

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>order</title>
<style>

    * {
        box-sizing: border-box;
        font-family: sans-serif;
        font-weight: bold;
    }

    body {
        margin: 0;
        padding: 1vw;
        width: 100vw;
        height: 100vh;
        display: flex;
        justify-content: space-between;
        align-items: center;
        flex-direction: column;
    }

    .container {
        width: 100%;
        height: 100%;
        padding: 5px;
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .bloco {
```

```
width: 50%;
height: 100%;
padding: 2px;
margin: 4px;
box-shadow: 10px 10px 10px rgba(0, 0, 0, .2);
display: flex;
justify-content: center;
flex-direction: column;
}

.bloco > div {
  background-color: lightpink;
}

.div1 {
  background-color: aqua;
  font-size: x-large;
  text-align: center;
}

.div2 {
  background-color: violet;
  font-size: x-large;
  text-align: center;
}

.div3 {
  background-color: rgb(248, 203, 121);
  font-size: x-large;
  text-align: center;
}

.div4 {
  background-color: turquoise;
  font-size: x-large;
  text-align: center;
}
```

```
}

#n0 {
  display: flex;
  height: 80%;
  justify-content: center;
  align-items: center;
}

#n0 > div {
  height: 50%;
  display: flex;
  justify-content: center;
  align-items: center;
  flex: 0 3 50%;
}

#n0 > .div1 {
  order: 3;
}

#n0 > .div2 {
  order: 2;
}

#n0 > .div3 {
  order: 0;
}

#n0 > .div4 {
  order: 1;
}

#n1 {
  display: grid;
  height: 80%;
  justify-items: start;
  align-items: center;
}

#n1 > div {
  height: 50%;
```

```
display: flex;
justify-content: center;
align-items: center;
flex: 0 3 50%;
```

```
}
```

```
#n1 > .div1 {
  order: 1;
```

```
}
```

```
#n1 > .div2 {
  order: 2;
```

```
}
```

```
#n1 > .div3 {
  order: 3;
```

```
}
```

```
#n1 > .div4 {
  order: 0;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<!--
```

Propriedade: order (Ordenar)

Objetivo: Essa propriedade define a ordem de um flex-item em relação aos seus flex-items irmãos.

Ou seja, ela sempre vai ser utilizada dentro de um flex-item e poderá atribuir a ele uma ordem de acordo com a quantidade de elementos que estiverem dentro daquele container. Isso fará com que um flex-item seja posicionado na frente ou atrás de outro flex-item, da maneira que nós quisermos, basta para isso somente acertar a numeração da posição correta.

Para ordenar, essa propriedade usa o esquema de índice, onde o 1º valor corresponde ao número "0" e os demais vão seguindo a ordem a partir do índice 0, o que determina o limite de números é a quantidade de

flex-items que temos para áquele container.

DETALHE IMPORTANTE!!! Sempre deve ser usada em flex-items ou grid-items, nunca deve ser utilizada sobre um elemento container.

ATENÇÃO!!! A ordem vai seguir o fluxo que o nosso flex-item ou grid pré-determinar, seja row ou column. Em um "display: grid" ele vai alterar a ordem dos eixos;

- Valores Aceitáveis:

- number: Número índice que vai pré-determinar a ordem do flex-item;
- unset: Para desligar a propriedade;
- initial: Volta ao valor inicial;
- inherit: Para herdar o valor do elemento pai;

- Valor Padrão: 0;

- Possui valor herdado? - NÃO;

OBS: ;

-->

```
<div class="container">
```

```
  <div class="bloco">
```

```
    order do 1º Elemento: 3<br>
```

```
    order do 2º Elemento: 2<br>
```

```
    order do 3º Elemento: 0<br>
```

```
    order do 4º Elemento: 1<br>
```

```
<span style="font-weight: normal;font-size: 10pt;">Perceba que a ordem serve somente para dentro do container onde o primeiro elemento equivale ao índice "0" e vai seguindo até o 4º elemento que equivale ao índice "3", isso poderia se seguir infinitamente, e poderíamos delimitar qualquer ordem aos elementos segundo a nossa vontade.
```

```
</span>
```

```
<div id="n0">
```

```
  <div class="div1">1º Elemento</div>
```

```
  <div class="div2">2º Elemento</div>
```

```
  <div class="div3">3º Elemento</div>
```

```
  <div class="div4">4º Elemento</div>
```

```
</div>
```

```
<span style="font-weight: bold;font-size: 8pt;">Atenção: Ordem aplicada sobre um flex-row;
```

```
</span>
```

```
</div>
```

```
<div class="bloco">
```

```
  order do 1º Elemento: 1<br>
```

```
  order do 2º Elemento: 2<br>
```

```
  order do 3º Elemento: 3<br>
```

```
  order do 4º Elemento: 0<br>
```

```
<span style="font-weight: normal;font-size: 10pt;">Perceba que usamos o mesmo exemplo que usamos no flexbox agora sendo aplicado em um grid, onde os grid-items que geram os eixos (linhas) vão trocar de posição...
```

```
</span>
```

```
<div id="n1">
```

```
  <div class="div1">1º Elemento</div>
```

```
  <div class="div2">2º Elemento</div>
```

```
  <div class="div3">3º Elemento</div>
```

```
  <div class="div4">4º Elemento</div>
```

```
</div>
```

```
<span style="font-weight: bold;font-size: 8pt;">Atenção: Ordem aplicada sobre um display: grid;
```

```
</span>
```

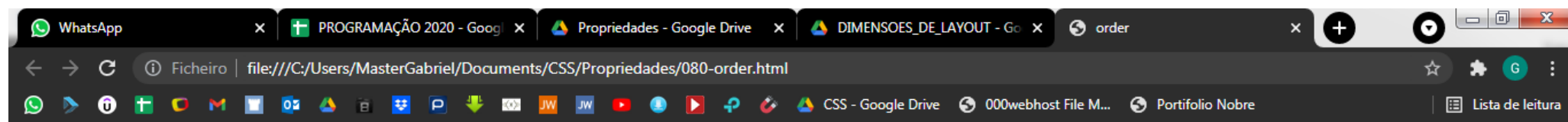
```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

RESULTADO...



order do 1º Elemento: 3
order do 2º Elemento: 2
order do 3º Elemento: 0
order do 4º Elemento: 1

Perceba que a ordem serve somente para dentro do container onde o primeiro elemento equivale ao índice "0" e vai seguindo até o 4º elemento que equivale ao índice "3", isso poderia se seguir infinitamente, e poderíamos delimitar qualquer ordem aos elementos segundo a nossa vontade.



Atenção: Ordem aplicada sobre um flex-row;

order do 1º Elemento: 1
order do 2º Elemento: 2
order do 3º Elemento: 3
order do 4º Elemento: 0

Perceba que usamos o mesmo exemplo que usamos no flexbox agora sendo aplicado em um grid, onde os grid-items que geram os eixos (linhas) vão trocar de posição...



Atenção: Ordem aplicada sobre um display: grid;

