haya espacio entre ellas


```
tr:nth-child(even){
    background-color: #6e6d6d;
}

tr:nth-child(odd){
    background-color: #868686;
}
```

Con este código, hacemos que las filas pares tengan un color y que las impares tengan otro. De esta manera le damos color automáticamente sin importar el tamaño de la tabla.

```
<div class="pag">
  <%if(pag>1){%>
    <button onclick="window.location.href='index.jsp?pag=<%=pag-1%>&titulo=<%=titulo%>&autor=<%=autor%>' ">Anterior</button>
  <%}else{%>
    <button disabled="disabled">Anterior</button>
  <%}%>
  <%if(pag*5>libros.size()){%>
    <button disabled="disabled">Siguiente</button>
  <%}else{%>
    <button onclick="window.location.href='index.jsp?pag=<%=pag+1%>&titulo=<%=titulo%>&autor=<%=autor%>' ">Siguiente</button>
  <%}%>
  <%for(int j = 1; j<=controlLibros.busquedaLibrosjson(titulo, autor).size()/5; j++){
    if(pag==j){%>
      <span><%=pag%></span>
    <%}else{%>
      <a href="index.jsp?pag=<%=j%>&titulo=<%=titulo%>&autor=<%=autor%>"><%=j%></a>
    <%}%>
  <%}%>
  <%>
</div>
```

Lo siguiente que tenemos es mostrar el número de páginas y los botones para poder movernos entre las páginas.

Los botones tienen el mismo estilo que el resto de los botones y los enlaces lo mismo.

A la div le damos el siguiente estilo:

```
.pag{
    font-size: 35px;
    align-items: center;
    text-align: center;
}
```

Con lo que hacemos más grande el tamaño de la letra y ponemos todos los botones en el centro.

```
<div style="...">
  <button onclick="window.print()" >Imprimir</button>
</div>
```

Lo último antes del footer es un botón de print, al cual le damos el mismo estilo que al resto de botones y lo centramos.

Con esto va todo el estilo de la web, en el siguiente ejercicio habrá capturas de la web.

Ejercicio 4

Para este ejercicio tenemos que añadir javascript a la página creada anteriormente.

Lo primero que hay que hacer es añadir validación de formularios, lo cual se podría hacer con el siguiente código:

```
document.addEventListener( type: "DOMContentLoaded", listener: function() {  
    document.getElementById( elementId: "formulario").addEventListener( type: 'submit', validarFormulario);  
});  
  
function validarFormulario(evento) {  
    evento.preventDefault();  
    var usuario = document.getElementById( elementId: 'username').value;  
    if(usuario.length < 3) {  
        alert('No has escrito nada en el o es demasiado corto');  
        return;  
    }  
    var clave = document.getElementById( elementId: 'password').value;  
    if (clave.length < 3) {  
        alert('La clave no es válida');  
        return;  
    }  
    this.submit();  
}
```

Con lo primero, lo que hacemos es enlazar el formulario con una función, por lo que al pulsar submit se lanzaría también la función.

Abajo tenemos la función, a esta le pasamos la variable evento que contiene información sobre lo que la ha lanzado. Primero haces preventDefault(), lo cual hace que se prevenga la ejecución del comportamiento por default del evento. En este caso, hacemos que no se envíe el formulario. Tras esto, usando getElementById conseguimos el valor de cada uno de los campos. Para ambos simplemente he puesto como restricción que tenga un tamaño mínimo. En caso de que fueran más pequeño que esto, se lanzaría una alerta mostrando un mensaje y no se enviaría el formulario. Si se cumple, enviamos el formulario.

Para registro, he añadido más campos, en este caso un segundo campo de contraseña, email, teléfono, nombre y apellido. Primero añadiríamos un eventListener como antes y para comprobar los valores uso lo siguiente:

```
function validarFormulario(evento) {
    evento.preventDefault();
    var usuario = document.getElementById( elementId: 'username').value;
    if(usuario.length < 3) {
        alert('El usuario es demasiado corto');
        return;
    }
    var clave = document.getElementById( elementId: 'password').value;
    if (clave.length < 3) {
        alert('La clave no es válida');
        return;
    }

    var clave2 = document.getElementById( elementId: 'password2').value;
    if (!clave.value===clave2.value) {
        alert('Las contraseñas no coinciden');
        return;
    }
}
```

Antes de nada, hacemos que no se envíe el formulario y vamos comprobando campo por campo. Primero comprobamos el usuario y contraseña, que sean más largas de lo que queremos. Tras esto, usando el segundo campo de contraseña, vemos si estas coinciden.

```
var validRegex = /^[a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$/;

var email = document.getElementById( elementId: 'email').value;
if (!email.match(validRegex)) {
    alert('El email no es válido');
    return;
}

var telefono = document.getElementById( elementId: 'telefono').value;
if (telefono.length!=9) {
    alert('El telefono debe ser de longitud 9');
    return;
}

var nombre = document.getElementById( elementId: 'nombre').value;
if(nombre.length < 3) {
    alert('El nombre es demasiado corto');
    return;
}

var apellido = document.getElementById( elementId: 'apellido').value;
if (apellido.length < 3) {
    alert('El apellido es demasiado corto');
    return;
}
}
```

Para comprobar si el email correcto vamos a usar una expresión regular, la cual correspondería con cualquier email. Para el teléfono tenemos la restricción de que tiene que ser de tamaño 9. El resto de comprobación sería igual que el resto. En caso de que no se cumpla alguno, mostramos una alerta con el error y no enviamos el formulario.

Lo siguiente necesario a hacer es cambiar los enlaces por botones. He dejado algunos enlaces para que se vea el estilo de estos, aunque la mayoría los he puesto como botones. Tendrían la siguiente forma:

```
<button class="boton" id="boton1" name="boton1"
onclick="window.location.href='${pageContext.request.contextPath}/register'">
Si no tienes cuenta regístrate</button><br><br>
```

Donde en la opción de onclick del botón, añadimos la función window.location y modificamos el valor de href, de esta manera, al pulsarlo, se actualiza ese valor al que queramos y nos lleva a esa página. Esto lo he hecho con varios enlaces a lo largo de la web app, incluidos los botones de moverse entre páginas, donde hay enlaces para ir a una página concreta y botones para ir a la siguiente o anterior página.

Ahora toca agregar una animación a la web app con imágenes. Para este caso añadí un logo a la página:

```

```

Como se puede, tenemos la opción de onmouseover, que hace una acción cuando un ratón pasa por encima y onmouseout para que pase algo cuando el ratón sale de encima.

```
function hover(img){
    img.src="logo2.png";
    img.width="83.5";
    img.height="50"
}

function unhover(img){
    img.src="logo1.png";
    img.width="160";
    img.height="50"
}
```

Las funciones que tenemos enlazadas a estas son esas. Simplemente, le pasamos la imagen como parámetro, y modificamos sus valores. En este caso, modificamos con img.src la imagen que mostramos y su tamaño con width y height. De esta manera si pasamos el ratón por el logo se mostrará una imagen y el resto del tiempo otra.



Aunque no se vea el ratón en la segunda imagen, sería la que se ve cuando pones el ratón encima.

Por último tenemos que añadir un botón que abra la pantalla de imprimir la página. En la web lo añadimos así:

```
<div style="...">
  <button onclick="window.print()" >Imprimir</button>
</div>
```

Donde onclick ponemos la función de javascript window.print() y ya al pulsar saldría una pantalla para imprimir la página.

Algunas capturas de la página:

 Hola, admin [Logout](#)

Titulo	Autor	Search			
Título	Autor	Enlace	Resumen	Descargas	
El Principito	Antoine de Saint-Exupéry		resumen1blalblalblal	1	
Abuelita	Hans Christian Andersen		resumen1blalblalblal	0	
Alicia en el pais de las maravillas	Lewis Carroll		resumen1blalblalblal	0	
Barbanegra	Daniel Defoe		resumen1blalblalblal	0	
Caperucita Roja	Charles Perrault		resumen1blalblalblal	0	

[Anterior](#) [Siguiente](#) 1 2

[Imprime](#)

David Navarro Jimena (Universidad de Málaga)
Copyright 2023 Todos los derechos reservados

LOGIN

Usuario

Contraseña

[Recuperar contraseña](#)

[LOGIN](#)

[Si no tienes cuenta regístrate](#)

[Accede sin cuenta](#)

David Navarro Jimena (Universidad de Málaga)
Copyright 2023 Todos los derechos reservados

The screenshot shows a registration form titled "REGISTRO" centered on a purple gradient background. The form is a white card with the following fields: "Usuario", "Contraseña", "Repite contraseña", "Email", "Telefono" (with a country code dropdown), "Nombre", and "Apellido". Below these fields is a link "Recuperar contraseña" and a green "SUBMIT" button. At the bottom of the page, a red footer contains the text: "David Navarro Jimena (Universidad de Málaga) Copyright 2023 Todos los derechos reservados".

The screenshot shows a password recovery form titled "RECUPERAR CONTRASEÑA" centered on a purple gradient background. The form is a white card with a "Usuario:" label and a text input field. Below the input field is a green "SUBMIT" button. Further down are three more green buttons: "Si tienes cuenta, haz login aquí", "Si no tienes cuenta, registrate", and "Accede sin cuenta". At the bottom of the page, a red footer contains the text: "David Navarro Jimena (Universidad de Málaga) Copyright 2023 Todos los derechos reservados".

Ejercicio 5

En este ejercicio hay que realizar una página web que transforme:

- Unidades de longitud, de metros a pies y pulgadas, y viceversa.
- Unidades de peso, de gramos a libras y onzas, y viceversa.

Para realizar esto, he hecho dos formularios separados para cada cosa. El primero es para los metros:

```

<form action="" id="pulgadas-form">
  <fieldset>
    <label><input type="radio" name="metrosa" id="metrosa" value="metros" checked>
    Metros a pies y pulgadas</label>
    <label><input type="radio" name="metrosa" id="pulgadasa" value="pulgadas">
    Pies y pulgadas a metros</label>
  </fieldset>
  <label for="metros">Metros:</label>
  <input type="number" name="metros" id="metros" step=".01" value=0>
  <label for="pies">Pies:</label>
  <input type="number" name="pies" id="pies" step=".01" value=0>
  <label for="pulgadas">Pulgadas:</label>
  <input type="number" name="pulgadas" id="pulgadas" step=".01" value=0>
  <input type="submit" value="Convertir" name="pulgadas-submit" id="pulgadas-submit">
</form>

```

Primero, he hecho un fieldset con dos input de tipo radio, de esta manera puedes elegir si quieres transformar de metros a pies y pulgadas o al revés. Tras esto tenemos tres inputs para cada uno de los tipos de unidades. Estas son de tipo número y con step hacemos que los números puedan tener como mucho dos decimales.

Para hacer funcionar esto, tenemos que enlazar el hecho de hacer submit del formulario con la función de javascript para transformar las unidades. Esto lo hacemos de esta manera:

```

document.addEventListener("DOMContentLoaded", function(){
  document.getElementById("pulgadas-form").addEventListener("submit", pulgadas);
});

```

Así hacemos que cuando se haga submit del formulario, se lance la función que queramos.

Y la función para transformación usamos:

```
function pulgadas(event){
    event.preventDefault();
    var radio = document.querySelector('input[name=metros]:checked').value;
    if(radio=='metros'){
        var pies = document.getElementById("metros").value*(39.37);
        document.getElementById("pulgadas").value=(pies-Math.trunc(pies/12)*12).toFixed(2);
        document.getElementById("pies").value=Math.trunc(pies/12);
    }else{
        var inches = (parseInt(document.getElementById("pies").value)*12+parseInt(document.getElementById("pulgadas").value));
        document.getElementById("metros").value=(inches/39.37).toFixed(2);
    }
}
```

Primero tenemos evento.preventDefault(), con esto hacemos que no se ejecute la función por defecto del elemento, que en este caso sería hacer submit ya que es un formulario.

Tras esto, conseguimos el valor de los input de tipo radio, para este caso como solo podemos tener uno seleccionado, nos vale con conseguir ese.

Si ese valor es igual a metros, transformamos de metros a pulgadas, en caso de que no, hacemos lo contrario.

Para hacer esto, conseguimos los valores que necesitemos, hacemos los cálculos, buscamos los campos que queramos y cambiamos el valor de este por el calculado.

Para las unidades de peso, sería más de lo mismo pero simplemente cambiamos los cálculos como se puede ver:

```
<form action="" id="libras-form">
  <fieldset>
    <label><input type="radio" name="gramosa" id="gramosa" value="gramos" checked>
    Gramos a libras y onzas</label>
    <label><input type="radio" name="gramosa" id="librasa" value="libras">
    Libras y onzas a gramos</label>
  </fieldset>
  <label for="gramos">Gramos:</label>
  <input type="number" name="gramos" id="gramos" step=".01" value=0>
  <label for="libras">Libras:</label>
  <input type="number" name="libras" id="libras" step=".01" value=0>
  <label for="onzas">Onzas:</label>
  <input type="number" name="onzas" id="onzas" step=".01" value=0>
  <input type="submit" value="Convertir" name="libras-submit" id="libras-submit">
</form>
```

```
document.addEventListener("DOMContentLoaded", function(){
    document.getElementById("libras-form").addEventListener("submit", libras);
});
```

```
function Libras(event){
    event.preventDefault();
    var radio = document.querySelector('input[name=gramosa]:checked').value;
    if(radio=='gramos'){
        var gramos= document.getElementById("gramos").value/(453.592);
        document.getElementById("onzas").value=((gramos-Math.trunc(gramos))*16).toFixed(2);
        document.getElementById("libras").value=Math.trunc(gramos);
    }else{
        var libras = (parseInt(document.getElementById("onzas").value)/16+parseInt(document.getElementById("libras").value));
        document.getElementById("gramos").value=(libras*453.592).toFixed(2);
    }
}
```


Esta página se puede cargar directamente en el navegador y se vería así:

<input checked="" type="radio"/> Metros a pies y pulgadas <input type="radio"/> Pies y pulgadas a metros			
Metros: 0	Pies: 0	Pulgadas: 0	Convertir
<input checked="" type="radio"/> Gramos a libras y onzas <input type="radio"/> Libras y onzas a gramos			
Gramos: 0	Libras: 0	Onzas: 0	Convertir

Si introducimos 2 metros por ejemplo y pulsamos convertir veríamos lo siguiente:

<input checked="" type="radio"/> Metros a pies y pulgadas <input type="radio"/> Pies y pulgadas a metros			
Metros: 2	Pies: 6	Pulgadas: 6,74	Convertir
<input checked="" type="radio"/> Gramos a libras y onzas <input type="radio"/> Libras y onzas a gramos			
Gramos: 0	Libras: 0	Onzas: 0	Convertir

O si queremos cambiar de pies y pulgadas a metros:

<input type="radio"/> Metros a pies y pulgadas <input checked="" type="radio"/> Pies y pulgadas a metros			
Metros: 0,91	Pies: 3	Pulgadas: 0	Convertir
<input checked="" type="radio"/> Gramos a libras y onzas <input type="radio"/> Libras y onzas a gramos			
Gramos: 0	Libras: 0	Onzas: 0	Convertir

Sería tan simple como cambiar el valor de los inputs de tipo radio y poner los valores.

Para las unidades de peso seria lo mismo:

<input type="radio"/> Metros a pies y pulgadas <input checked="" type="radio"/> Pies y pulgadas a metros			
Metros: 0,91	Pies: 3	Pulgadas: 0	Convertir
<input checked="" type="radio"/> Gramos a libras y onzas <input type="radio"/> Libras y onzas a gramos			
Gramos: 500	Libras: 1	Onzas: 1,64	Convertir

<input type="radio"/> Metros a pies y pulgadas <input checked="" type="radio"/> Pies y pulgadas a metros			
Metros: 0,91	Pies: 3	Pulgadas: 0	Convertir
<input type="radio"/> Gramos a libras y onzas <input checked="" type="radio"/> Libras y onzas a gramos			
Gramos: 2409,71	Libras: 5	Onzas: 5	Convertir