04. RESPONSIVE WEB DESIGN

índice

01:

Introducción al diseño web adaptativo

01. Tipos de diseño web

- Al crear una interfaz web existen diferentes tipos de diseño.
- Estos métodos se pueden utilizar de forma combinada.

Diseño fijo

- Dispone de medidas fijas.
  - No modificadas para los distintos dispositivos.
- Diseño que no cumple las normas de un diseño web responsivo o responsive web design.
- Ejemplo

Diseño elástico

- Se definen las dimensiones con unidades «em».
  - El «em» mide el ancho de la letra «M» mayúscula de una tipografía dada y a un tamaño dado.
    - No mide siempre lo mismo (16 px en navegadores actuales).
  - El diseño se adapta cuando el visitante del sitio cambie el tamaño del texto.
  - No se adaptará según los cambios de tamaño de la ventana del navegador.
- Ejemplo

Diseño líquido o fluido

- El diseño se adapta a la ventana del dispositivo, definiendo los tamaños mediante porcentajes.
  - Perjudica la experiencia de usuario.
  - No permite controlar el diseño.
  - El diseño varía constantemente según el tamaño del dispositivo.
  - No se puede diseñar al píxel para todos los tamaños.
- Ejemplo

Diseño web responsivo o adaptable

- Se basa en el uso de media queries.
  - Define estilos condicionales con puntos de ruptura o puntos de interrupción, aplicables en determinadas situaciones.
- Ejemplo
  - Aplica un estilo diferente en pantallas de ancho superior a 1200px, inferior a 1200px, e inferior a 600px.

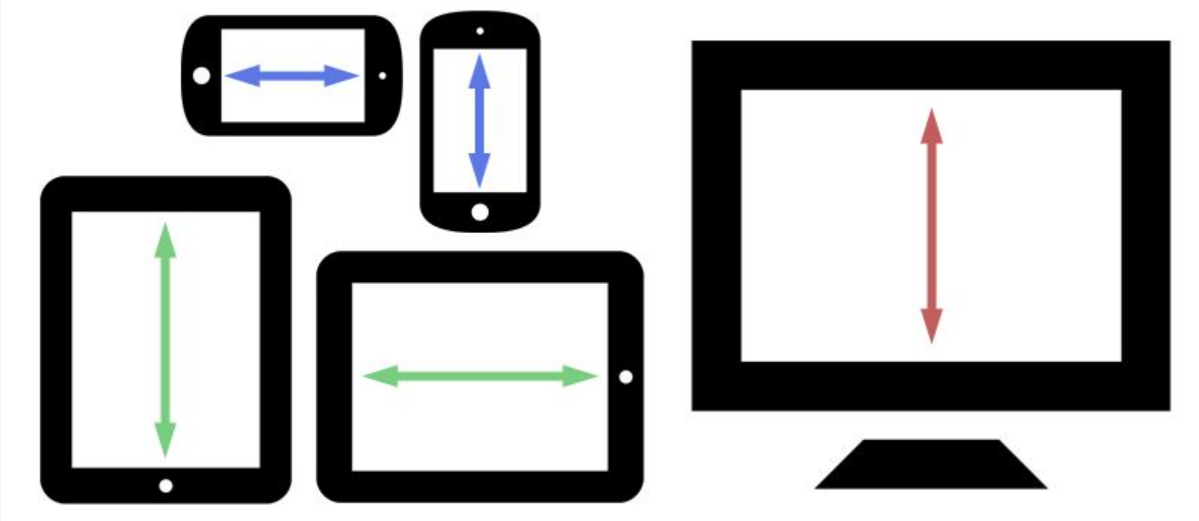
Diseño flexible

- Utiliza las propiedades CSS "min-width" y "max-width" para que las anchuras de los bloques puedan adaptarse dentro de unos mínimos y máximos.

- Lo ideal es utilizar un diseño web responsivo, preferentemente no elástico y con mezcla de diseño flexible y líquido según el contenedor.
- Se ofrece la **mejor experiencia de usuario** y podremos **considerar las distintas medidas de los dispositivos** utilizando tan solo HTML y CSS.

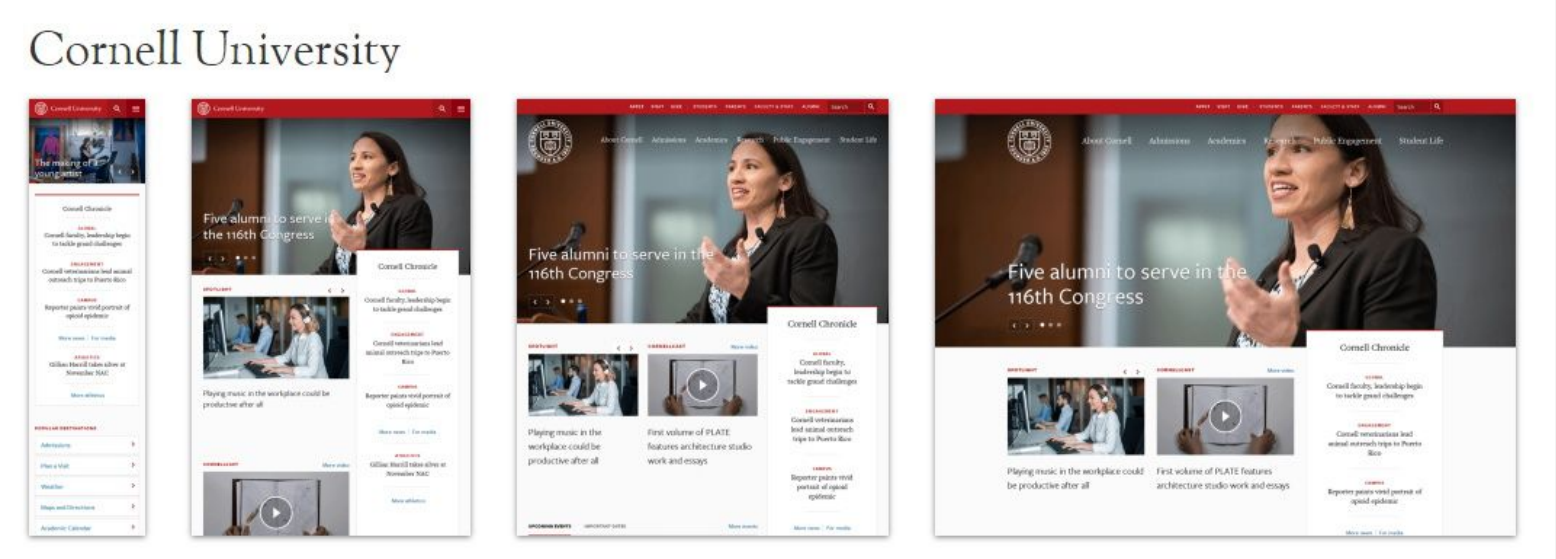
02. Responsive web design

- Cada vez es más frecuente acceder a Internet con diferentes tipos de dispositivos.
  - Diferentes pantallas y resoluciones, distintos tamaños y formas.



02. Responsive web design

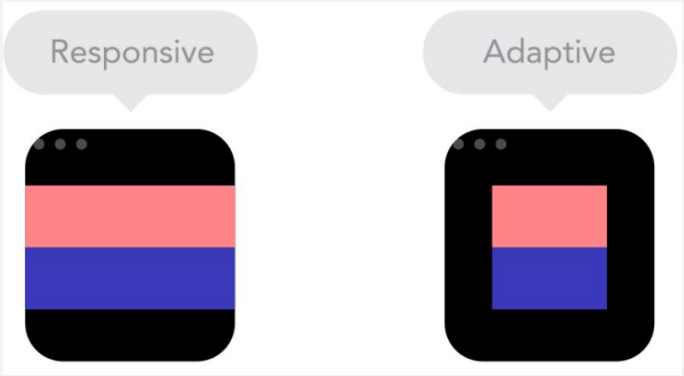
- Se pretende diseñar una sola web, que se adapte visualmente al dispositivo utilizado **Responsive Web Design** (o RWD).
  - Ejemplos



02. Responsive web design

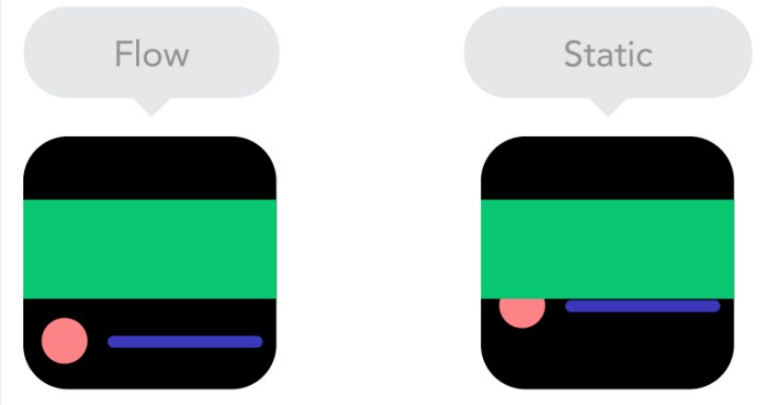
Conceptos generales

- Diferencia entre diseño responsivo y diseño adaptativo.



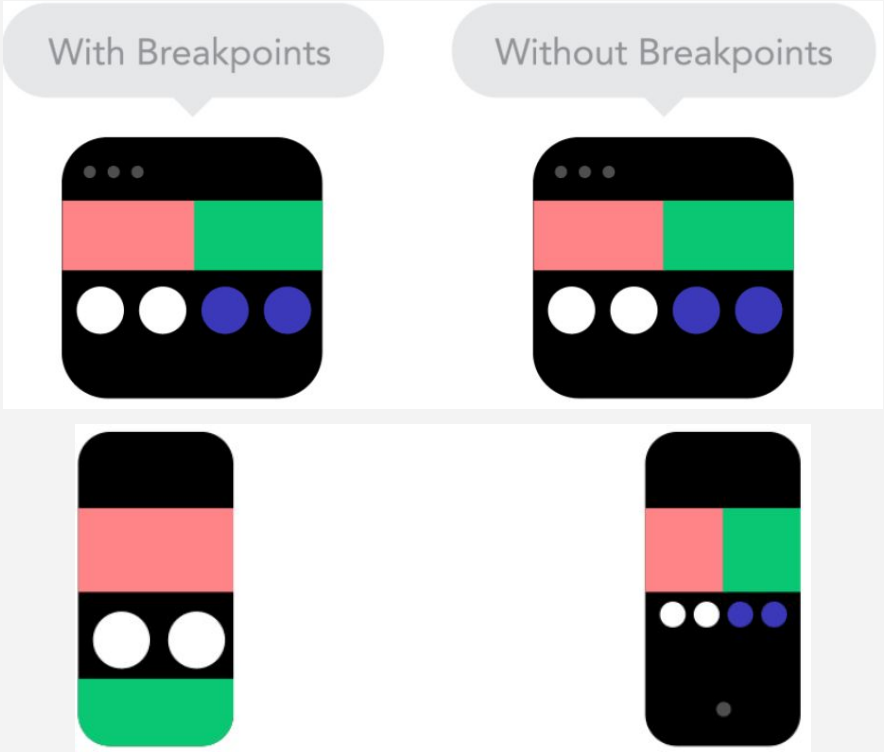
02. Responsive web design

- Trabajar con unidades relativas.
  - Evitar unidades fijas o estáticas.
- Utilizar propiedades como min-width o max-width.
- Mantener el flujo de los elementos cuando cambian de tamaño.
  - Evitar que se solapen unos con otros.



02. Responsive web design

- Utilizar «puntos de control».



02. Responsive web design

- **Importante**: conocer los formatos de pantalla más comunes.
  - myDevice.io
- Conocer el público del sitio web.
  - Desde dónde acceden más los usuarios (móvil o escritorio).

Estrategias de diseño

- Mobile first: primero nos centramos en el diseño para dispositivos **móviles** y luego en el resto.
- Desktop first: nos centramos en el diseño para dispositivos **de escritorio** y luego en el resto.

03. Bases del responsive web design

Diseño con porcentajes

- Los porcentajes son relativos al contenedor padre
  - Si especificamos un porcentaje a un elemento, el navegador va a tomar dicho porcentaje del contenedor.
  - Si queremos basarnos en el tamaño del navegador, hay que usar unidades de viewport.
- Ejemplo
  - Utilizar porcentajes no garantiza un diseño adaptativo de calidad.
  - **Problema**: la suma del tamaño de los elementos (70% + 30%) junto a los bordes (2px por cada lado), margin o padding, es superior al 100% del contenedor padre (descuadra el diseño).
  - Solución:
    - Eliminar bordes y reducir porcentajes.
    - Usar box-sizing: border-box.
    - Utilizar Flexbox o Grid.

03. Bases del responsive web design

Tamaños máximos y mínimos

- Uso de las propiedades max-width y min-width.
  - Incompatibles si se define la propiedad width.

```
.picture {
  min-height: 200px;     /* Por defecto, height es 0 */
  background: grey;      /* Simplemente, para verlo visualmente */

  max-width: 1024px;
  min-width: 800px;
}
```

03. Bases del responsive web design

Viewport

- Hace referencia a la región visible del navegador.
  - Parte de la página que se visualiza en el navegador.
  - Los usuarios pueden redimensionar la ventana para reducir el tamaño del viewport y simular que se trata de una pantalla y dispositivo más pequeño.

```
<meta name="viewport" content="initial-scale=1, width=device-width">
```

Propiedades	Valor	Significado
width	device-width	Indica un ancho para el viewport.
height	device-height	Indica un alto para el viewport.
initial-scale	1	Escala inicial con la que se visualiza la página web.
minimum-scale	0.1	Escala mínima a la que se puede reducir al hacer zoom.
maximum-scale	10	Escala máxima a la que se puede aumentar al hacer zoom.
user-scalable	no/fixed	**yes/zoom**

Más info

02:

Posicionamiento de elementos en CSS

01. Media queries en CSS3

- Hay situaciones en las que determinados componentes visuales deben aparecer en un tipo de dispositivos, o deben existir ciertas diferencias.
- Con las **media queries** podemos definir estilos condicionales, aplicables únicamente en determinadas situaciones.
    - El uso más extendido es para establecer estilos diferentes para cada ancho de pantalla.

```
<meta name="viewport" content="initial-scale=1, width=device-width">
```

01. Media queries en CSS3

- Reglas de CSS que permiten crear un bloque de código que sólo se procesará en los dispositivos que cumplan los criterios especificados como condición.

```
@media screen and (*condición*) {
  /* reglas CSS */
  /* reglas CSS */
}

@media screen and not (*condición*) {
  /* reglas CSS */
  /* reglas CSS */
}
```

Tipos de medios

Tipo de medio	Significado
screen	Monitores o pantallas de ordenador. Es el más común.
print	Documentos de medios impresos o pantallas de previsualización de impresión.
speech	Lectores de texto para invidentes (Antes **aural**, el cuál ya está obsoleto).
all	Todos los dispositivos o medios. El que se utiliza **por defecto**.

01. Media queries en CSS3

- Ej: definir diferentes estilos dependiendo del dispositivo que estamos utilizando.

```
@media screen and (max-width: 640px) {
  .menu {
    background: blue;
  }
}

@media screen and (min-width: 640px) and (max-width: 1280px) {
  .menu {
    background: red;
  }
}

@media screen and (min-width: 1280px) {
  .menu {
    background: green;
  }
}
```

- Ej: aplicar un estilo diferente en pantallas de ancho superior a 1024px, inferior a 1024px e inferior a 480px.

```
.click:after {
    content:"En pantalla grande";
}
@media screen and (max-width: 1024px) {
    .click:after {
        content:"Tablet";
    }
}
@media screen and (max-width: 480px) {
    .click:after {
        content:"Movil";
    }
}
```

01. Media queries en CSS3

Propiedades o condiciones

Nombre	Descripción
`width`	Anchura del viewport
`height`	Altura del viewport
`aspect-ratio`	Relación de aspecto anchura-altura del viewport
`orientation`	Orientación del viewport
`resolution`	Densidad de pixeles del dispositivo
`scan`	Proceso de escaneo del dispositivo
`grid`	Si el dispositivo es grid o bitmap
`update-frequency`	Velocidad de actualización del dispositivo para modificar la apariencia del contenido
`overflow-block`	Cómo maneja el dispositivo el contenido que excede los límites del viewport a lo largo del eje de bloque
`overflow-inline`	Cómo maneja el dispositivo el contenido que excede los límites del eje *inline*
`color`	Componente de número de bits por color del dispositivo, o cero si el dispositivo no es a color.
`color-index`	Número de entradas en la tabla de búsqueda de color del dispositivo, o cero si el dispositivo no usa una tabla.
`monochrome`	Bits por pixel en el buffer de marco monocromático del dispositivo, o 0 si el dispositivo no es monocromático.
`hover`	Si se puede posicionar el puntero sobre los elementos

01. Media queries en CSS3

Condicionales CSS

- No forman parte de las media queries en sí.
- Regla @supports para establecer condicionales y crear reglas similares a @media pero dependiendo de si el navegador del usuario soporta una característica concreta.

```
@supports not (display: grid) and (display: flex) {
  .content {
    display: flex;
    justify-content: center;
  }
}

@supports not (display: flex) {
  .content {
    display: block;
  }
}
```

01. Media queries en CSS3

Operadores lógicos en media queries

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

```
@media not screen and (monochrome) { ... }
```

02. Revisión de posicionamiento de elementos en CSS

- En CSS básico si tenemos varios **elementos en línea** (uno detrás de otro) aparecerán colocados de **izquierda hacia derecha.**
- Si son **elementos en bloque** se verán colocados **desde arriba hacia abajo**.
- Los elementos se combinarán y anidarán, formando esquemas más complejos.

02. Revisión de posicionamiento de elementos en CSS

Posicionamiento CSS

- Inicialmente se ha trabajo con posicionamiento **estático**.
  - Elementos posicionados según el HTML.
- Modos alternativos: propiedad **position**.

Valor	Significado
static	Posicionamiento estático. Utiliza el orden natural de los elementos HTML.
relative	Posicionamiento relativo. Los elementos se mueven ligeramente en base a su posición estática.
absolute	Posicionamiento absoluto. Los elementos se colocan en base al contenedor padre.
fixed	Posicionamiento fijo. Idem al absoluto, pero aunque hagamos scroll no se mueve.

02. Revisión de posicionamiento de elementos en CSS

Posicionamiento CSS

- En posicionamientos distintos a static se emplea:

Propiedad	Valor	Significado
top:	`auto` - `size`	Empuja el elemento una distancia desde la parte superior hacia el inferior.
bottom:	`auto` - `size`	Empuja el elemento una distancia desde la parte inferior hacia la superior.
left:	`auto` - `size`	Empuja el elemento una distancia desde la parte izquierda hacia la derecha.
right:	`auto` - `size`	Empuja el elemento una distancia desde la parte derecha hacia la izquierda.
z-index:	`auto` - `número`	Coloca un elemento en el eje de profundidad, más cerca o más lejos del usuario.

  - Ej.: left: 20px, mueve **desde la izquierda** 20px **a la derecha**.

02. Revisión de posicionamiento de elementos en CSS

Posicionamiento relativo

- Los elementos se colocan igual que en el posicionamiento estático, pero dependiendo del valor de las propiedades top, bottom, left o right varía la posición del elemento.

Posicionamiento absoluto

- Coloca los elementos **utilizando como punto de origen el primer contenedor con posicionamiento diferente a estático**.
- Utiliza el documento completo como referencia.

02. Revisión de posicionamiento de elementos en CSS

Posicionamiento fijo

- Igual que el posicionamiento absoluto, salvo que el elemento se muestra en una posición fija **según la región visual del navegador**.
  - Ej.: aunque el usuario haga scroll en la página web, el elemento seguirá en el mismo sitio posicionado.

Profundidad (niveles)

- La propiedad **z-index** establece el nivel de profundidad en el que está un elemento sobre los demás.
- Un elemento se colocará encima o debajo de otro.

02. Revisión de posicionamiento de elementos en CSS

Float. Elementos flotantes.

- La propiedad **float** cambia el flujo de ordenación de elementos.
  - Ej.: un elemento «flotará» a la izquierda o a la derecha de otro elemento.

Propiedad	Valor	Significado
`float`	none	left
`clear`	none	left

```
ul {
  background: grey;
}

li {
  background: blue;
  width: 100px;
  padding: 8px;
  margin: 8px;
  color: white;
}

ul, li {
  float: left;
}
```

- Ej.: los ítems de la lista flotan uno a continuación de otro.
- La propiedad **clear** impide elementos flotantes en la zona indicada, a la izquierda del elemento (left), a la derecha (right) o en ambos lados (both).

04. RESPONSIVE WEB DESIGN