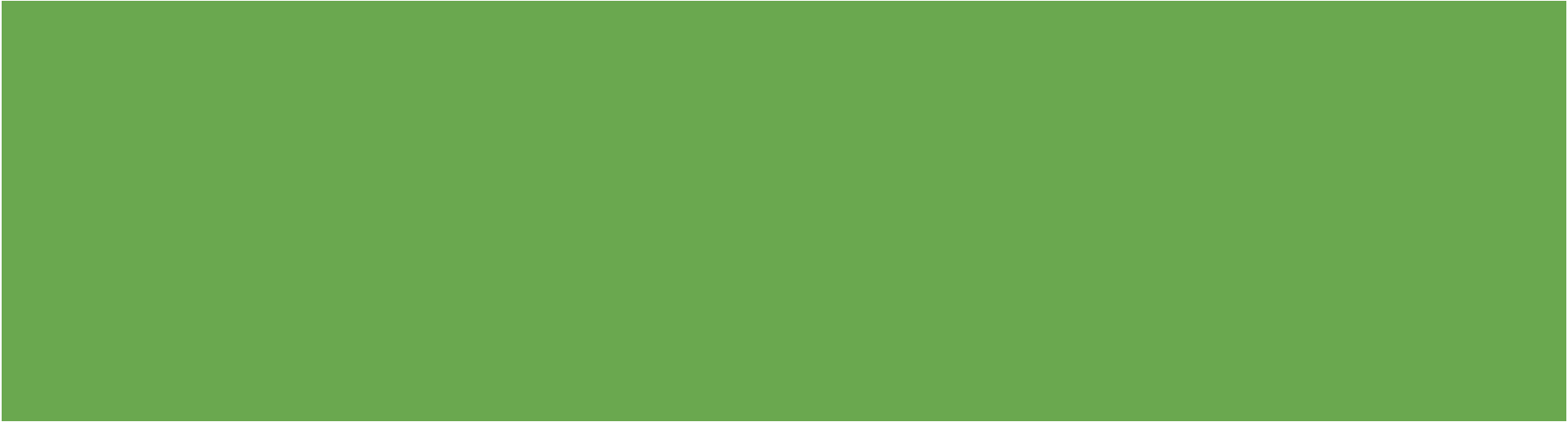


# Tema 3.3

DISEÑO RESPONSIVO: DISEÑO PARA TABLETAS Y ORDENADORES



# Introducción

El enfoque de diseño web responsivo estiliza el contenido de forma diferente según el tipo de dispositivo utilizado para ver el sitio.

El enfoque "**mobile-first**" se centra en estilizar el contenido de forma adecuada para los dispositivos móviles más pequeños. También utiliza un diseño de una sola columna para evitar el desplazamiento horizontal y minimizar el desplazamiento vertical. Este diseño crea una experiencia más agradable para los usuarios de dispositivos móviles.

# Introducción

El proceso de diseño para tabletas y ordenadores de sobremesa permite una mayor **flexibilidad** en el diseño de la página web.

**Un diseño para tableta puede utilizar un diseño de dos columnas, mientras que un diseño de escritorio puede utilizar un diseño de varias columnas.**

# Introducción

El diseño para tabletas y ordenadores de sobremesa se simplifica mediante el uso de *media queries*. Si adoptas un enfoque "mobile-first", utiliza las media queries para crear el aspecto deseado para las ventanas de visualización de distintos tamaños, como las de las tabletas, los ordenadores portátiles o sobremesa.

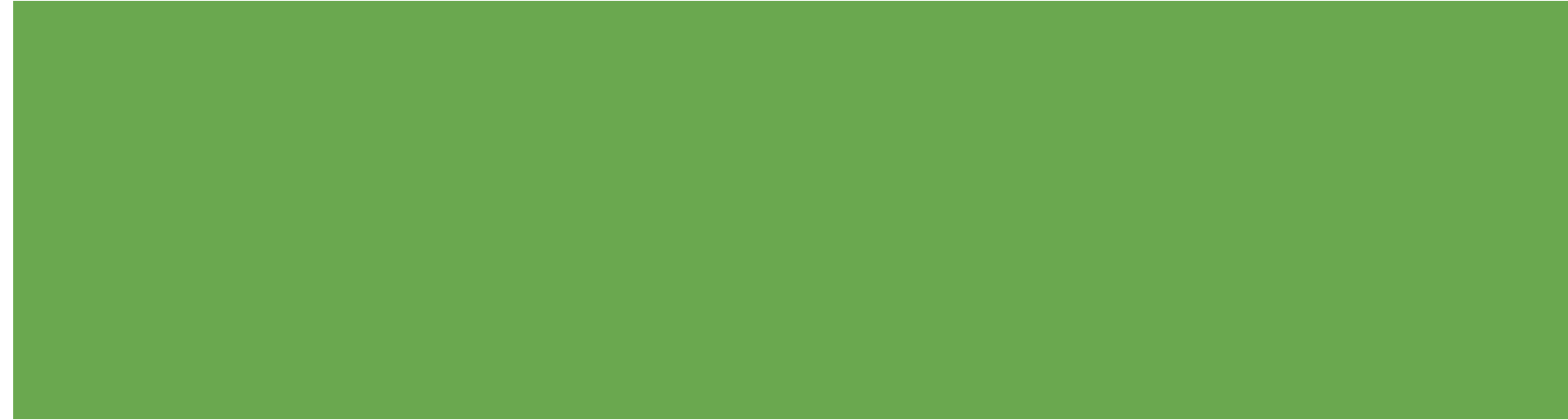


# Roadmap

En esta unidad aprenderemos a realizar las siguientes actividades:

- Añadir media query y estilos para tabletas.
- Añadir media query y estilos para desktop.
- Modificar los breakpoints de la ventana gráfica.
- Insertar y estilizar pseudoclases.
- Añadir un gradiente lineal.

# 1. Media Querys



# 1. Media Querys

Una media query detecta el tipo de medio (como pantalla o impresión) y las capacidades del dispositivo en el que se ejecuta el navegador (como el tamaño de la ventana gráfica en píxeles o su orientación, vertical u horizontal).

Basándose en la información, la media query aplica estilos que funcionan bien para esa ventana gráfica.

# 1. Media Querys

Las medias querys llevan la capacidad de respuesta a un nivel completamente nuevo. Permiten aplicar diferentes reglas de estilo CSS en respuesta al entorno, como el medio en el que se está viendo la página web (pantalla o impresión) o el tamaño de la ventana gráfica.

Las medias querys pueden aplicar estilos para mover, ocultar o mostrar el contenido de la página, cambiar el texto o los colores, o añadir cualquier otro estilo para facilitar la lectura de la página en una situación concreta.



# 1. Media Querys

Las media querys pueden incrustarse en la etiqueta de enlace que conecta una hoja de estilo externa con un archivo HTML, o pueden insertarse en la propia hoja de estilo externa. El siguiente código proporciona un ejemplo básico de una media query insertada en la etiqueta de enlace de una página HTML:

```
<link rel="stylesheet" href="css/styles.css" media="screen">
```

```
<link rel="stylesheet" href="css/stylesheet.css" media="print">
```

# 1. Media Querys

El atributo **media** se utiliza para determinar qué hoja de estilo debe aplicarse. En este ejemplo, la hoja de estilo `styles.css` que se encuentra en la carpeta `css` se aplicará si la página web se muestra en la pantalla.

La hoja de estilo `stylesprint.css` se aplicará si la página web se imprime. Esta es una técnica común que permite a los desarrolladores dar estilo con colores ricos e imágenes coloridas en la pantalla, pero dar estilo con texto negro en un fondo blanco cuando la misma página web se imprime.

## 2. Breakpoints



## 2. Breakpoints

Las media queries pueden hacer más que detectar el tipo de medio actual. También pueden determinar el tamaño de la ventana gráfica, la parte visible de la página web. Para entender el código y la sintaxis de cómo una media query detecta el tamaño de la ventana gráfica, debes establecer un **breakpoint**, el punto en el que quieres que la página web cambie.

## 2. Breakpoints

El breakpoint es donde se aplican diferentes estilos a la página web para que cambie de forma que sea más fácil de leer y navegar para una determinada ventana gráfica, como la de un usuario que ve la página web en un teléfono inteligente.

**Table 6–1 Common Viewport Breakpoints**

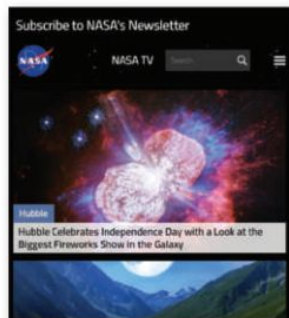
Device	Minimum Viewport Width	Maximum Viewport Width
Small smartphones	320px	480px
Larger smartphones and tablets	481px	768px
Tablets in landscape orientation, laptops, and small desktop monitors	769px	1279px
Large desktop monitors	1280px	NA

## 2. Breakpoints

(a) 320px



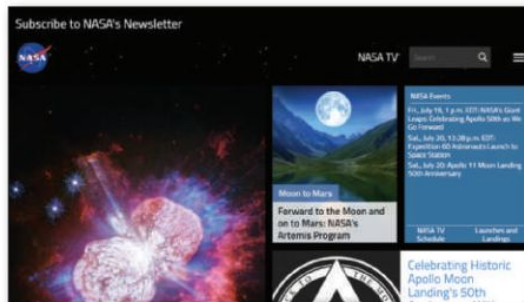
(b) 520px



(c) 768px



(d) 1000px



(e) 1440px



## 3. Media Query Expressions



### 3. Media Query Expressions

Una media query puede utilizar una expresión lógica para comprobar si una ventana gráfica ha alcanzado un punto de ruptura concreto. La expresión lógica incluye el nombre de una característica de la media query, una característica del entorno y un valor breakpoint que debe comprobarse.



### 3. Media Query Expressions

Si la expresión lógica se evalúa como "true", la media query aplica los estilos que siguen. A continuación se muestra un ejemplo de media query que incluye un tipo de medio y una expresión lógica:

```
<link rel="stylesheet" href="css/styles-mobile.css" media="screen and (max-width: 480px)">
```

### 3. Media Query Expressions

Este código indica a los navegadores que utilicen la hoja de estilo `styles-mobile.css` en la carpeta `css` cuando las pantallas tienen un ancho de ventana menor o igual a 480px (que es un tamaño máximo de ventana de visualización común para un teléfono y, por lo tanto, un punto de ruptura común para cambiar los estilos). En este caso, `max-width` es la característica y 480px es el valor del punto de ruptura que se comprueba en la condición lógica.

### 3. Media Query Expressions

Una media query también puede comprobar los puntos de rotura mínimos y máximos, como en el siguiente ejemplo:

```
<link rel="stylesheet" href="css/styles-tablet.css" media="screen and (min-width: 481px) and (max-width: 768px)">
```

### 3. Media Query Expressions

Llegados a este punto, es posible que te preguntes cómo seleccionar los valores a utilizar para los valores de los breakpoint, especialmente cuando se introducen constantemente tantos dispositivos móviles nuevos de todos los tamaños.

La solución es preparar distintas media queries para los dispositivos más usados en el momento. Cuando lleguen nuevos dispositivos al mercado tendrás que modificarlo.

### 3. Media Query Expressions

Una segunda forma de implementar las media queries es codificarlas directamente en un único archivo CSS utilizando @media

```
@media screen {  
  body {  
    background-color: #ccc;  
  }  
}
```

### 3. Media Query Expressions

En este caso, la pantalla @media identifica el medio al que se aplicarán los estilos. Aunque el ejemplo muestra sólo un estilo, es posible tener muchos estilos diferentes.

Otra aplicación de una media query en un archivo CSS para aplicar estilos a una página web impresa podría tener este aspecto:

### 3. Media Query Expressions

```
@media print {  
  body {  
    width: 100%;  
    color: #000;  
    background-color: #fff;  
  }  
}
```

### 3. Media Query Expressions

También es posible utilizar expresiones lógicas como vimos anteriormente con esta nueva sintaxis:

```
@media screen and (min-width: 481px) {  
    .mobile {  
        display: none;  
    }  
}
```



### 3. Media Query Expressions

Otros ejemplos con dos expresiones lógicas podrían ser:

```
@media all and (min-width: 481px) and (max-width: 768px) {
```

```
  body {
```

```
    color: #000;
```

```
  }
```

```
}
```

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

### 3. Media Query Expressions

Esta media query se aplica a todos los tipos de medios cuando ambas características son verdaderas. La anchura de la ventana gráfica debe ser de al menos 481px pero no puede superar los 768px. Los tres tipos de medios más comunes son screen, print y all.

Table 6–2 Common Media Query Features	
Feature	Description
max-width min-width	Width of the viewport in pixels
max-resolution min-resolution	Resolution of the output device in dots per inch or dots per centimeter
max-height min-height	Height of the viewport in pixels
orientation	Orientation of the device (landscape or portrait)

### 3. Media Query Expressions

Además de aparecer como link a hojas de estilos distintas:

```
<link rel="stylesheet" href="css/styles-mobile.css" media="screen and (max-width: 480px)">
```

```
<link rel="stylesheet" href="css/styles-tablet.css" media="screen and (max-width: 768px)">
```

```
<link rel="stylesheet" href="css/styles-desktop.css" media="screen and (max-width: 1279px)">
```

También se puede incluir dentro de una única hoja de estilos (main.css):

```
@media (max-width: 480px) {...}
```

```
@media (max-width: 768px) {...}
```

```
@media (max-width: 1279px) {...}
```

## **4. Uso de pseudoclases dinámicas**



## 4. Uso de pseudoclases dinámicas

¿Ha visitado alguna vez un sitio web en el que los enlaces cambian de color cuando los señala (pasa por encima) o hace clic en un enlace?

El color del enlace cambia debido al uso de pseudoclases dinámicas en la hoja de estilo. Las pseudoclases dinámicas permiten cambiar el estilo de un enlace en función de cuatro estados del mismo: link, visitado, hover y activo.

## 4. Uso de pseudoclases dinámicas

Es posible definir un estilo único para los enlaces normales, visitados, hover y activos creando cuatro reglas de estilo distintas con

- `a:link`
- `a:visited`
- `a:hover`
- `a:active`

**Table 6–3 Dynamic Pseudo-Classes**

Dynamic Pseudo-Class	Used to Style
<code>:link</code>	Unvisited link
<code>:visited</code>	Link that has been clicked
<code>:hover</code>	Link when the mouse is hovering over it
<code>:active</code>	Link at the moment it is clicked

## 4. Uso de pseudoclases dinámicas

Estas pseudoclases dinámicas **deben utilizarse en el siguiente orden**: link, visited, hover, active. No es necesario utilizar todas las pseudoclases; sin embargo, si decide omitir una pseudoclase de su diseño, asegúrese de mantener el mismo orden de las pseudoclases en el código CSS.

Estas pseudoclases dinámicas **se utilizan más a menudo en una ventana gráfica de escritorio** porque los dispositivos móviles y las tabletas son dispositivos táctiles y no tienen la opción de pasar por encima ni de hacer clic.

## 4. Uso de pseudoclases dinámicas

Otro tipo de pseudoclase dinámica se llama **:focus**. Normalmente se utiliza para dar estilo a un elemento de entrada de formulario cuando el elemento recibe el foco.



## 5. Uso de gradientes



## 5. Gradientes

Como has aprendido, CSS es una poderosa herramienta para dar formato a las páginas web. Ya has trabajado con varias propiedades CSS, pero CSS tiene muchas más propiedades que explorar.

Una de esas propiedades es el gradiente. Un **gradiente** es una **transición gradual de un color a otro**. Puedes crear dos tipos de degradados usando CSS: **lineal y radial**

## 5. Gradientes

Los **degradados lineales** pueden hacer la transición desde varios **ángulos** diferentes. La transición por defecto es de arriba a abajo. Los degradados lineales también pueden hacer la transición hacia arriba, hacia la izquierda, hacia la derecha o en diagonal.

## 5. Gradientes

Para crear un degradado lineal, utilice la función de degradado lineal, `linear-gradient`. El siguiente es un ejemplo de cómo utilizar la función de gradiente lineal:

```
body {  
    background: linear-gradient(#fff, #0000ff);  
}
```

## 5. Gradientes

Table 6–4 Linear Gradients

Direction	Examples
top to bottom (default)	<pre>body {     background: linear-gradient: (#fff, #0000ff); }</pre>
left to right	<pre>body {     background: linear-gradient: (to right, #fff, #0000ff); }</pre>
diagonal	<pre>body {     background: linear-gradient: (to bottom right, #fff, #0000ff); }</pre>
specified angle	<pre>body {     background: linear-gradient: (180deg, #fff, #0000ff); }</pre>

## 5. Gradientes

Los **degradados radiales** se especifican por su centro. El color comienza en el **centro** y transita en dirección radial hacia otro u otros colores. Para crear un degradado radial, debe especificar al menos dos colores. El siguiente es un ejemplo de degradado radial:

```
body {  
    background: radial-gradient(#ff0000, #fff, #0000ff);  
}
```

# Ejercicio

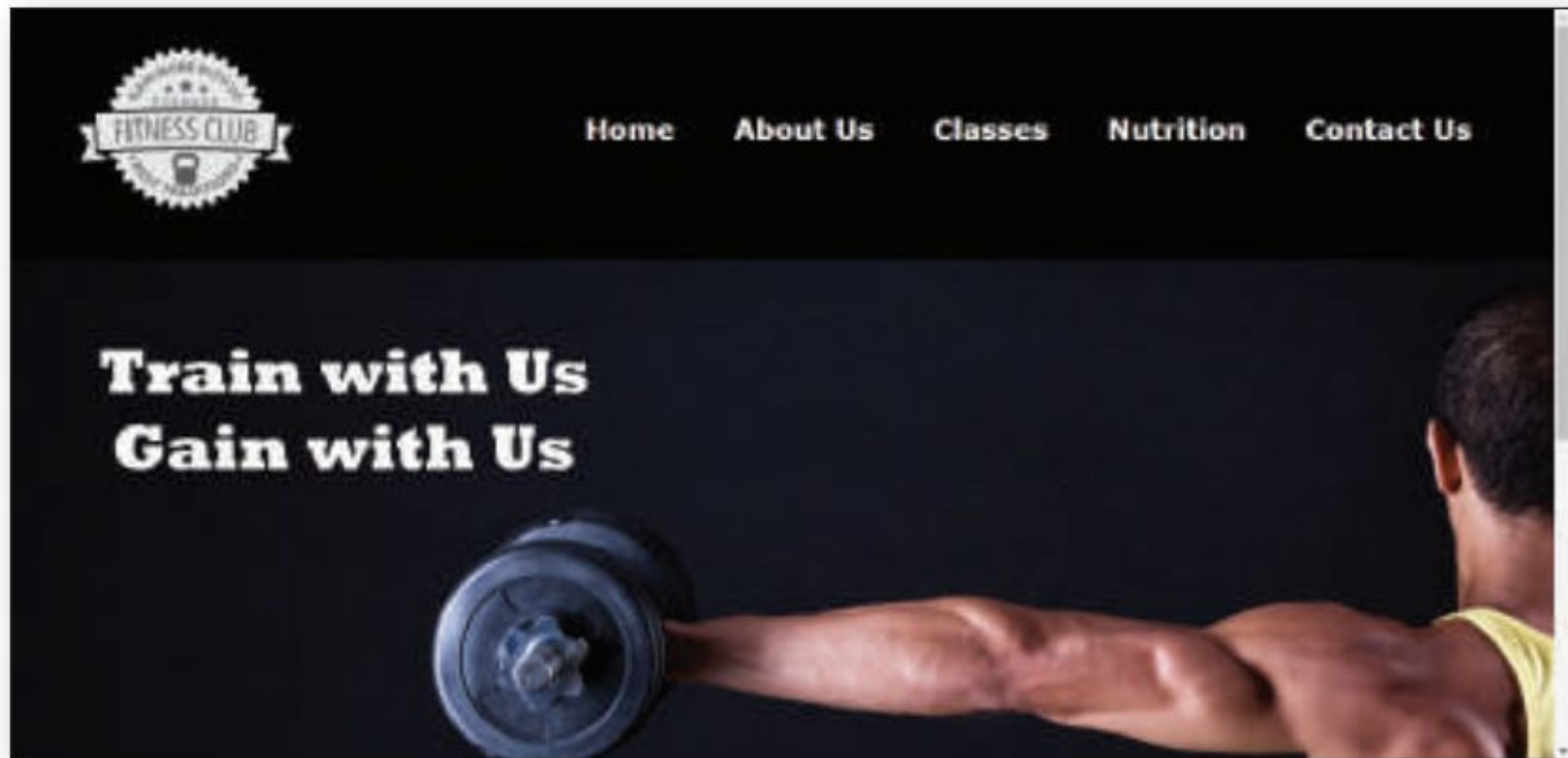


# Ejercicio

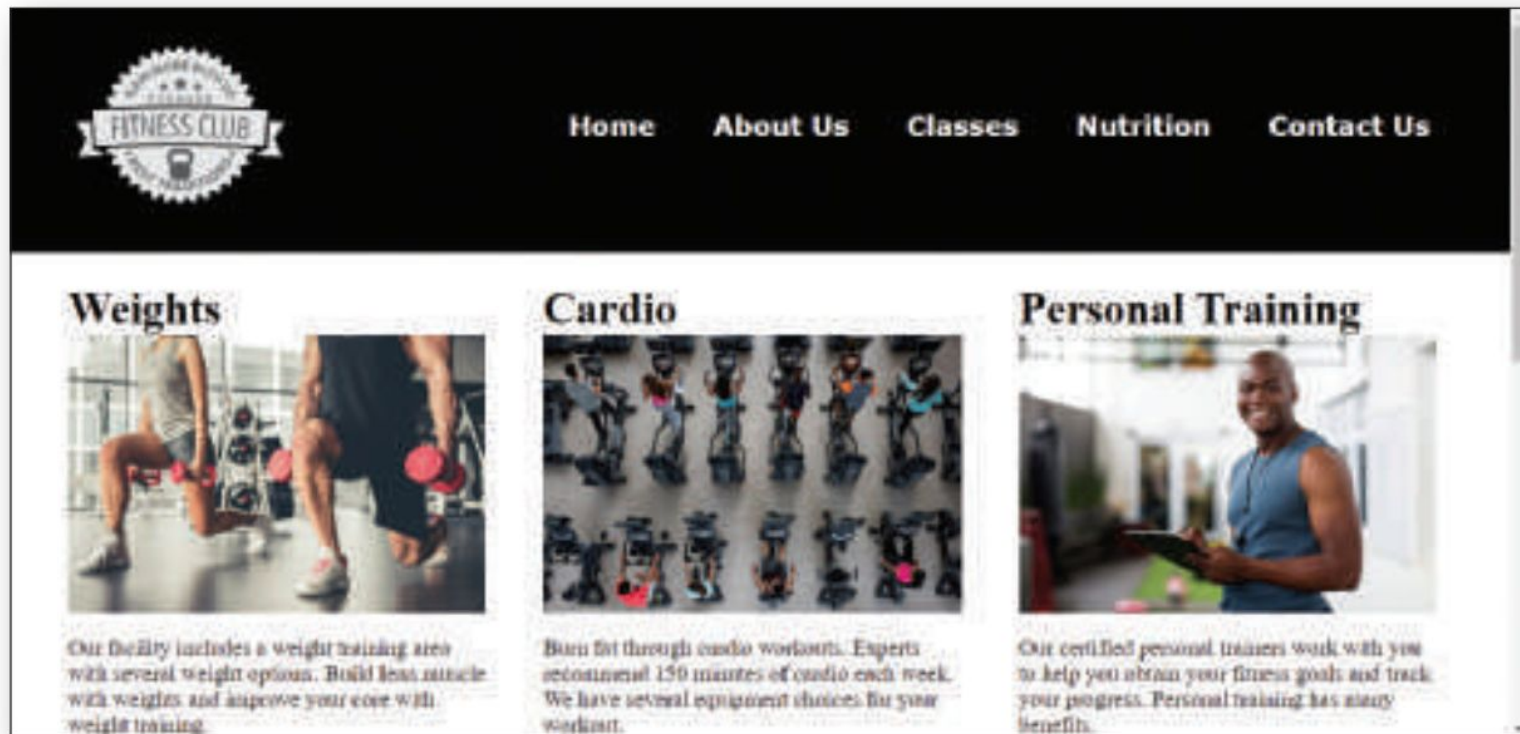
1. Tenéis que hacer que la página fitness sea responsiva.
2. Tenéis que añadir pseudoclases dinámicas para desktop.
3. Tenéis que añadir un gradiente.



# Home



# About Us Page



## Weights



Our facility includes a weight training area with several weight options. Build lean muscle with weights and improve your core with weight training.

## Cardio



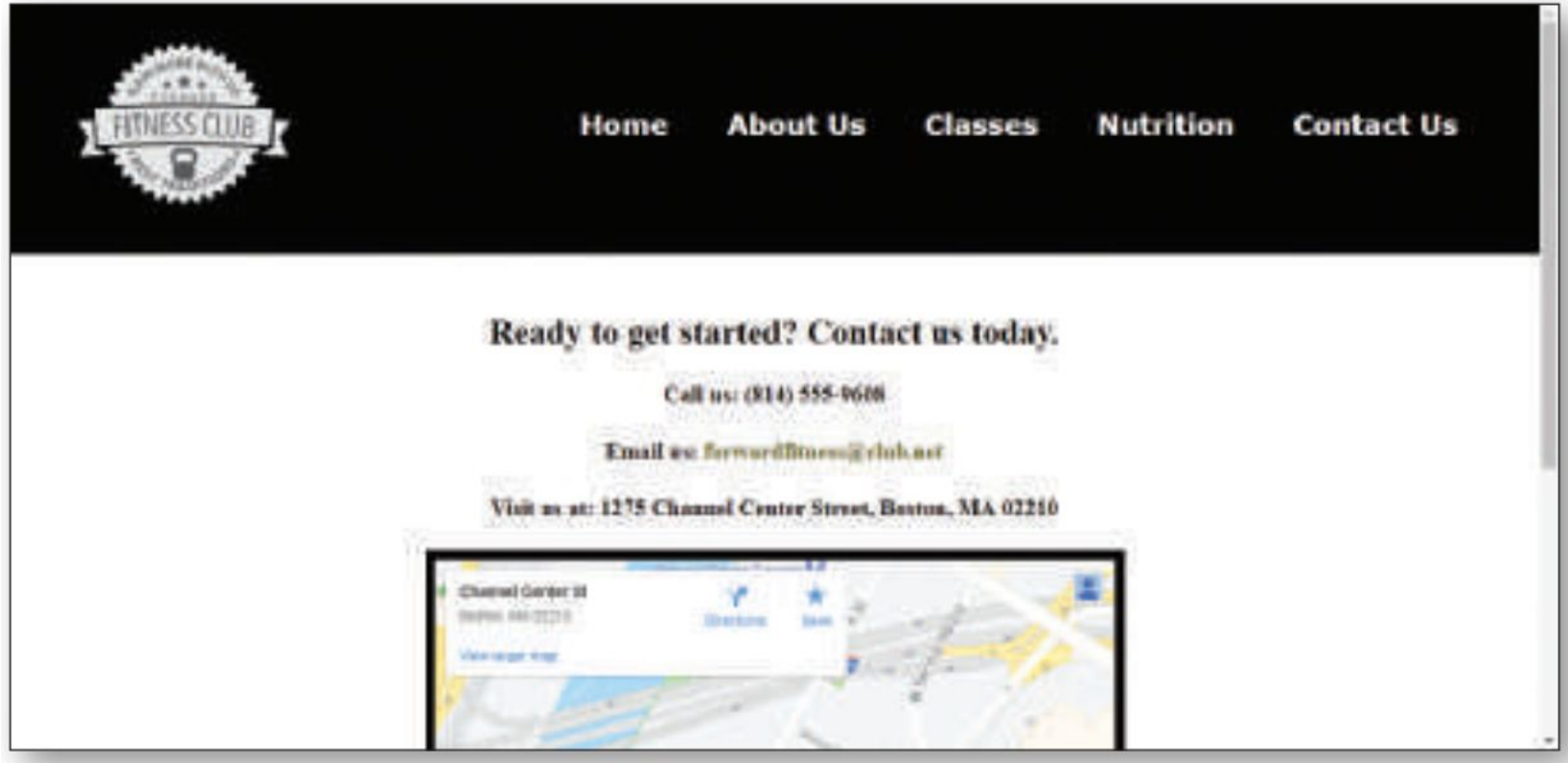
Burn fat through cardio workouts. Experts recommend 150 minutes of cardio each week. We have several equipment choices for your workout.

## Personal Training



Our certified personal trainers work with you to help you obtain your fitness goals and track your progress. Personal training has many benefits.

# Contact Us Page

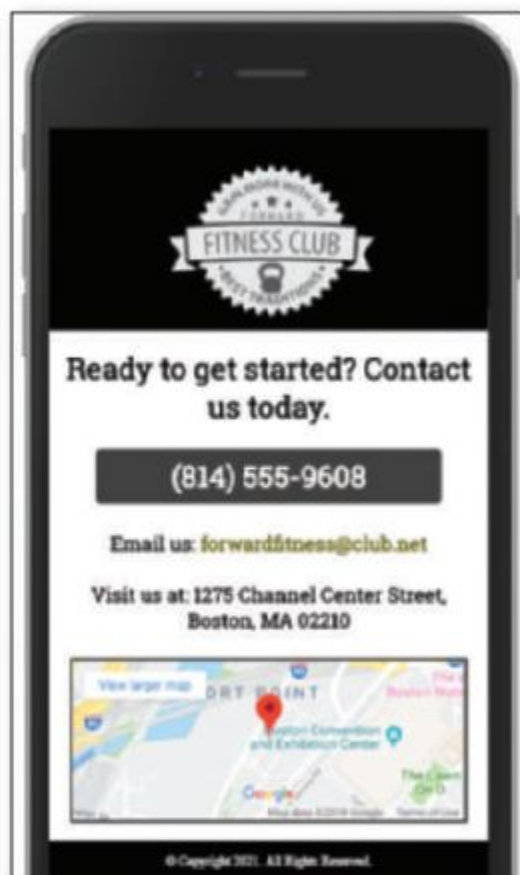




(a) Home Page



(b) About Us Page



(c) Contact Us Page