

Taller de Programación Web

MSc. Yuliana Jáuregui Rosas



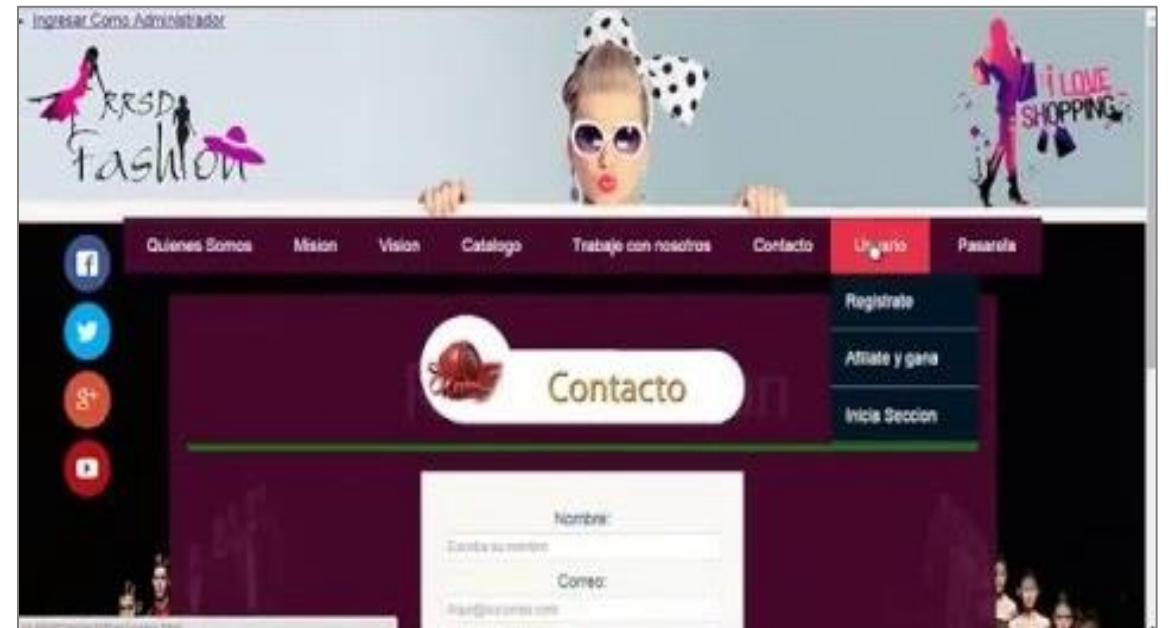
Universidad
Tecnológica
del Perú

Saberes previos

- ¿Qué etiquetas se usan para construir una tabla?
- ¿Que etiquetas se usan para construir una lista ordenada y desordenada?
- ¿Qué etiqueta se usa para construir una casilla de verificación, un botón de opción y un botón de envío?

CSS - Cascading Style Sheets (Hojas de estilo en cascada)





Desaprende lo que te limita

Utilidad

- ¿Creen que es importante el diseño (estilo gráfico) de una página web?
- ¿Influirá el diseño en el numero de visitantes de nuestra página?

Logro



Al finalizar esta sesión el estudiante conoce los fundamentos básicos de las hojas de estilo implementándolo en el diseño de una página web

Temario



- CSS
- Niveles CSS
- HTML y CSS
- Estructura semántica HTML
- Estructura CSS
- Como incluir CSS
- Tipos de selectores
- Unidades de medidas
- Propiedades comunes
- Propiedades básicas

CSS

- Las hojas de estilo CSS son un conjunto reglas que se definen en un archivo .css y que describen el **aspecto** que deben tener los diferentes **elementos HTML** de una página.
- Trabaja bajo la **filosofía de patrones o plantillas**, es decir, no es necesario especificar cada uno de los elementos, sino que se pueden definir reglas como:
 - *“Los títulos de nivel 1 y 2 deben ser de color rojo y tamaño de fuente de 16 y 14 pixeles respetivamente.”*
 - *“El texto de los párrafos deben estar alineados a la izquierda, con tamaño de fuente de 12 pixeles y color de fuente azul marino.”*

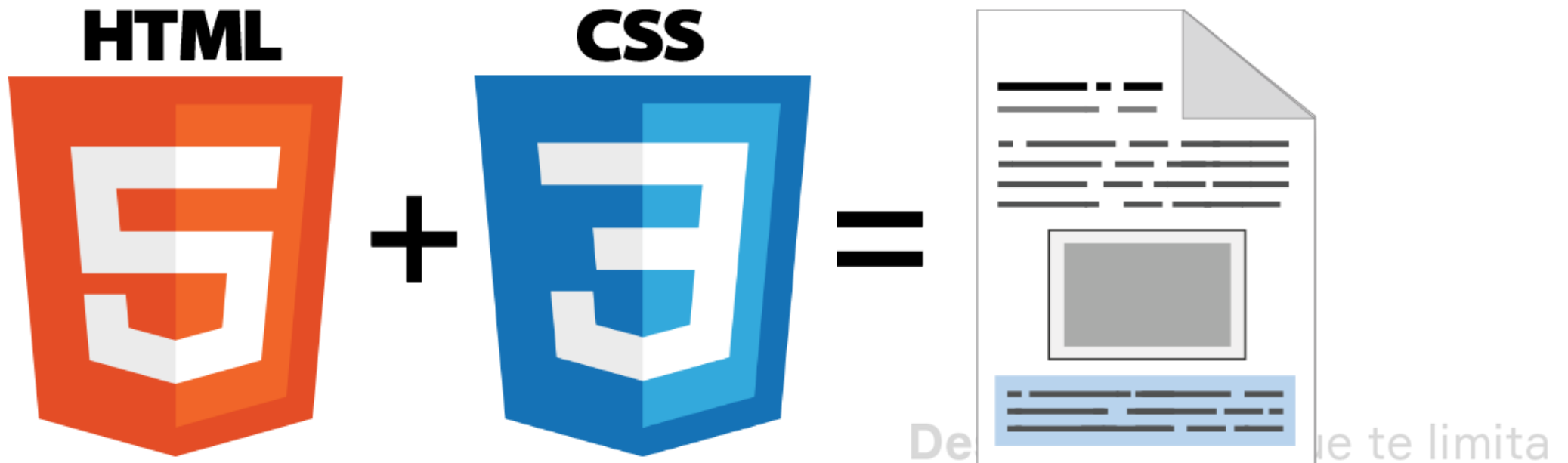
Niveles CSS

- A lo largo de su historia, **CSS** ha evolucionado en diferentes versiones, denominados niveles:

| Nivel | Año | Descripción |
|---------------|------|---|
| CSS1 | 1996 | Propiedades de fuente, colores, alineación, etc... |
| CSS2 | 1998 | Propiedades de posicionamiento, tipos de medios, etc... |
| CSS2.1 | 2005 | Corrige errores de CSS2 y modifica ciertas propiedades |
| CSS3 | 2011 | Inicio de módulos separados con funcionalidades nuevas |

HTML y CSS

En las páginas web el lenguaje HTML se usa para **estructurar** el contenido semánticamente (títulos, subtítulos, texto, etc.) y CSS para la **maquetación y estética** del mismo.

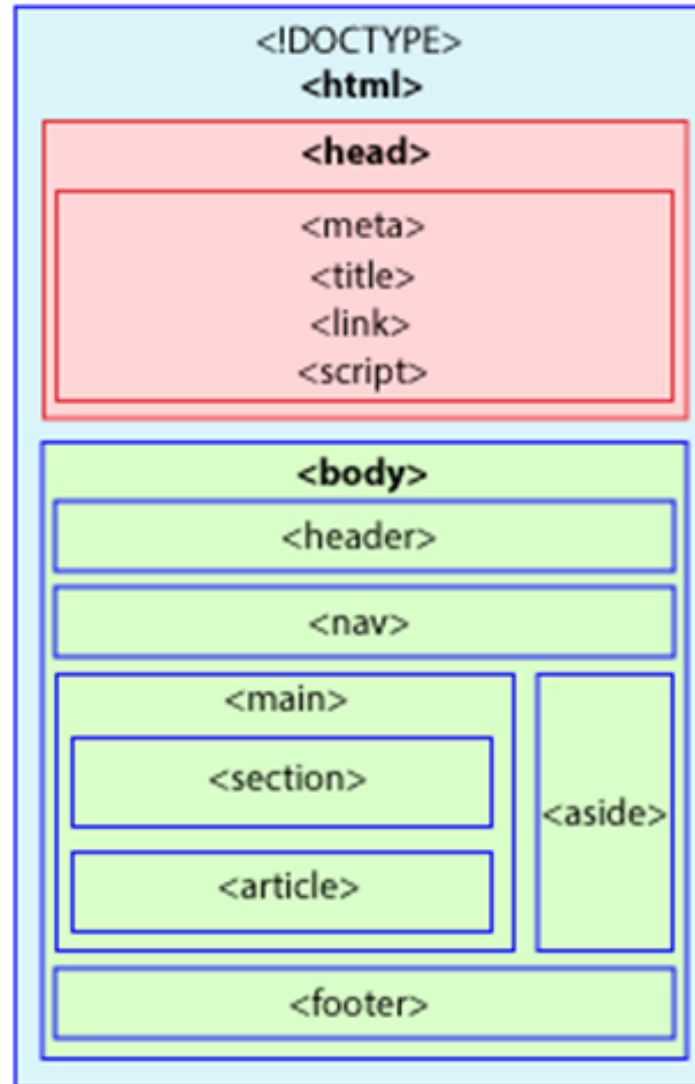


Estructura Semántica HTML



- Es necesario estructurar el contenido de una pagina HTML para poder manejar mejor su maquetación (aspecto).
- HTML5 incluye elementos semánticos, o marcados semánticos, que ayudan a definir las distintas divisiones de una página web.
- Las versiones anteriores solo utilizaba la etiqueta <div>.

Estructura Semántica HTML



Desaprende lo que te limita

Estructura Semántica HTML

- Elementos de la cabecera:

| Elemento | Descripción |
|----------|--|
| <meta> | Define los metadatos que no pueden ser definidos usando otro elemento HTML. |
| <title> | Título del documento. Se muestra en la barra superior del navegador o en las pestañas de página. |
| <link> | Usada para enlazar JavaScript y CSS externos con el documento HTML actual. |
| <style> | Etiqueta de estilo usada para escribir CSS interno. |

Estructura HTML Semántico

- Elementos del cuerpo:

| Elemento | Descripción |
|-----------|---|
| <header> | Define la cabecera de una web o de un elemento. |
| <nav> | Define una sección que contiene un menú de navegación. |
| <main> | Define el contenido principal del documento. Solamente existe un elemento <main> en el documento. |
| <section> | Define secciones de una web. |
| <article> | Define unidades de contenido que podrían existir independientemente del resto del contenido. |
| <aside> | Define a barra lateral de una página web. Puede contener todo tipo de contenido. Sin embargo, se suele utilizar para mostrar enlaces. |
| <footer> | Define el pie de página. También se puede utilizar para definir el pie otros elementos. |
| <address> | Define una sección que contiene información de contacto. |

Estructura CSS

La **estructura CSS** se basa en reglas que tienen el siguiente formato:

```
selector { propiedad : valor }
```

Donde:

- **Selector:** Es el elemento HTML que vamos a seleccionar del documento para aplicarle un estilo concreto.
- **Propiedad:** Característica de un elemento.
- **Valor:** Valores concretos de una característica.

Estructura CSS

- Se puede aplicar mas de una propiedad a un selector, en este caso se separa las propiedad con un punto y coma (;).

```
selector {  
    propiedad : valor ;  
    propiedad : valor  
}
```

Como incluir CSS

- Se puede incluir CSS a una pagina web de tres formas diferentes:
 - **CSS embebido:** a través del atributo **style**.
 - **CSS interno:** dentro del lenguaje HTML.
 - **CSS externo:** para este caso se crea un archivo *.css y luego se enlaza con la pagina web.

CSS embebido

- Aplica estilos en un documento HTML de forma directa en las propias etiquetas, a través del atributo **style**:

```
<p>Uno elige <span style="color:blue"> lo que es</span>!</p>
```

- Esta manera de aplicar estilos, **NO se recomienda** salvo en casos muy específicos, pues se pierde la independencia de la presentación y contenido

CSS interno

- Permite incluir estilos en un documento HTML a través de la etiqueta `<style></style>` que se añade en la cabecera del documento.
- Esta forma de aplicar estilos tampoco es recomendable.
- Ejemplo:

```
<html>
  <head>
    <title>Título de la página</title>
    <style type="text/css">
      p {
        color: rgb(32,32,32);
        text-align: left;
      }
    </style>
  </head>
  <body>
    ...
  </body>
</html>
```

CSS externo

- En la cabecera de la pagina web `<head> </head>` se incluye la etiqueta `<link>`:

```
<link rel="stylesheet" type="text/css" href="estilo.css" />
```

- Donde:
 - **rel="stylesheet"** indica que el enlace es con una hoja de estilos
 - **type="text/css"** indica que el archivo es de texto, en sintaxis CSS
 - **href="estilo.css"** indica el nombre del archivo *.css

Tipos de selectores

Selector de
elemento

Selector de
identificación

Selector de
clase

Selector de
atributo

Selector de
pseudoclase

Selector de elemento

- Selecciona el elemento de una página utilizando el nombre de la etiqueta.
- **Ejemplo:**

```
p{  
    color: blue;  
    font-family: Tahoma;  
    font-size: 14pt  
}
```


Selector de identificación (id)

- Selecciona el elemento de una página usando el atributo **id**, para usar este selector en un archivo CSS se antepone la almohadilla '#' al valor del atributo **id**.
- El atributo 'id' sólo admite un único valor, además este valor debe ser único entre todos los elementos de una pagina.
- Los selectores id permite **aplicar un estilo exclusivo a un elemento en concreto**, independientemente del que se le haya aplicado de forma general.



Selector de identificación (id)

- Ejemplo:

```
<p>Aprendiendo CSS</p>
<p id="resaltar1">Los identificadores son unicos</p>
<p>Su selector comienza con #</p>
<p id="resaltar2">Identifica un elemento entre varios</p>
<p>Aplica un estilo propio</p>
```

Aprendiendo CSS

Los identificadores son unicos

Su selector comienza con #

Identifica un elemento entre varios

Aplica un estilo propio

```
p{
    color: blue;
    font-family: Tahoma;
    font-size: 14pt
}

#resaltar1{
    color: red;
    font-weight: bold
}

#resaltar2{
    color: orange;
    font-weight: bold
}
```

Desaprende lo que te limita

Selector de Clase (class)

- Selecciona el elemento de una página usando el atributo **class**, para usar este selector en un archivo CSS se antepone un punto '.' al valor del atributo **class**.
- Una clase puede estar presente en varios elementos de una página, es así que el selector `.miclase` apuntará a todos los elementos de la página que tengan declarado `class="miclase"`.
- El atributo 'class' admite valores múltiples separados por un espacio en blanco.



Selector de Clase (class)

- Ejemplo:

```
<p class="miclase1">Feliz semana</p>  
<p class="miclase1 miclase2">Aprendiendo uso de clases en CSS</p>
```

```
.miclase1{ /* Aplica por igual a los 2 'p' anteriores */  
  color: green  
}  
.miclase1.miclase2{ /* Sólo aplica al segundo 'p' */  
  font-family: Courier New;  
  font-size: 30px  
}
```

Feliz semana

Aprendiendo uso de clases en CSS

Nota: en caso de colisión de estilos prevalece la segunda regla con independencia del orden por ser más específica.

Desaprende lo que te limita

Selector de atributo

- Selecciona los elementos en una página con el atributo especificado.

```
<a href="ejemplo.org" title="hola">Mi página</a>
```

```
a[title]{  
  color: green;  
  font-style: italic  
}
```

Mi página

Selector de Pseudoclase

- El selector de pseudoclase es una herramienta para poder apuntar desde la hoja de estilos a un elemento del DOM (Document Object Model) en base a alguna información o detalle ajeno al propio elemento, por ejemplo cuando el puntero esté sobre él.

```
<a href="ejemplo.org" title="hola">Mi página</a>
```

```
a:hover{  
    color: red  
}
```

Mi página

Mi página

Unidades de medida

- Las medidas en CSS se emplean, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto. Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida). Tenemos:

Unidades absolutas

Unidades relativas

Unidades absolutas

- Se utiliza el valor directamente, sin necesidad de realizar cálculos intermedios. Son muy poco flexibles, así tenemos:

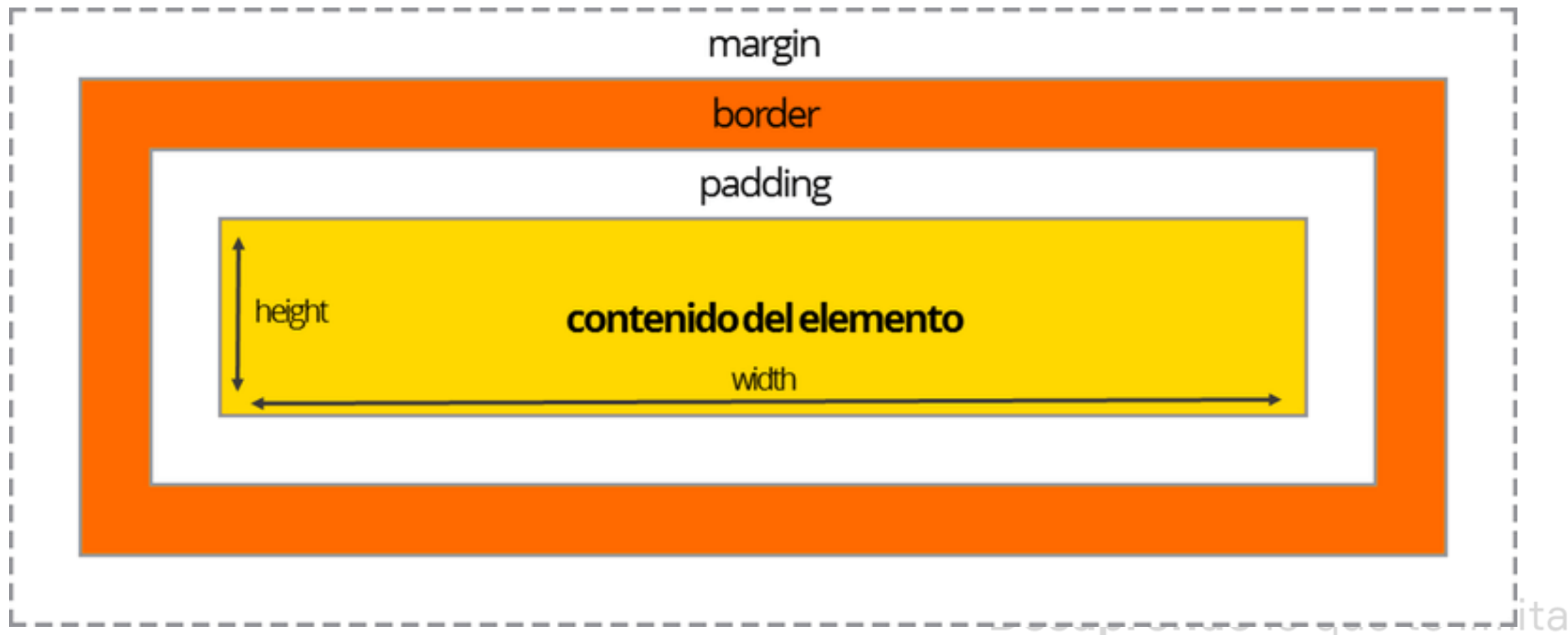
| Representación | Unidad de medida | Descripción |
|----------------|------------------|---|
| in | pulgadas | Una pulgada equivale a 2.54 centímetros. |
| cm | centímetros | |
| mm | milímetros | |
| pt | puntos | Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros. |
| pc | picas | Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros |

Unidades relativas

- No están completamente definidas, ya que su valor está referenciado a otro valor. Son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.

| Representación | Descripción |
|----------------|---|
| em | Relativa al tamaño de letra del elemento (no confundir con la etiqueta de HTML) es decir, si un elemento tiene un tamaño de fuente de 16 pixeles, entonces 1 em será igual a 16px. |
| rem | Es un em basado en la raíz, si el elemento HTML tiene un tamaño de fuente de 16px, entonces 1rem, sería igual a 16px. |
| ex | Relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento. |
| px | Relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML. Representa pixeles |
| % | Está formado por un valor numérico seguido del símbolo %, de manera predeterminada cada elemento de bloque usa un ancho del 100%. |

Propiedades CSS comunes a todos los elementos



Propiedades básicas

- **Maquetación básica**

| Propiedad | Descripción |
|----------------|---|
| width | Ancho de un elemento. |
| height | Alto de un elemento. |
| vertical-align | Alineamiento vertical dentro de un elemento. |
| margin | Espacio que se añade entre el elemento y sus vecinos. Se puede diferenciar por lado (arriba, abajo, izquierda, derecha). |
| padding | Relleno interior que se añade en los bordes del contenido del elemento, a diferencia de margin, cuenta para el tamaño del elemento. |
| float | Mueve el elemento todo lo posible hacia el lado indicado. Esta propiedad se usa en el posicionamiento flotante de CSS. El tema del posicionamiento en CSS no es trivial y conviene estudiar cómo funciona antes de usar esta propiedad. |

Propiedades básicas

- Fuentes y texto**

| Propiedad | Descripción |
|-----------------|--|
| font-family | Tipo de letra |
| font-size | Tamaño de letra |
| font-weight | Peso (normal, negrita, ...) |
| font-style | Estilo (normal, cursiva, ...) |
| text-decoration | “Decoraciones” como subrayado, tachado, etc. |
| text-align | Alineación del texto (izquierda, derecha, etc.) |
| text-transform | Mostrar un texto en mayúsculas, minúsculas o la primera letra de cada palabra en mayúsculas. |

Propiedades básicas

- **Color y fondos**

| Propiedad | Descripción |
|-------------------|---|
| color | Color del elemento. Se puede especificar en diferentes formatos como palabras predefinidas (red, green, etc.) RGB o como valor hexadecimal. |
| background-color | Color del fondo del elemento. |
| background-image | Permite especificar una imagen de fondo. |
| background-repeat | Permite usar una imagen a modo de mosaico en diferentes modalidades. |
| box-shadow | Crear un efecto de sombra para un elemento. |

Propiedades básicas

- **Listas**

| Propiedad | Descripción |
|------------------|--|
| list-style-image | Usar la imagen especificada como viñeta para la lista. |
| list-style-type | Diferentes estilos de viñetas y estilos de numeración para elementos de lista. |

Propiedades básicas

- Borde**

| Propiedad | Descripción |
|---------------|--|
| border | Añade un borde a un elemento y establece algunas propiedades (grosor, estilo de línea, etc.) |
| border-color | Color del borde. |
| border-style | Diferentes estilos para el borde (sólido, puntos, etc.) |
| border-radius | Permite crear esquinas redondeadas para un elemento. |

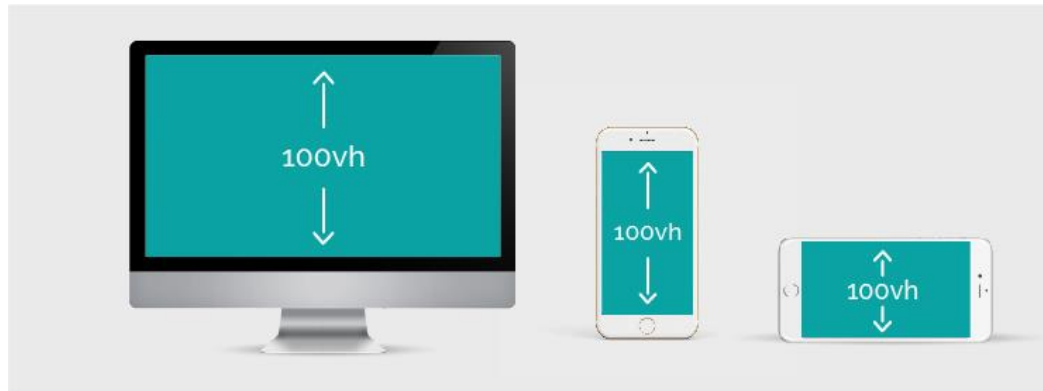
Unidades Viewport Height (VH) y Viewport Width (VW)

- Las medidas vh y vw son medidas relativas de acuerdo al viewport. Vh hace referencia a la centésima parte de la altura del viewport y vw a la centésima parte del ancho del viewport.
 - $1vh = 1\%$ de la altura del viewport
 - $100vh =$ altura del viewport
 - $1vw = 1\%$ del ancho del viewport
 - $100vw =$ ancho del viewport

Unidades Viewport Height (VH) y Viewport Width (VW)

- Estas medidas tienen múltiples usos, por ejemplo cuando queremos tener una imagen de fondo que ocupe todo el ancho y alto de nuestra pantalla.

vh



VW



Desaprende lo que te limita

Unidades Viewport Height (VH) y Viewport Width (VW)

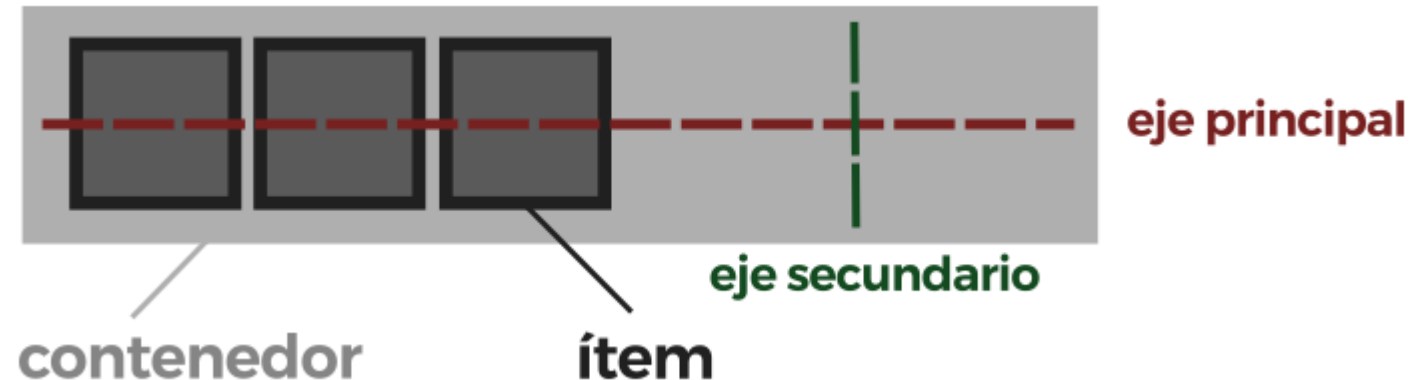
- Las unidades de viewport pueden parecer similares a los porcentajes, pero son muy diferentes. En el caso de porcentajes, el ancho o alto del elemento hijo es determinado con respecto a su padre.



Desaprende lo que te limita

Flexbox

- Permite un contexto flexible para todos sus hijos directos.
- **Flexbox** es un sistema de elementos flexibles que fue diseñado como un modelo unidimensional de layout, y como un método que pueda ayudar a distribuir el espacio entre los **ítems** de un **contenedor** y mejorar las capacidades de alineación.
- Entre sus elementos básico se tiene:



Flexbox - Contenedor

- Al asignar el valor flex o inline-flex a la propiedad display, los hijos directos de este contenedor se convierten en ítems flex.

```
.contenedor{  
  display: flex;  
}
```

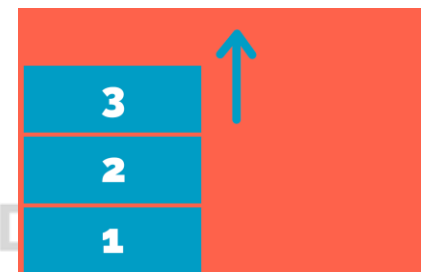
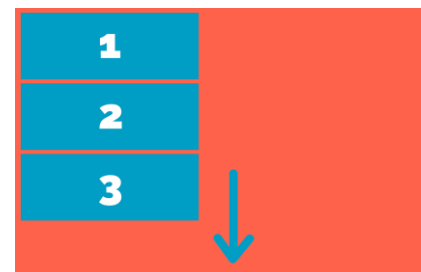
Propiedades del contenedor

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

flex-direction

Valores

- **row:** de izquierda a derecha.
- **row-reverse:** de derecha a izquierda.
- **column:** de arriba a abajo.
- **column-reverse:** de arriba a abajo.



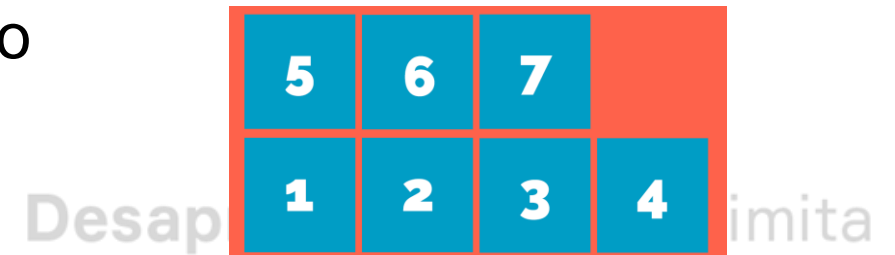
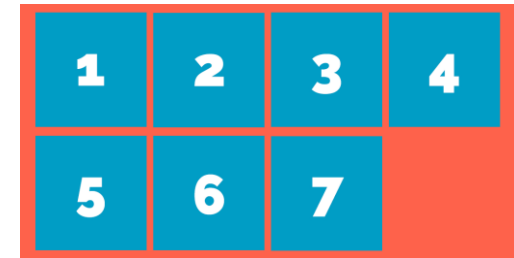
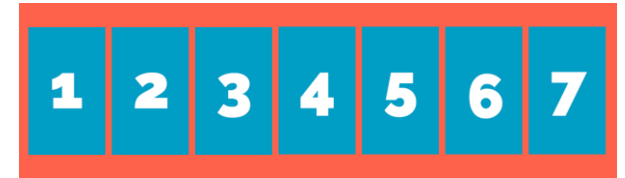
que te limita

flex-wrap

- Por defecto flexbox alinea todos los items en una única fila. flex-wrap muestra a los items en múltiples líneas.

Valores

- **no-wrap:** todos los items en una única línea. Con esta propiedad los ítems podrían salirse del margen.
- **wrap:** reparte a los items en varias líneas.
- **wrap-reverse:** se comporta igual que wrap pero los items se colocarán de arriba a abajo



Desap

imita

flex-flow

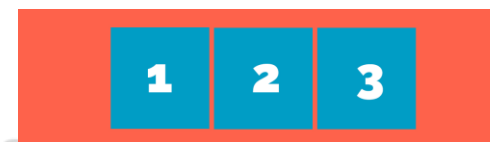
- Esta propiedad es una forma rápida para establecer las propiedades **flex-direction** y **flex-wrap**. El valor por defecto es “row nowrap”

justify-content

- La propiedad justify-content es usada para alinear los ítems a lo largo del eje principal.

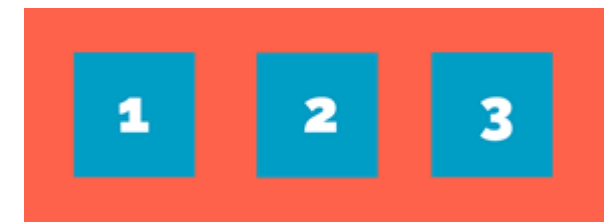
Valores

- flex-start:** alinea los ítems al inicio del margen del contenedor.
- flex-end:** alinea los ítems al final.
- center:** para alinearlos al centro.



justify-content

- **space-between:** distribuye los ítems para que haya un espacio equitativo entre cada ítem.
- **space-around:** crea un espacio equitativo a la derecha e izquierda de cada ítem.
- **space-evenly:** el espacio entre elementos y los márgenes del contenedor es equitativo.

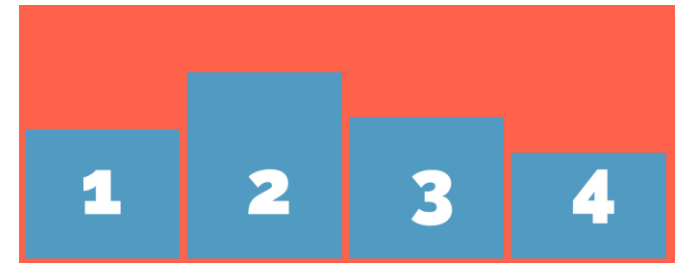
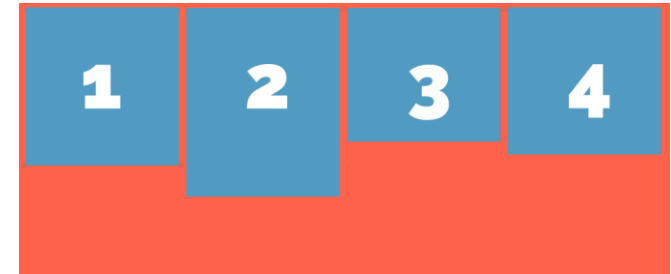


align-items

- La propiedad align-items es usada para alinear los ítems en el eje secundario. Normalmente se usa cuando hay una única línea (cuándo hay más de una, se debe utilizar align-content).

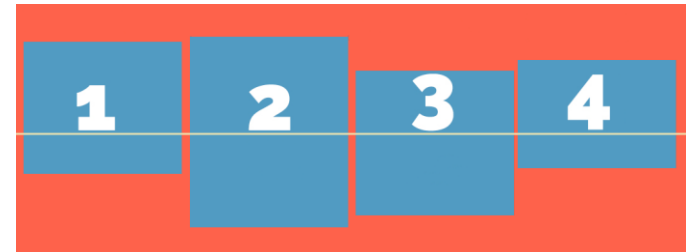
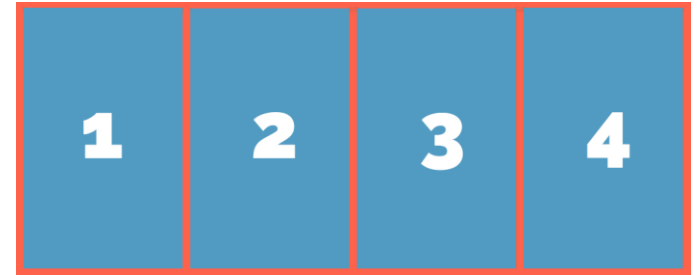
Valores

- flex-start:** indica que se posicionarán al comienzo del eje secundario.
- flex-end:** se posicionarán al final del eje secundario.



align-items

- **center:** se posicionarán en el centro del eje secundario.
- **stretch:** ocuparán el tamaño total del eje secundario (a no ser que hayamos marcado que esos elementos tengan un tamaño diferente).
- **baseline:** para el posicionamiento de los elementos se tendrá en cuenta el texto que hay escrito dentro.



align-content

- La propiedad align-content ajusta las líneas dentro de un contenedor flex cuando hay espacio extra en el eje transversal. Esta propiedad no tiene efecto en cajas flexibles de una sola línea.

Valores

- flex-start:** indica que las filas se colocarán todas pegadas entre sí (obviamente no aparecerán exactamente pegadas si le hemos colocado un margin), desde el inicio del eje secundario.
- flex-end:** las filas se colocarán pegadas entre sí, pero esta vez pegadas al final del eje secundario.

| | | | |
|---|----|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |
| | | | |

| | | | |
|---|----|---|---|
| | | | |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |

align-content

- **center:** se posicionarán en el centro del eje secundario, pegadas entre sí.
- **stretch:** Sus dimensiones crecerán para ocupar todo el espacio disponible (a no ser que se haya colocado una dimensión diferente en los elementos).
- **space-between:** indica que las filas se separarán entre sí, dejando un espacio proporcional entre ellas.
- **space-around:** indica que las filas se separarán, dejando un espacio entre ellas proporcional, también con el borde.

| | | | |
|---|----|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |

| | | | |
|---|----|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |

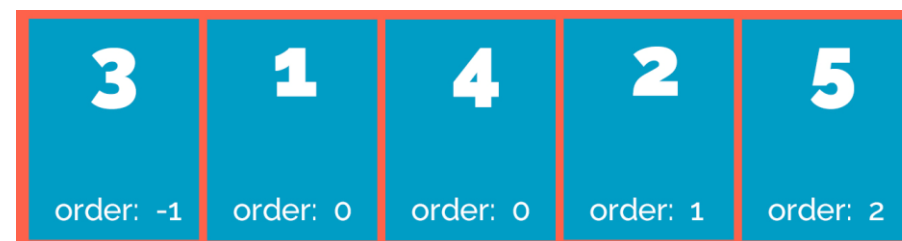
| | | | |
|---|----|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |

| | | | |
|---|----|---|---|
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 | 10 | | |

Propiedades de los items

- **Order:** Con esta propiedad controlamos el orden de los **elementos - flex** que están dentro del **contenedor - flex**.

```
.hijo {  
  order: <integer>;  
}
```



- **flex-grow:** Esta propiedad especifica el factor que determina hasta que punto un elemento - flex crecerá en relación con el resto de los elementos - flex.

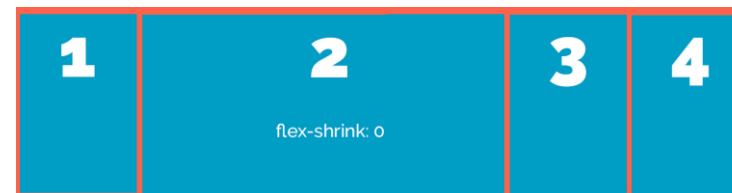
```
.hijo {  
  flex-grow: <number>;  
}
```



Propiedades de los items

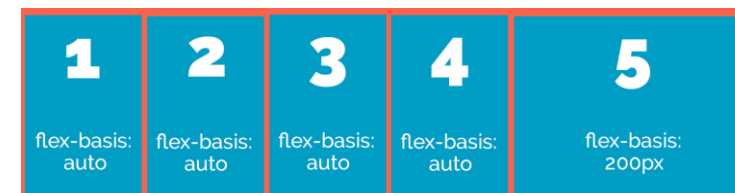
- **flex-shrink:** Esta propiedad especifica cómo el **elemento - flex** se reducirá con respecto al resto de los elementos flexibles dentro del mismo contenedor.

```
.hijo{  
  flex-shrink: <number>;  
}
```



- **flex-basis:** Esta propiedad tiene los mismos valores que las propiedades width y height y especifica el tamaño principal del elemento - flex, distribuyéndose de acuerdo con los factores flex.

```
.hijo {  
  flex-basis: auto | <width>;  
}
```



Propiedades de los items

- **align-self:** Esta propiedad permite la alineación por defecto (o la específica por align-items) para ser anulados por elementos - flex individuales.

```
.hijo {  
  align-self: auto | flex-start | flex-end | center | baseline | stretch;  
}
```



Desaprende lo que te limita

Aplicaciones

*Implementar en el laboratorio
lo explicado en clase*

Resumiendo...

- CSS es _____.
- Las tres formas como se pueden incluir CSS en una archivo web es: _____, _____ y _____.
- La forma mas recomendada de incluir código CSS es _____.
- Mencione 4 unidades de medidas que se usa en CSS _____.



**Universidad
Tecnológica
del Perú**