

# *Dicas CSS*

# Seletor Universal

Já vimos esta aplicação no *CCS Reset*.

Para que uma regra CSS seja aplicada a todos elementos da página, usamos o seletor \*

**\*{margin: 0 auto}**

# *Grupamento de Seletores*

Existem muitas formas de selecionar elementos para aplicar regras de forma específica, além de classes e ID's e inclusive associando estas formas, **compartilhando uma mesma regra.**

# Grupamento de Seletores

Para isto, separamos os seletores com virgula:

```
#texto-1, #texto-2, #texto-3{  
    max-width:33%;  
    margin-top:10px;  
}
```

# *Especificidade*

Podemos criar especificidades para selecionar elementos e aplicar regras CSS.

Veja um exemplo:



# Especificidade

```
#imagens figcaption{  
    font-weight:bold;  
}
```

Aqui estamos dizendo que a regra vale somente para toda **tag figcaption** que estiver somente dentro do elemento com a **id #imagens**.

# *Especificidade*

Separamos com espaços para isto, sendo que os últimos seletores declarados são os mais internos no html.



# *Diretiva* ***!important***

Esta diretiva faz com que a regra que a contém prevaleça, independente de especificidade, herança ou efeito cascata.

**Exemplo:** na regra ao lado qual irá prevalecer?

```
h1 {  
    color:red;  
}  
  
h1 {  
    color:yellow;  
}
```

**Exemplo:** a digitada por último, ou seja a mais perto do HTML segundo o efeito cascata (cor amarela).

```
h1{  
    color:red;  
}  
  
h1{  
    color:yellow;  
}
```

**Exemplo:** aplicamos a diretiva como mostrado ao lado, para anular e forçar que a outra regra prevaleça.

```
h1{  
    color:red !important;  
}  
  
h1{  
    color:yellow;  
}
```

# ***Propriedades Úteis para Diagramação***



# *display*

Esta propriedade controla a visibilidade e como os elementos serão exibidos na tela.

Existem muitos valores e possibilidades para ela.

# *display*

**none:** deixa elementos invisíveis, desativando-os. Muito usada com as media queries para esconder imagens, links, etc... em telas pequenas (celulares).

```
 img {  
    display: none;  
}
```

Uma coisa importante a ressaltar que a propriedade ***display: none*** não é a mesma coisa da propriedade ***visibility: hidden***. Nessa última o elemento não aparece na tela mas é renderizado na página, ou seja, vemos um espaço vazio no lugar do elemento; já a propriedade ***display: none*** não renderiza o elemento e, o espaço que era ocupado por ele, é ocupado pelo elemento seguinte no fluxo do documento.

# *display*

**block:** os elementos são colocados um abaixo do outro, exatamente como parágrafos, cabeçalhos e listas, ocupando toda linha/largura da tela, a menos que tenham *float* ou outra regra que prevaleça.

É o valor padrão.

# *display*

**inline:** faz com que o elemento seja renderizado dentro do bloco na mesma linha, exatamente como nas *tags a* e *span* por exemplo.



# Flexbox

# Flexbox

**Método responsivo de  
diagramar websites.**

# Flexbox

Para usá-lo, basta colocar seus elementos (divisões ou seções) em uma seção ou divisão pai (container por exemplo):

# Flexbox

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
</div>
```

# Flexbox

Após colocar larguras e alturas e cores para estes elementos, ou mesmo conteúdo, use a propriedade ***display*** com o valor ***flex*** para iniciar as configurações.

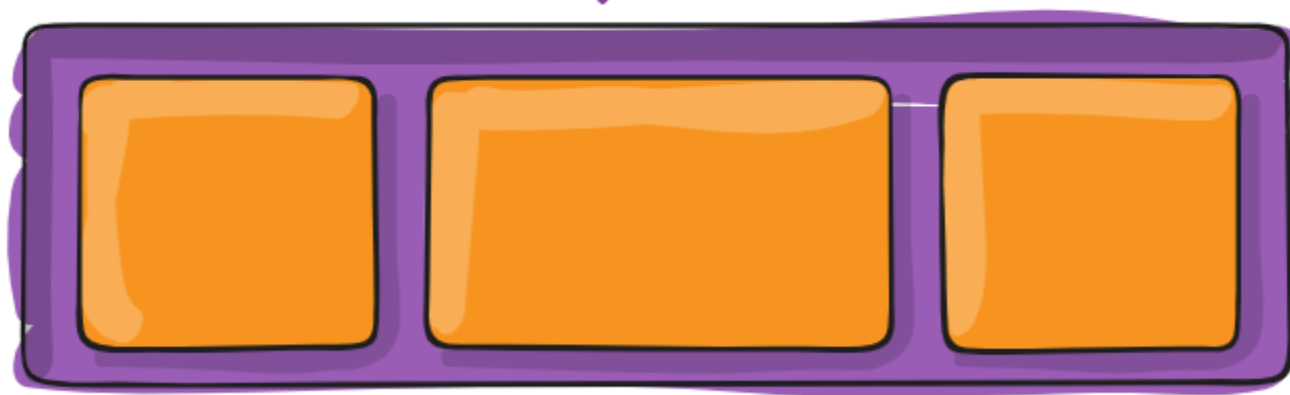


# Flexbox

```
.flex-container {  
  display: flex;  
}
```

# Flexbox

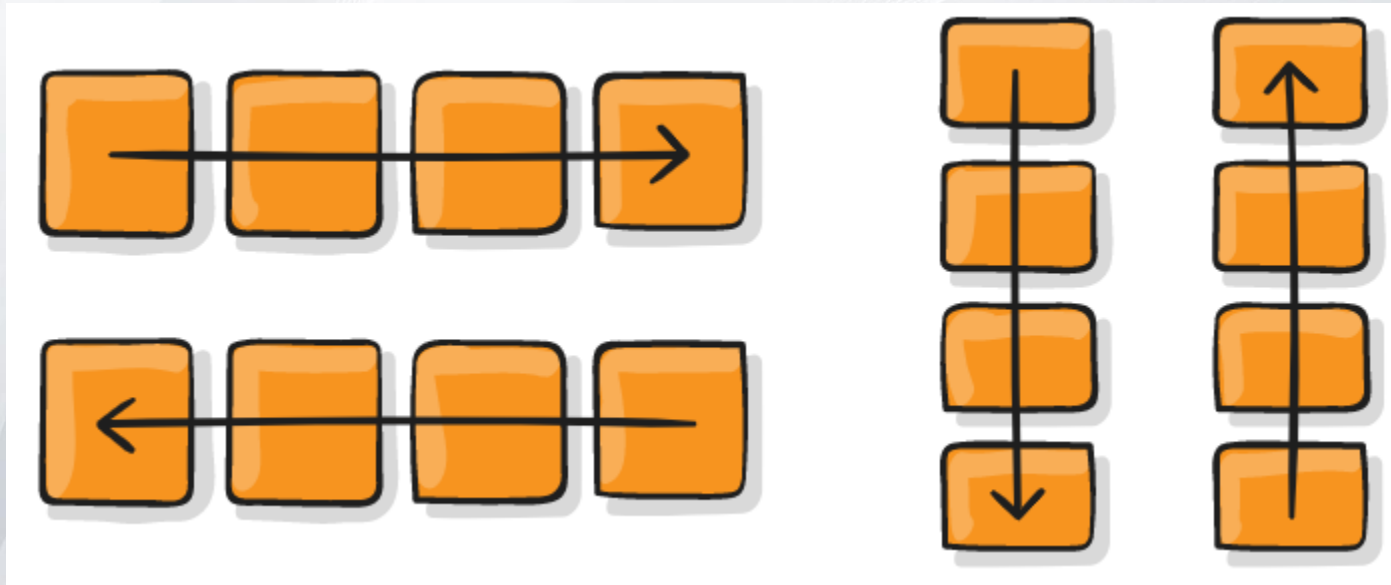
container



# Propriedades e Recursos

# flex-direction

Estabelece o eixo principal, definindo assim a direção em que os *flex items* são alinhados no *flex container*.



# flex-direction

**row (padrão):** esquerda para a direita (*left to right*).

**column:** mesmo que *row*, mas de cima para baixo



# flex-direction

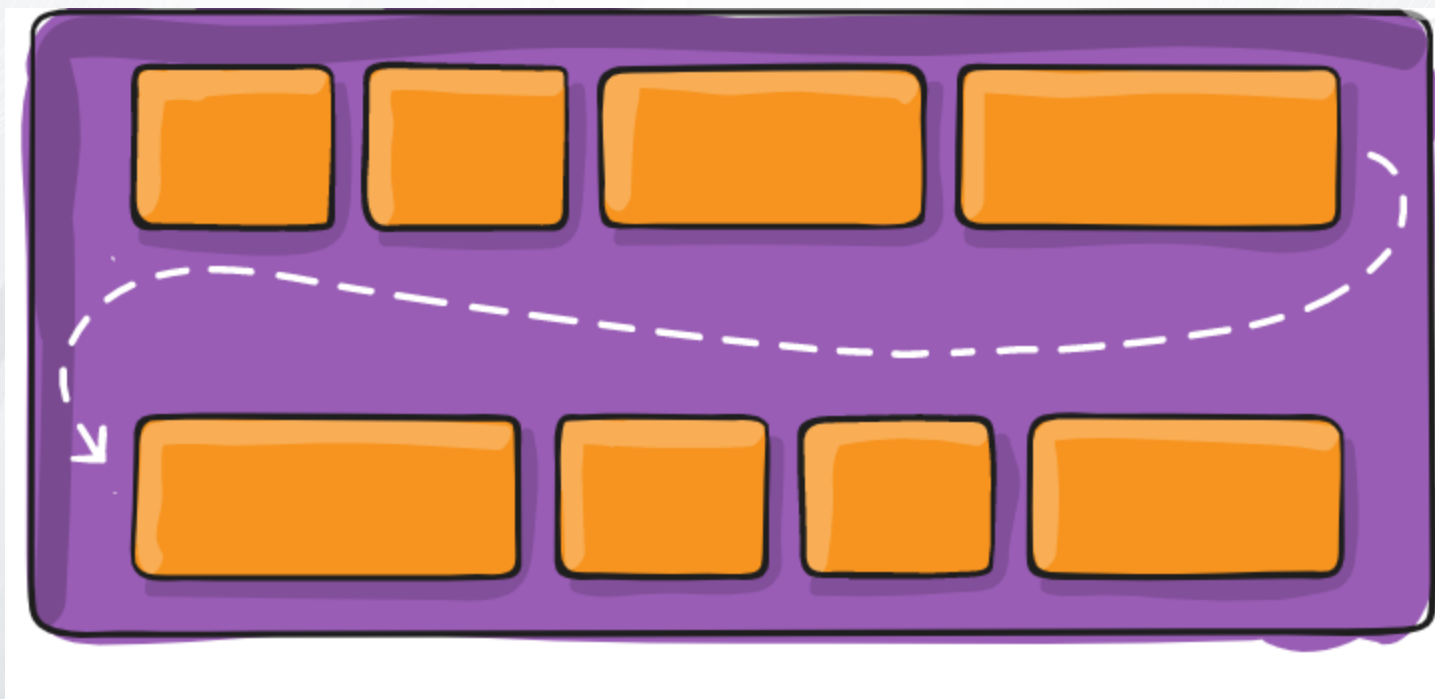
**row-reverse:** direita para a esquerda

**column-reverse:** mesmo que *row-reverse* mas  
de baixo para cima

# flex-wrap

Por padrão, os *flex items* vão todos tentar se encaixar em uma só linha. Com esta propriedade você pode modificar esse comportamento e permitir que os itens quebrem para uma linha seguinte conforme for necessário.

# flex-wrap



# flex-wrap

**nowrap (padrão):** todos os *flex items* ficarão em uma só linha

**wrap:** os *flex items* vão quebrar em múltiplas linhas, de cima para baixo

# justify-content

Esta propriedade define o alinhamento dos itens ao longo do eixo principal. Ajuda a distribuir o espaço livre que sobrar no *container* tanto se todos os *flex items* em uma linha são inflexíveis, ou são flexíveis mas já atingiram seu tamanho máximo.

flex-start



flex-end



center



space-between



space-around



space-evenly





# flex-wrap

**flex-start (padrão):** os itens são alinhados junto à borda de início (*start*) de acordo com qual for a *flex-direction* do container.

**flex-end:** os itens são alinhados junto à borda final (*end*) de acordo com qual for a *flex-direction* do container.

# flex-wrap

**start:** os itens são alinhados junto à borda de início da direção do *writing-mode* (modo de escrita).

**end:** os itens são alinhados junto à borda final da direção do *writing-mode* (modo de escrita).

**left:** os itens são alinhados junto à borda esquerda do container, a não ser que isso não faça sentido com o *flex-direction* que estiver sendo utilizado. Nesse caso, se comporta como *start*.

**right:** os itens são alinhados junto à borda direita do *container*, a não ser que isso não faça sentido com o *flex-direction* que estiver sendo utilizado. Nesse caso, se comporta como *start*.

# flex-wrap

**center:** os itens são centralizados na linha.

**space-between:** os itens são distribuídos de forma igual ao longo da linha; o primeiro item junto à borda inicial da linha, o último junto à borda final da linha.

# flex-wrap

**space-around:** os itens são distribuídos na linha com o mesmo espaçamento entre eles.

**space-evenly:** os itens são distribuídos de forma que o espaçamento entre quaisquer dois itens da linha (incluindo entre os itens e as bordas) seja igual.

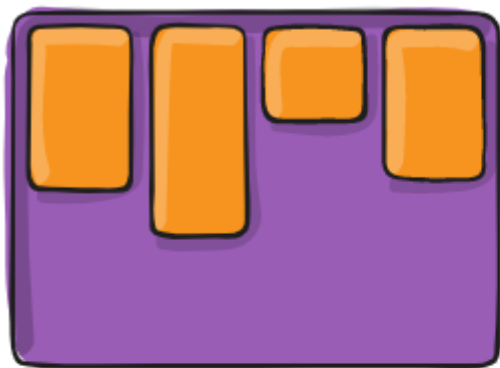


# align-items

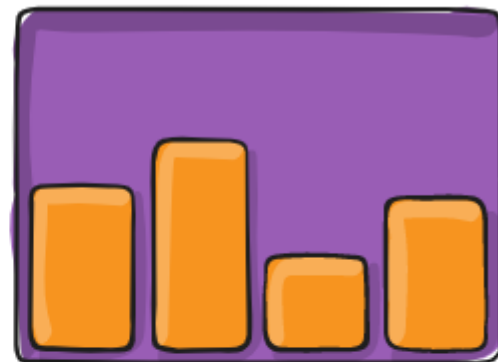
Define o comportamento padrão de como *flex items* são alinhados de acordo com o eixo transversal (*cross axis*). De certa forma, funciona de forma similar ao *justify-content*, porém no eixo transversal (perpendicular ao eixo principal).



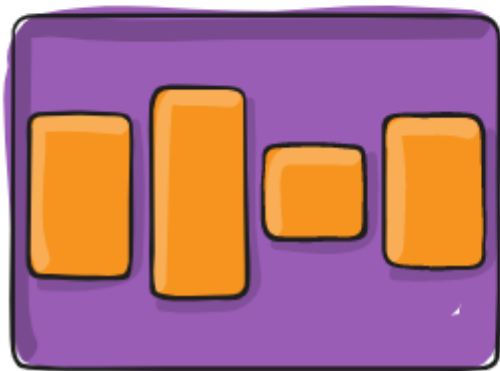
flex-start



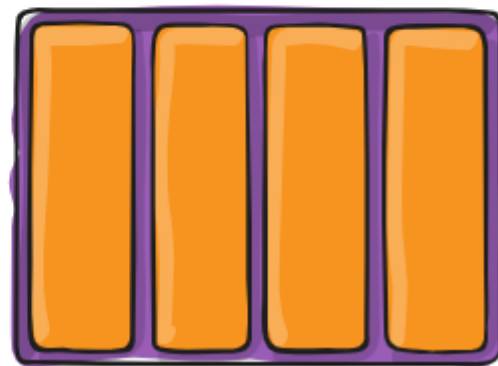
flex-end



center



stretch



# align-items

**stretch (padrão):** estica os itens para preencher o container, respeitando o *min-width/max-width*).

**flex-start/ start / self-start:** itens são posicionados no início do eixo transversal. A diferença entre eles é sutil e diz respeito às regras de *flex-direction* ou *writing-mode*.

# align-itens

**center:** itens são centralizados no eixo transversal.

# Flexbox

Use estes recursos se sua diagramação for simples,  
baseada em linhas e simétricas.