

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no">
```

```
<title> </title>
```

```
<link rel="stylesheet" href="css
```

```
<link rel="stylesheet" href="css
```

```
</head>
```

```
<body>
```

```
<div id="container">
```

```
<div id="menu">
```

```
<ul>
```

```
<li><a href="index.html">Home </a> </li>
```

```
<li><a href="opinioes.html">Opiniões </a> </li>
```

```
<li><a href="trechos.html">Trechos </a> </li>
```

```
<li><a href="autor.html">Autor </a> </li>
```

```
<li><a href="contato.html">Contato </a> </li>
```

```
</ul>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width,initial-scale=1,user-scalable=no">
```

```
<title> </title>
```



# Tecnologia Web I

**Gabriela Silveira**

```
<li><a href="index.html">H
```

```
<li><a href="opinioes.html"
```

```
<li><a href="trechos.html">
```

```
<li><a href="autor.html">Au
```

```
<li><a href="contato.html">
```

```
</ul>
```

```
<head>
```

```
<meta charset="U
```

```
<meta name="vie
```

```
<title> </title>
```

```
<link rel="stylesh
```

```
<link rel="stylesh
```



# Responsivo x Fluido

- Ambos se adaptam a diferentes resoluções de tela. A diferença é que o responsivo responde a cada uma, ou seja temos que saber as resoluções que serão usadas e fazer diferentes códigos para cada uma delas. Já o fluido ele se reorganiza automaticamente, as divisões do site se reorganizam para caber no tamanho da tela, pois usa medidas relativas no seu css.





# Responsivo x Fluido

- *Responsivo*: Responde ao tamanho. Quando a resolução for de 1024 ou superior, vai redimensionar o site para X; menor que 900, redimensiona para Y. Menor que 400, para Z.

Utilizamos *Media Queries*

- *Fluido*: Flui conforme o tamanho. Não usamos mais pixel para largura, nem para tamanho do texto ou imagem. O nosso site vai ocupar a largura total do monitor.



## Design Responsivo

Desktop

1920 x 1080

Tablet

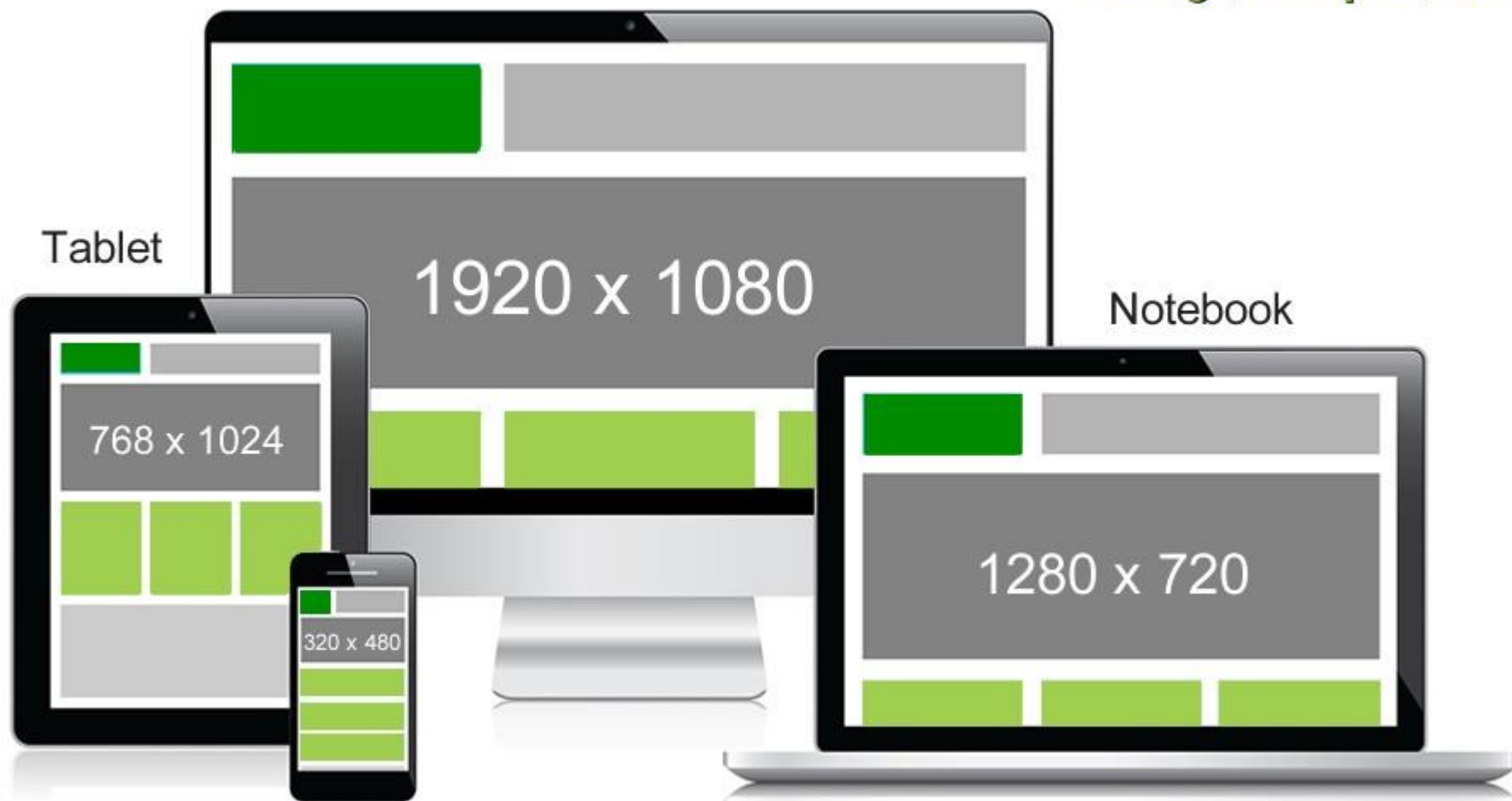
768 x 1024

Notebook

1280 x 720

320 x 480

Smart Phone







# Media Queries

- *Media queries* são consultas de mídia, ou seja, usamos para que o browser consulte o tamanho da tela, a resolução, para aí sim identificar no CSS3 as características que vai usar para o site. Usamos para sites responsivos.
- Ex. `<link rel="stylesheet" media="(max-width: 600px)" href="small.css" />`



# Tecnologia Web I

&lt;/&gt;

{ }

&lt;?&gt;





# Tecnologia Web I



320 = 4 colunas  
Mobile

768 = 12 colunas  
tablet

1200 = 20 colunas  
widescreen



480 = 7 colunas  
Mobile landscape

960 = 16 colunas  
desktop







# Tamanhos

- <https://www.mydevice.io/>





# Breakpoints

- Cada quebra no layout é conhecida como breakpoints. Ou seja, para cada resolução diferente que eu for usar no meu css, utilizando para isso um css único, vou criar breakpoints.





# Breakpoints

- Ex: Vamos fazer uma página bem básica de html, só com `<h1>` no body. Escrevam: testando responsivo.
- No CSS:
- */\*layout mobile\*/*
- *body { background-color: red; } h1 {color: white;}*





# Breakpoints

- `/*tablets*/`
- `@media (min-width:768px) {`
- `body { background-color: green; } h1 {color: orange;} }`
- `/*desktops */`
- `@media (min-width:960px) {`
- `body { background-color: pink; } h1 {color: purple;} }`
- `/*widescreen*/`
- `@media (min-width:1200px) {`
- `body { background-color: black; } h1 {color: grey; }`





# Unidades de medida relativas a viewport

- As unidades de medidas variam de acordo com o tamanho da viewport (tamanho da tela do usuário). Tudo pensando em proporções! No css:

- `1vw` - 1% da largura da viewport
- `1vh` - 1% da altura da viewport
- `1vmin` - 1% da menor dimensão da viewport
- `1vmax` - 1% da maior dimensão da viewport



# Unidades de medida relativas a viewport

- Exemplo: se eu quero que um elemento tenha a largura sendo a metade da largura do monitor, eu chamo:

```
#elemento {  
    width: 50vw;  
}
```







# Unidades de medida relativas a viewport

- Esses valores de medida são válidos para qualquer propriedade CSS que admita unidade de medida CSS, tais como as propriedades margin, padding, border e font-size entre outras





# Escala

- Como vamos trabalhar com porcentagens, os nossos elementos vão se redimensionar de acordo com a resolução. Porém, utilizando breakpoints, vamos querer que eles não se redimensionem proporcionalmente ao atingir esses pontos de quebra. Queremos que mantenha a escala de 1 pra 1. Para isso, no head do site colocamos:
  - `<meta name="viewport" content="width=device-width, initial-scale=1"/>`





# 9 princípios básicos do site responsivo

<https://blog.froont.com/9-basic-principles-of-responsive-web-design/>





# Convertendo as medidas

- Precisamos converter o que temos em pixel para porcentagem, assim ela não terá um valor fixo e irá se adaptar ao tamanho da tela! Mas como?





# Convertendo as medidas **texto**

- Para os texto transformamos os pixels em EM, uma unidade de medida própria. Em CSS (font-size) a EM está relacionada às fontes (tipografia) onde 1em equivale a 16px que é justamente o tamanho padrão, em média, das fontes nos navegadores.
- *1em = 16px; Vamos testar?*





# Convertendo as medidas **texto**

- REM - é calculado em cima do valor da TAG Body, Html ou Navegador. (R vem de root, raiz)
- EM - já utiliza no cálculo o valor do elemento pai, criando uma hierarquia no cálculo







# Convertendo tudo!

- Parece simples, mas é um pouco mais complexo. Precisamos ter em mente que a porcentagem para fazer a conversão será em cima do elemento pai, de onde o elemento a ser convertido está inserido.
- A fórmula na verdade é:

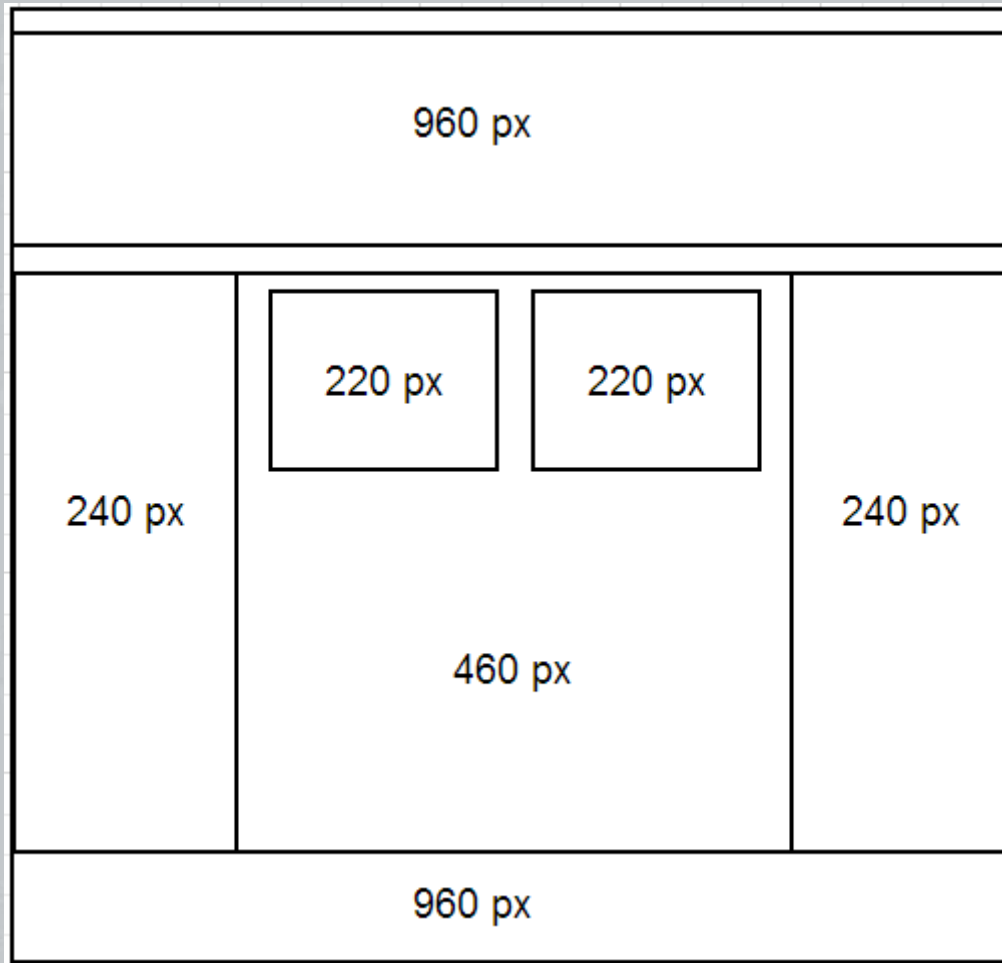
**Meta / Contexto = Resultado**

Resultado x 100, pq é porcentagem.





## Exemplo

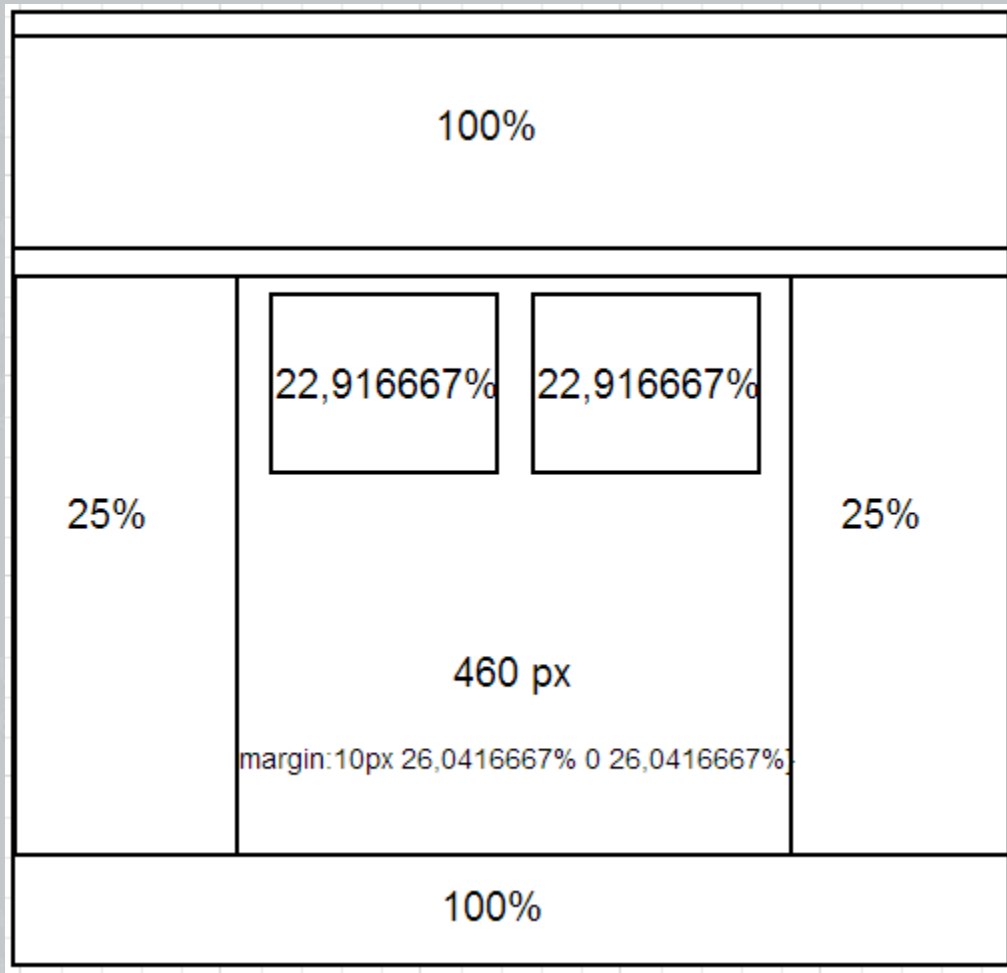


- O tamanho máximo seria de 960px. Logo,  $960\text{px} = 100\%$





# Exemplo



- O header e o footer ocupam tudo.

$$960/960=1;$$

- $240/960=0,25 = 25\%$

- $220/960=0,229166 \times 100$





# Pra gente não ter que fazer contas...

## - FRAMEWORKS!!!!!!!!!!!!!!





# Framework

- Os frameworks são sistemas para agilizar e facilitar a vida dos programadores na hora de fazer um site.
- *Porém, não adianta saber mexer neles e não saber html e css! Por isso trabalhamos duro no inicio e agora vamos facilitar.*





# Framework

- [Link para uma listinha com alguns bons frameworks para desenvolvimento front-end](#)







# Bootstrap

- Como criar um site responsivo sem enlouquecer com os cálculos e regras dos media queries e das porcentagens??
- Alguns sites feitos com bootstrap:

<https://www.awwwards.com/websites/bootstrap/>





# Bootstrap

<http://getbootstrap.com.br/>



Bootstrap é o mais popular framework HTML, CSS, e JS para desenvolvimento de projetos responsivo e focado para dispositivos móveis na web.

Baixar Bootstrap

Atualmente v3.3.7

