

```
<!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>flex-basis</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: 50%;
        padding: 5px;
        display: flex;
        justify-content: space-between;
        align-items: center;
    }

    .bloco {
```

```
width: 25%;
height: 100%;
padding: 2px;
margin: 4px;
box-shadow: 10px 10px 10px rgba(0, 0, 0, .2);
display: flex;
justify-content: flex-end;
flex-direction: column;
}

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

.div1 {
  background-color: aqua;
  max-height: 50%;
  font-size: x-large;
  text-align: center;
}

.div2 {
  background-color: violet;
  max-height: 50%;
  font-size: x-large;
  text-align: center;
}

.div3 {
  background-color: rgb(248, 203, 121);
  max-height: 50%;
  font-size: x-large;
  text-align: center;
}
```

```
.div4 {
  background-color: turquoise;
  max-height: 50%;
  font-size: x-large;
  text-align: center;
}

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

#length > div {
  flex-basis: 120px;
}

#length > div:hover {
  flex-basis: 150px;
  transition: all .2s ease-in-out;
}

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

#auto > div {
  flex-basis: auto;
}

#auto > div:hover {
  flex-basis: 150px;
  transition: all .2s ease-in-out;
}
```

```
#fill {
  display: flex;
  height: 80%;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}
#fill > div {
  flex-basis: fill;
}
#fill > div:hover {
  flex-basis: 150px;
  transition: all .2s ease-in-out;
}

#max-content {
  display: flex;
  height: 80%;
  justify-content: center;
  align-items: center;
  flex-wrap: wrap;
}
#max-content > div {
  font-size: small;
  flex-basis: max-content;
}
#max-content > div:hover {
  flex-basis: 5px;
  transition: all .2s ease-in-out;
}

#min-content {
  display: flex;
  height: 80%;
  justify-content: center;
```

```
    align-items: center;
    flex-wrap: wrap;
}
#min-content > div {
    font-size: small;
    flex-basis: min-content;
}
#min-content > div:hover {
    flex-basis: 5px;
    transition: all .2s ease-in-out;
}

#fit-content {
    display: flex;
    height: 80%;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
}
#fit-content > div {
    font-size: small;
    flex-basis: fit-content;
}
#fit-content > div:hover {
    flex-basis: 5px;
    transition: all .2s ease-in-out;
}

#content {
    display: flex;
    height: 80%;
    justify-content: center;
    align-items: center;
    flex-wrap: wrap;
}
#content > div {
```

```
    font-size: small;
    flex-basis: content;
  }
  #content > div:hover {
    flex-basis: 150px;
    transition: all .2s ease-in-out;
  }
</style>
</head>
<body>

<!--
```

Propriedade: flex-basis (Flex Base)

Objetivo: Essa propriedade define uma largura base para um ou mais flex-item(s).

O que seria uma largura base?

A largura base seria um comprimento inicial aplicado a um flex-item, ele vai ter que obedecer o comprimento que passamos para ele, nunca poderá ser menor, deverá ser sempre igual ou maior.

Essa propriedade é muito útil para quando desejamos criar um layout adaptado a largura do container, gerando flex-items que vão sempre obedecer ao mesmo estilo de largura não importa o tamanho do nosso container.

DETALHE IMPORTANTE!!! Sempre deve ser usada em flex-items, nunca em display grids, e muito menos nos containers.

- Valores Aceitáveis:

- length: Podemos usar qualquer valor absoluto ou relativo;

- auto: Pegamos o valor automático em relação ao conteúdo;
- fill: Ainda não é reconhecido em alguns navegadores!!! O flex-item vai preencher toda a largura;
- max-content: Ainda não é reconhecido em alguns navegadores!!! O flex-item vai ocupar o espaço do maior conteúdo dentro dele;
- min-content: Ainda não é reconhecido em alguns navegadores!!! O flex-item vai ocupar o espaço do menor conteúdo dentro dele;
- fit-content: Ainda não é reconhecido em alguns navegadores!!! O flex-item vai quebrar o conteúdo até achar o menor conteúdo possível e vai ocupar o espaço dele;
- content: O flex-item vai pegar o tamanho do conteúdo completo;
- unset: Para desligar a propriedade;
- initial: Volta ao valor inicial;
- inherit: Para herdar o valor do elemento pai;

- Valor Padrão: auto;

- Possui valor herdado? - NÃO;

OBS: Caso o nosso flex-item tenha uma largura superior a largura do nosso container, ele vai obedecer só até onde vai a largura do container;

-->

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

```
  <div class="bloco" style="justify-content: center;">
```

Passe o mouse por cima das divs para ver como elas aumentam a largura e depois retraem para a largura base que foi definida.

Veja vários exemplos disso com o uso da propriedade:
flex-basis
nos nossos

```
flex-items<br>&rArr;
</div>

<div class="bloco">
  flex-basis: length
  <span style="font-weight: normal;font-size: 10pt;">Podemos colocar qualquer valor relativo ou absoluto,
    colocamos 120px e um hover de 150px, perceba que enquanto uma div estica a outra não tem o seu tamanho
    afetada.
  </span>
  <div id="length">
    <div class="div1">A</div>
    <div class="div2">B</div>
    <div class="div3">C</div>
    <div class="div4">D</div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados somente em flex-box;</span>
</div>

<div class="bloco">
  flex-basis: auto
  <span style="font-weight: normal;font-size: 10pt;">Por ter o valor "auto" a div pega o tamanho relativo ao
    conteúdo dentro dela...
  </span>
  <div id="auto">
    <div class="div1">A</div>
    <div class="div2">BC</div>
    <div class="div3">C</div>
    <div class="div4">DEF</div>
  </div>
  <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados somente em flex-box;</span>
</div>

<div class="bloco">
  flex-basis: fill
  <span style="font-weight: normal;font-size: 10pt;">Ainda não está funcionando nessa propriedade, por isso não
    preencheu toda a largura disponível, em vez disso, ele copiou o valor "auto" que é o valor padrão...
```



```
</span>
<div id="fill">
  <div class="div1">A</div>
  <div class="div2">B</div>
  <div class="div3">C</div>
  <div class="div4">D</div>
</div>
<span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados somente em flex-box;</span>
</div>
```

```
</div>
```

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

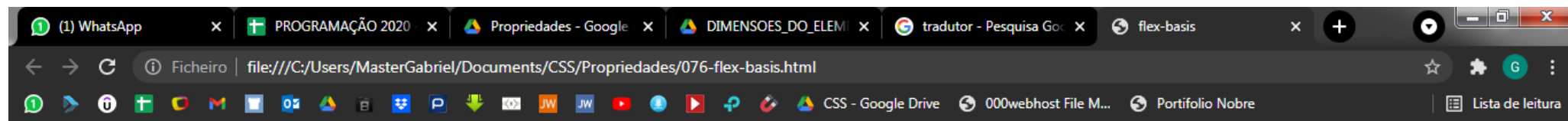
```
  <div class="bloco">
    flex-basis: max-content
    <span style="font-weight: normal;font-size: 10pt;">Ainda não está um funcionamento, ele copia o comportamento
      do valor "content"...
    </span>
    <div id="max-content">
      <div class="div1"><p>Lorem</p></div>
      <div class="div2"><p>Lorem ipsum </p></div>
      <div class="div3"><p>Lorem ipsum dolor, sit.</p></div>
      <div class="div4"><p>Lorem ipsum dolor, sit amet consectetur adipisicing elit.</p></div>
    </div>
    <span style="font-weight: bold;font-size: 8pt;">Atenção: Valores aplicados somente em flex-box;</span>
  </div>
```

```
<div class="bloco">
  flex-basis: min-content
  <span style="font-weight: normal;font-size: 10pt;">Ainda não está um funcionamento, ele copia o comportamento
    do valor "content"...
  </span>
  <div id="min-content">
    <div class="div1"><p>Lorem</p></div>
    <div class="div2"><p>Lorem ipsum </p></div>
```



```
</body>
</html>
```

RESULTADO...



Passe o mouse por cima das divs para ver como elas aumentam a largura e depois retraem para a largura base que foi definida.

Veja vários exemplos disso com o uso da propriedade: **flex-basis** nos nossos flex-itens
⇒

flex-basis: length

Podemos colocar qualquer valor relativo ou absoluto, colocamos 120px e um hover de 150px, perceba que enquanto uma div estica a outra não tem o seu tamanho afetado.



Atenção: Valores aplicados somente em flex-box;

flex-basis: auto

Por ter o valor "auto" a div pega o tamanho relativo ao conteúdo dentro dela...

ABCCDEF

Atenção: Valores aplicados somente em flex-box;

flex-basis: fill

Ainda não está funcionando nessa propriedade, por isso não preencheu toda a largura disponível, em vez disso, ele copiou o valor "auto" que é o valor padrão...

ABCD

Atenção: Valores aplicados somente em flex-box;

flex-basis: max-content

Ainda não está um funcionamento, ele copia o comportamento do valor "content"...

LoremLorem ipsumLorem ipsum dolor, sit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit.

Atenção: Valores aplicados somente em flex-box;

flex-basis: min-content

Ainda não está um funcionamento, ele copia o comportamento do valor "content"...

LoremLorem ipsumLorem ipsum dolor, sit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit.

Atenção: Valores aplicados somente em flex-box;

flex-basis: fit-content

Ainda não está um funcionamento, ele copia o comportamento do valor "content"...

LoremLorem ipsumLorem ipsum dolor, sit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit.

Atenção: Valores aplicados somente em flex-box;

flex-basis: content

Ele vai usar como base o tamanho de todo o conteúdo dentro da div...

LoremLorem ipsumLorem ipsum dolor, sit.

Lorem ipsum dolor, sit amet consectetur adipiscing elit.

Atenção: Valores aplicados somente em flex-box;

