

```
<!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>width</title>
  <style>
    body {
      padding: 0;
    }
    /* Desligamos o padding do nosso documento */

    .fundo {
      width: 100vw;
      height: 100vh;
    }
    /*Essa é a formatação da div que faz wrap sobre as demais divs.*/

    .fundo > div[class^=largura]{
      border-width: 2px; /*Separamos os borders aqui, por que se criássemos um border único ele iria sobrescrever*/
      border-style: solid;
      margin-bottom: 10%;
      height: 40px;
    }
    /*Essa formatação é aplicada sobre todas as divs dentro da div "fundo", mas que tiverem o prefixo "largura" no
      nome de suas classes.*/

    .largura-auto {
      width: auto;
      border-color: #00598daa;
    }
    /*Perceba que quando o width tem a largura "auto" ele herda a largura do elemento pai*/

    .largura-vwvh {
      width: 50vw;
```

```
border-color: #008d44aa;
}
/*Perceba que width com 50% do viewport pega a metade da área que nós temos disponível para visualizar, como a
área que nós temos foi delimitada pelo elemento pai "fundo", o nosso vw vai pegar a área do elemento pai. Mas faça
o teste e desligue a largura do fundo, você vai perceber a diferença entre a div "vwvh" e "percent", quando o
percent respeita as margens da body, o vw desrespeita as margens, sobresaindo-se sobre a div percent. Por que para
ele margens não importam.*/

.largura-percent {
width: 50%;
border-color: red;
}
/* Perceba a diferença entre a div "percent" e a div "vwvh", para isso desligue a largura da div "fundo", você vai
perceber que percent respeita as margens da tag "body", enquanto "vwvh" não respeita, se sobre-saindo sobre a div
"percent" */

.largura-pixels {
width: 350px;
border-color: violet;
}
/* Perceba que quando trabalhamos com pixels, temos larguras totalmente fixas independentes do nosso viewport */

.largura-fit-content {
width: fit-content;
border-color: goldenrod;
}
/*Perceba que com fit-content a largura vai ser atribuída de acordo com a quantidade de conteúdo dentro da div
até que ela ocupe toda a largura da área visível do nosso documento. Se a quantidade de conteúdo dentro de um
elemento ultrapassar a largura da área visível, então o conteúdo gerará overflow*/

.largura-min-content {
width: min-content;
border-color: rgb(0, 247, 255);
}
/*Perceba que o min pega o maior elemento dentro da div, que no caso é a palavra CONTENT (em maiúsculo) e usa esse
elemento como parâmetro para definir a largura do elemento.*/
```

```
.largura-max-content {  
  width: max-content;  
  border-color: gray;  
}
```

/*O max-content agirá parecido ao fit-content, MAS NÃO IGUAL, pois o max-content vai forçar a largura da div para que todo o conteúdo caiba dentro da largura do elemento, gerando até mesmo um scroll para isso.*/

```
</style>
```

```
</head>
```

```
<body>
```

```
<!--
```

Propriedade: width (largura)

Objetivo: Define uma largura para o conteúdo do nosso elemento, ou seja, o width NÃO VAI AFETAR a nossa borda, margem e padding.

Valores possíveis:

- auto: pega a largura automaticamente;
- length: podemos ter um tamanho por através unidades de medida;
- max-content: vai definir a largura de acordo com o conteúdo do elemento, se o elemento tiver vários elementos como conteúdo, o max-content vai fazer o máximo para todos os elementos sejam renderizados dentro do mesmo espaço de largura, ou seja, o elemento vai esticar de acordo com o conteúdo;
- min-content: vai definir a largura de acordo com o maior elemento que estiver como conteúdo de um elemento onde a propriedade min-content foi declarada. Ou seja, ele vai fazer o máximo para não cortar um elemento ao meio se ele transbordar a largura de um elemento;
- fit-content: vai definir a largura de acordo com o conteúdo do elemento, igual ao max-content, se o elemento tiver vários elementos como conteúdo, o fit-content vai fazer o máximo para todos os elementos sejam renderizados dentro do mesmo espaço de largura, ou seja, o elemento vai esticar de acordo com o conteúdo;
- initial: podemos voltar a o valor padrão dessa propriedade;
- inherit: Terá como herança o valor do elemento pai;
- revert: ele reinicia a propriedade, como o initial, mas somente para o navegador Safari;

- unset: Desliga a propriedade;

Valores Padrão:

- auto;

Possuí valor herdado? - NÃO

OBSERVAÇÕES:

- O width possui herança explícita, ou seja, a largura do elemento será herdada sobre os elementos filhos de uma tag que possui width;
- Se houverem as propriedades "min-width" ou "max-width" definidos sobre a mesma tag, eles vão ganhar precedência sobre a propriedade "width";
- CUIDADO COM O DISPLAY BLOCK DO NAVEGADOR! Muitas pessoas acham que a propriedade width tem 100% de largura automática, mas na verdade não é isso, o padrão do nosso width é ter o valor "auto", ou seja, automaticamente ele vai herdar a largura do elemento pai, que no caso é o body, e o body possui por padrão o display block, que ocupa toda a linha do seu elemento. Então cuidado! Não pense que o width vai fazer com que a largura do seu elemento ocupe 100% de uma página, ele vai ficar automático em relação ao elemento pai, seja ele qual for;

-->

```
<div class="fundo">
```

```
<div class="largura-auto">
```

AUTO - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia aperiam consequuntur.

```
</div>
```

```
<div class="largura-vwvh">
```

VWCH - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia aperiam consequuntur.

```
</div>
```

```
<div class="largura-percent">
```

```
    PERCENT - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem  
    natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia  
    aperiam consequuntur.
```

```
</div>
```

```
<div class="largura-pixels">
```

```
    PIXELS - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem  
    natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia  
    aperiam consequuntur.
```

```
</div>
```

```
<div class="largura-fit-content">
```

```
    FIT-CONTENT - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem  
    natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia  
    aperiam consequuntur.
```

```
</div>
```

```
<div class="largura-min-content">
```

```
    MIN-CONTENT - Lorem ipsum dolor sit amet consectetur adipisicing elit.
```

```
</div>
```

```
<div class="largura-max-content">
```

```
    MAX-CONTENT - Lorem ipsum dolor sit amet consectetur adipisicing elit. Vitae, nam, earum quisquam exercitationem  
    natus sapiente hic magni accusamus pariatur veniam omnis ipsam nesciunt dicta alias quae, repellat officia  
    aperiam consequuntur.
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```