

Frontend I

Qué es?

El Frontend es la implementación de una interfaz para que un usuario pueda interactuar con una información.

el Backend se encarga de conectar esta interfaz con un servidor para que el usuario pueda acceder a la información



Frontend Backend



Herramientas

Las herramientas del programador Frontend son:



HyperText Markup Language o HTML es una estructura para documentos, que soporta hipertexto. Este nace en 1991 de manos de Tim Bernes-Lee del CERN como un sistema hipertexto con el único objetivo de servir como medio de transmisión de información entre los científicos que se ocupaban de la Física de alta energía ,como parte de la iniciativa World Wide Web.

El Cascading Style Sheets o CSS es una herramienta que nos permite aplicar estilos a la información estructurada en un HTML. CSS fue propuesto por primera vez por Håkon Wium Lie el 10 de octubre de 1994, quien trabajaba con Tim Berners-Lee en el CERN cuando surgió el HTML



Javascript es un lenguaje de programación que nos permite darle dinamismo a una página creada en HTML y CSS. JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript.

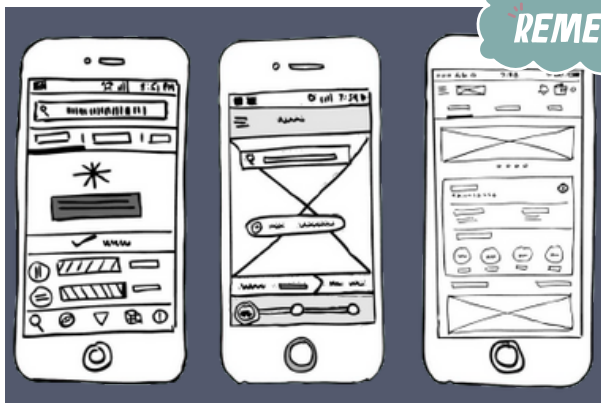


Cómo se hace una interfaz?

Al iniciar el desarrollo de una interfaz debemos estructurar el trabajo; primero con un boceto de los elementos y su flujo y luego la estructura de archivos o carpetas que utilizaremos

Wireframes

Los wireframes son bocetos donde podemos ubicar o maquetar los elementos de una interfaz. Hay dos tipos: de baja y de alta fidelidad.



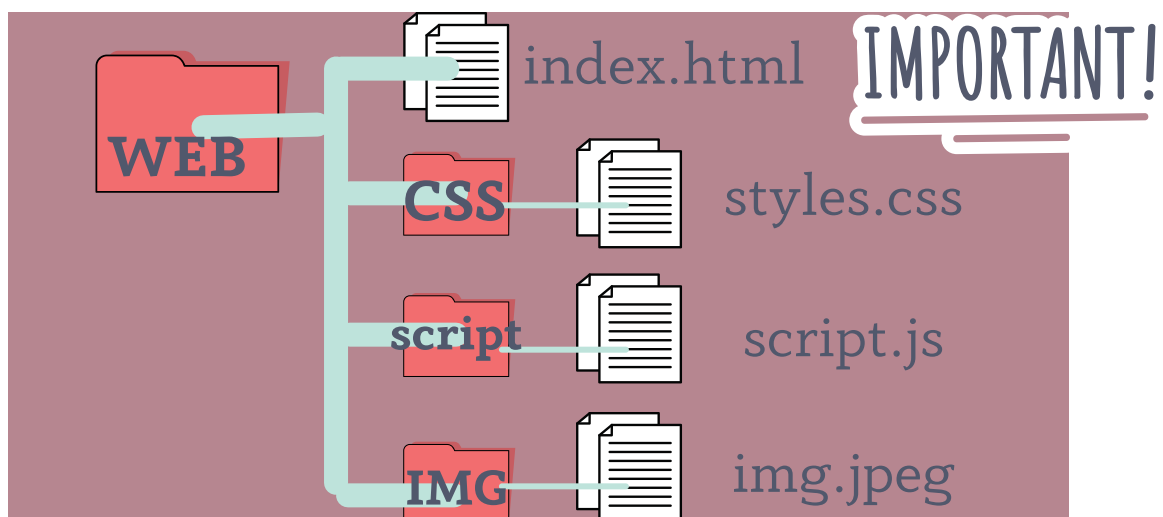
REMEMBER

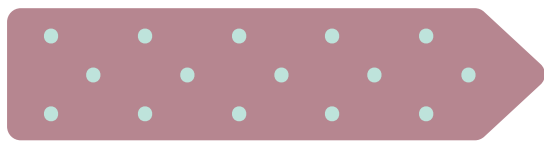
Para realizar estos bocetos podemos utilizar papel o software, como Figma.



Estructura de carpetas

Un proyecto digital debe seguir una estructura de carpetas o archivos. De manera que, el archivo index.html siempre debe ser el archivo principal. Mientras que, los style.css, los scripts.js y las imágenes deben ir en carpetas.





html

HTML es un lenguaje de marcado que se estructura gracias a elementos llamados TAG o etiquetas. HTML se compone de distintos elementos:

- **Bloque:** estos ocupan todo el espacio de su elemento padre o contenedor
- **Línea:** estos ocupan solo el espacio de su etiqueta.



Utilizamos Visual Code Studio para escribir nuestro código



Anatomía html

REMEMBER!

```
<!doctype html>
<html lang="es">
  <head>
    <title>Titulo de documento</title>
    ... otras etiquetas
  </head>
  <body>
    <p class="parrafo"> Este es un parrafo </p>
    ... otras etiquetas
  </body>
</html>
```

- **!DOCTYPE HTML:** regla para considerar un doc como HTML
- **HTML:** elemento raíz que envuelve el contenido del doc.
- **Head:** contenedor de <meta> y <title>, es decir, define la información que se le da al navegador, como el título de la página o los links para cargar los estilos asociados.
- **Body:** contenedor de la información que se muestra a los usuarios como el texto, e imágenes.



Comentarios

<!--	Código HTML o Contenido	-->
------	-------------------------	-----

Inicio del comentario

COMENTARIO

Fin del comentario



Tags

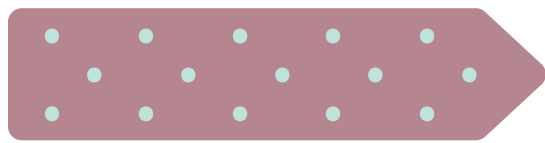


Más etiquetas...



IMPORTANT!

- **Header:** indica el encabezado del documento
- **Main:** rodea el tema central del documento
- **Nav:** envuelve los principales enlaces de navegación
- **Section:** agrupa el contenido relacionado semánticamente.
- **Article:** envuelve el contenido independiente relacionado.
- **Footer:** Pie de página
- **Div:** relaciona grupos de contenido sin relación



CSS

Hojas de Estilo en Cascada (Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML. Hay tres formas de conectar el CSS al HTML:

Inline CSS

```
<p style="color: blue;">This is a paragraph.</p>
```

REMEMBER!

Internal CSS

```
<head>
  <style type = text/css>
    body {background-color: blue;}
    p { color: yellow;}
  </style>
</head>
```

External CSS

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



Regla css

- Los selectores definen sobre qué elementos se aplicará un conjunto de reglas CSS

IMPORTANT!

selector {

propiedad : valor ;

propiedad : valor ;

propiedad : valor ;

}

— una regla por línea
terminada con punto y coma

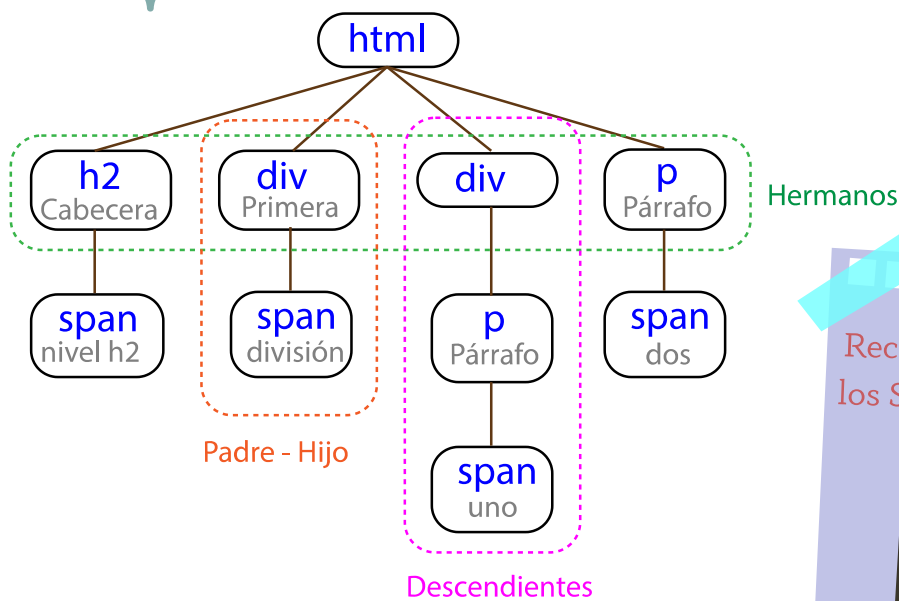
└ terminación de regla
opcional (al ser la última)

Tipos de Selectores

Los selectores se clasifican en Tipo(<>), Clase(.), ID(#), Atributo (attr) y Universales(*). Estos se combinan según la relación que tengan. Además, existen los **pseudoselectores** (::) o elementos que hacen referencia a partes de la estructura pero, fuera del HTML.



Ejemplo Selectores



IMPORTANT!

Universal Selector * {}

ID Selector #id {}

Class Selector .class {}

Type Selector h1, h2, h3 {}

Adjacent Sibling Selector h1 + p {}

Child Selector ul > li {}

General Sibling Selector h1 ~ p {}

Descendant Selector p a {}

Attribute Selector
div[attribute="SomeValue"] {}



Fuentes

SERIF

E Extremos con serifa o remate

TRANSMITE
Modernidad
Pureza
Claridad
Estabilidad
Minimalismo

FUENTES
Times New Roman
Garamond
Courier
Abhaya Libre
Maitree

EJEMPLO EN EL MERCADO: Google

SAN SERIF

E Terminación limpia y recta

TRANSMITE
Modernidad
Pureza
Claridad
Estabilidad
Minimalismo

FUENTES
Helvetica
BEBAS NEUE
Archivo Narrow
Questiol
Roboto

EJEMPLO EN EL MERCADO: facebook

SCRIPT

E Inclínación
Sectores con trazos más finos

TRANSMITE
Diversión
Elegancia
Creatividad
Intriga
Seducción

FUENTES
Euphoria
Playlist
Brush
Tobler
Justicia

EJEMPLO EN EL MERCADO: Cadillac

tamaños

pt	px	em	%
6pt	8px	0.5em	50%
7pt	9px	0.55em	55%
8pt	11px	0.7em	70%
9pt	12px	0.75em	75%
10pt	13px	0.8em	80%
11pt	15px	0.95em	95%
12pt	16px	1em	100%
13pt	17px	1.05em	105%
14pt	19px	1.2em	120%
15pt	21px	1.3em	130%
16pt	22px	1.4em	140%
17pt	23px	1.45em	145%
18pt	24px	1.5em	150%
20pt	26px	1.6em	160%

Estilos para fuentes

Font style
font-style: normal | italic | oblique

Font Variant
font-variant: normal | small-caps

Font Weight
font-weight: normal | bold | bolder | lighter | 100 - 900

Vertical Alignment
vertical-align: baseline | 10px | sub | super | top | text-top | middle | bottom | text-bottom | initial

Text Transform
text-transform: capitalise | lowercase | uppercase

Font Size
font-size: 12px | 0.8em | 80%

Space Between Characters
letter-spacing: normal | 4px

Line Height
line-height: normal | 3em | 34%

Horizontal Alignment
text-align: left | right | center | justify

Text Align Last
text-align-last: auto | left | right | center | justify | start | end | initial | inherit

Text Decoration
text-decoration: none | underline | overline | line-through

Indent First Line text-indent: 25px

Font Family
font-family: 'Open Sans', sans-serif

Text Justify
text-justify: auto | inter-word | inter-character | none | initial | inherit

Text Overflow
text-overflow: clip | ellipsis | string | initial | inherit

Text Shadow
text-shadow: h-shadow v-shadow blur-radius color | none | initial | inherit



Colores

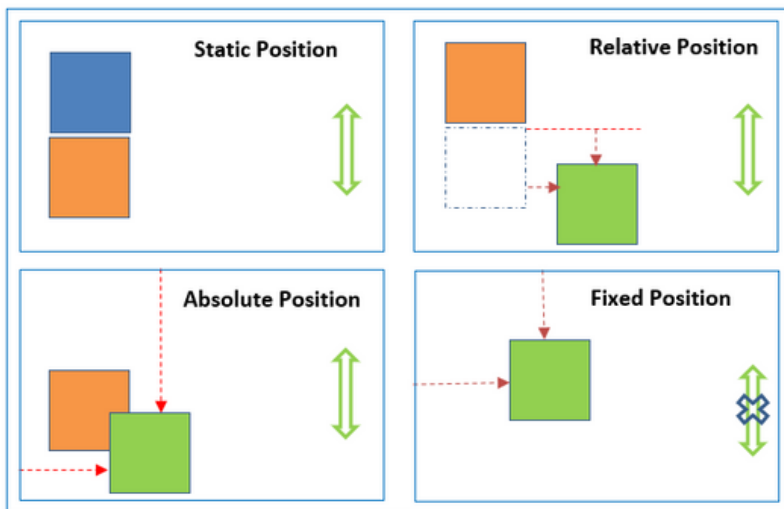
Los colores pueden aplicarse a los elementos HTML mediante dos tipos de códigos, los HEX o hexadecimales, y los RGBA (combinaciones de Rojo, Verde, Azul y Opacidad). Para saber más sobre los colores...

```
<style>
  element { color: code; }
</style>
```



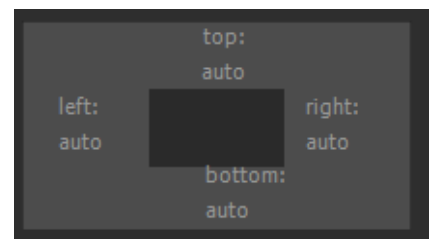
Posición

Los valores de las position en CSS son Absolute, Relative, Static, Fixed. Estas se deben establecer con valores siguiendo un orden específico .

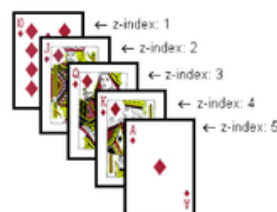


IMPORTANT!

```
<style>
  element {
    position: top, right, bottom,
    left; }
</style>
```



La propiedad z-index nos permite posicionar elementos por capas:



Position

position: static | relative | absolute
| fixed | sticky

Position Properties

top | right | bottom | left

Float Element

float: left | right | none

Clear Floating Elements

clear: none | left | right | both

Z Index

z-index: 3 | auto | inherit



Fondos

La propiedad `background` es un atajo para definir los valores individuales del fondo en una única regla CSS

`background-image : url(directory path);`



`background-size`



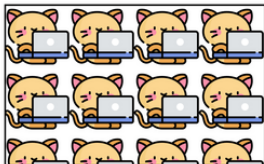
Cover



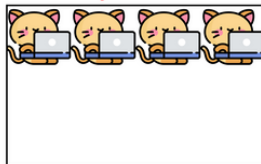
Contain



repeat



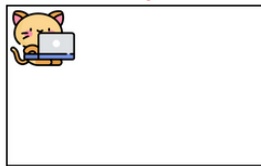
repeat-x



repeat-y



no-repeat



right Y-Axis

bottom
X-Axis



Background Image

`background-image: url();`

Background Repeat

`background-repeat: repeat-x | repeat-y | repeat | space | round | no-repeat`

Background Attachment

`background-attachment: scroll | fixed | local | initial | inherit`

Background Color

`background-color: #2AA9E0`

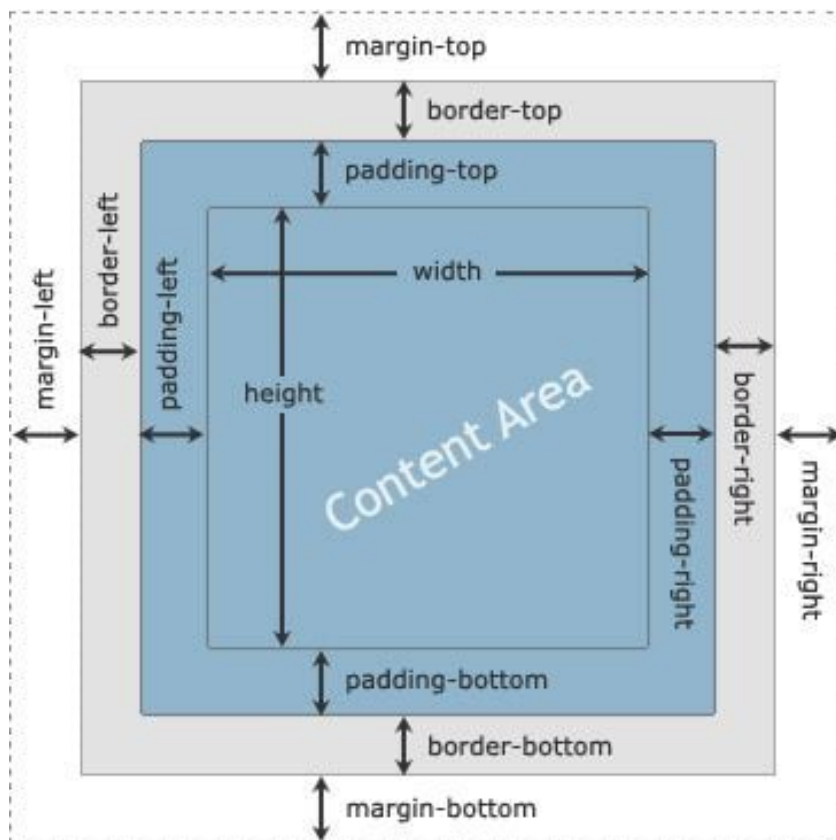
Background Position

`background-position: top | right | bottom | left | center`



Modelo de cajas

El modelo de caja CSS es un módulo CSS que **define cajas rectangulares, incluyendo sus rellenos y márgenes**, que son generadas para los elementos y que se disponen de acuerdo al modelo de formato visual



Box Sizing

`box-sizing: border-box | content-box`

Margin

`margin: 2px 4px 6px 8px | 0 auto`

Padding

`padding: 2px 4px 6px 8px`

Border Color

`border-color: #2AA9E0`

Border Style

`border-style: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset`

Border Width

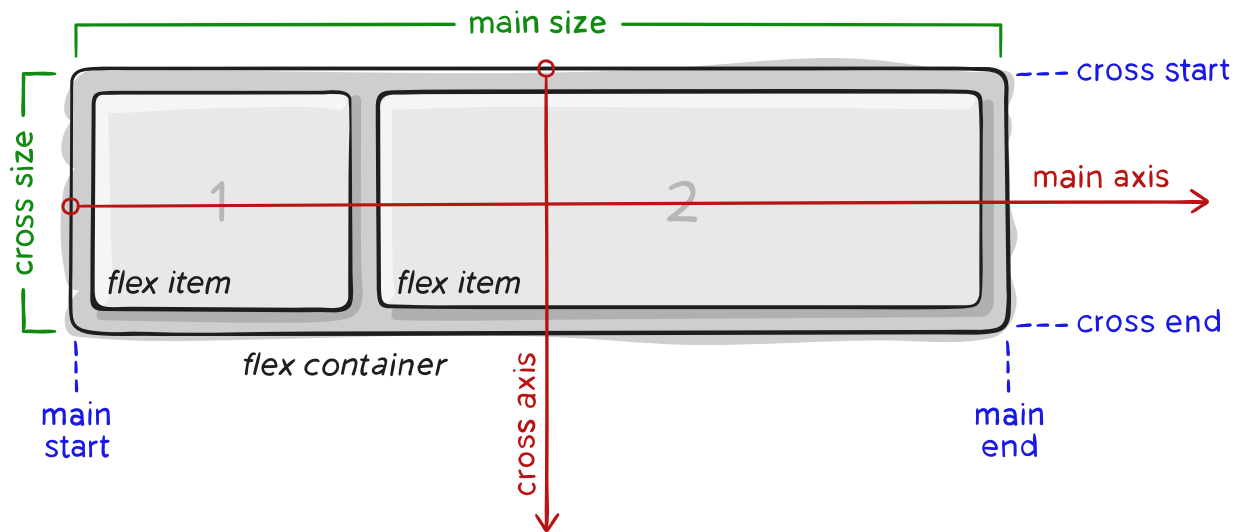
`border-width: 10px`

Este modelo nos permite posicionar los elementos gracias a los contenedores, Gracias al uso de propiedades display como Grid o Flex



Flebox

Diseño de Caja Flexible, comúnmente conocido como Flexbox, es un modelo de diseño- Este es un módulo de diseño de CSS3 que, se creó para mejorar la forma en la que se hace diseño responsive



IMPORTANT!

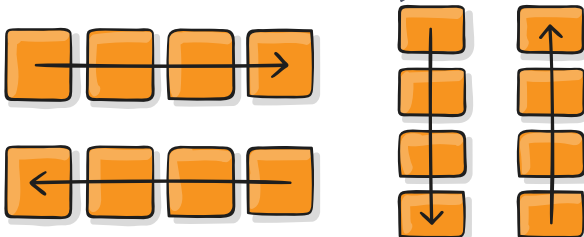
```
<style>  
  element {  
    display: flex;  
  }  
</style>
```

Diseño de Caja Flexible o Flex se inicia al darle la propiedad `display` a un elemento padre, De esta forma sus hijos se alinean automáticamente

container



Flex-direction: row, column...

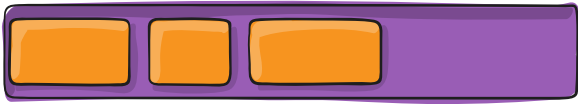


Order



Justify-content

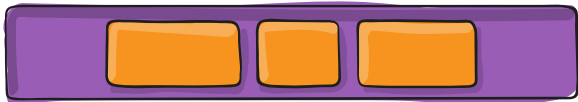
flex-start



flex-end



center



space-between



space-around

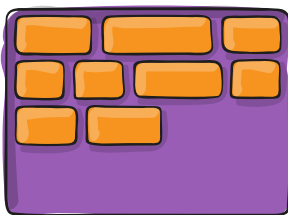


space-evenly

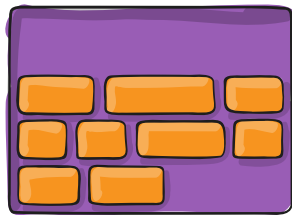


align-content

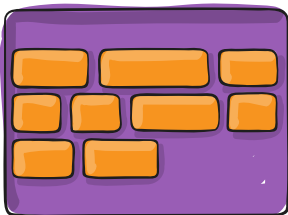
flex-start



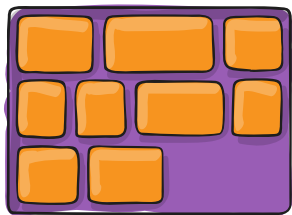
flex-end



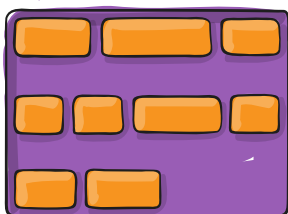
center



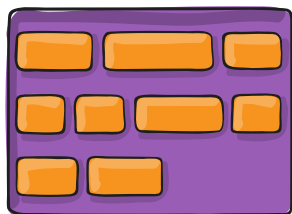
stretch



space-between

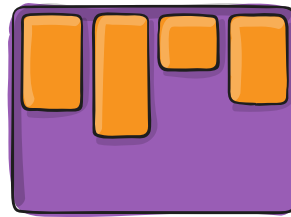


space-around

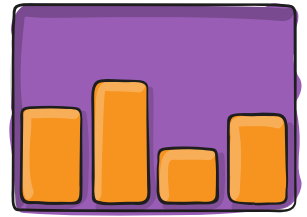


Align-items

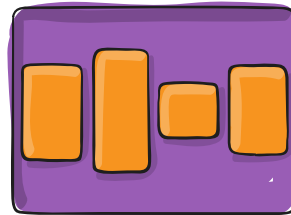
flex-start



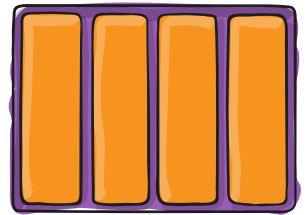
flex-end



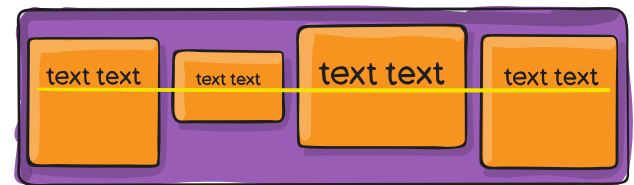
center



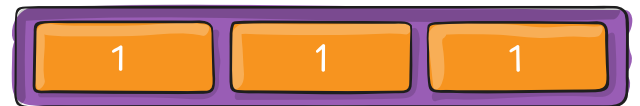
stretch



baseline

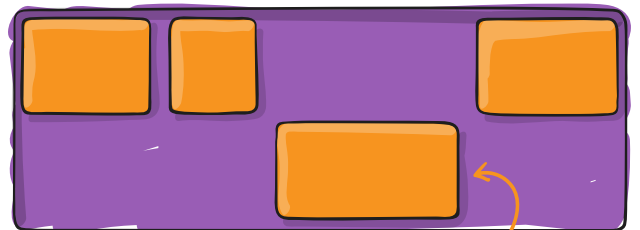


Flex-grow



align-self

flex-start



flex-end

Flex-wrap

