**SENAI-SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL**

**PETERSON BRAYAN DE ANDRADE**

**VINICIUS DOS ANJOS FERREIRA**

**YOLELI ALICE DE OLIVEIRA MUNJENJI**

**BOX MODEL E MEDIA QUERIES**

**TRABALHO DE PROGRAMAÇÃO DE APLICATIVOS**

**CURITIBA**

**2017**

**PETERSON BRAYAN DE ANDRADE**

**VINICIUS DOS ANJOS FERREIRA**

**YOLELI ALICE DE OLIVEIRA MUNJENJI**

**BOX MODEL E MEDIA QUERIES**

Trabalho apresentado á unidade curricular do 1° Módulo de Técnico em informática, Senai Boqueirão com requisito para obtenção do título de programação de Aplicativos.

Orientador: Prof(a): Cezar Jenzura

**CURITIBA**

**2017**

**RESUMO**

Andrade, Anjos ferreira e oliveira Munjenji, Peterson, Vinicius Brayan eYoleli Alice. Programação de aplicativos.Técnico em informática .Senai.Curitiba,2016

Trabalho apresenta os modelos de caixas que determina as características dimensionais , visuais e de posicionamento que regem o layout dos elementos de marcação e suas propriedades e apresenta também o media queries que é vários layout para cada dispositivos , um celular não tem o mesmo modelo que um desktop.

**Palavra-Chaves:** CSS.Box model. Media querie.Propriedades.layout

LISTA DE FIGURAS

FIGURA 1 BOXMODEL....................................................................................12

FIGURA 2- BLOCO DE CONTUDO..................................................................12

FIGURA 3- BOX ANÔNIMOS............................................................................12

FIGURA 4- BOX INLINE ANÔNIMO..................................................................12

FIGURA 5- ÁREAS NO BOX MODEL CSS.......................................................13

FIGURA 6- YOUTUBE PARA MOBILE.............................................................28

FIGURA 7- TABLELESS PARA MOBILE..........................................................28

FIGURA 8- YOUTUBE PARA DESKTOP..........................................................29

FIGURA9-TABLELESS PARA DESKTOP.........................................................29

LISTA DE TABELAS

TABELA 1 TIPO DE BORDER-STYLE.............................................................16

TABELA 2- DISPLAY.........................................................................................19

FIGURA 3- TIPOS DE MÍDIAS..........................................................................20

FIGURA 4-TIPOS DE OPERADORES.............................................................20

FIGURA 5-SUPORTE PARA MEDIA QUERIES...............................................27

SUMÁRIO

1.0 INTRODUÇÃO...............................................................................................8

1.1 OBJETIVO.....................................................................................................9

1.2 JUSTIFICATIVA...........................................................................................10

2 MÓDULO CSS BASIC BOX MODEL.............................................................11

2.1 BLOCO DE CONTEÚDO OU CONTAINER................................................12

2.2 BOX NÍVEL DE BLOCO ANÔNIMO............................................................12

2.3 BOX INLINE ANÔNIMO...............................................................................12

2.4 DIAGRAMA DO BOX MODEL.....................................................................13

2.5 PROPRIEDADE CSS DO BOX MODEL......................................................14

2.5.1MARGIN................................................................................................14,15

2.5.2 PADDING..................................................................................................15

2.5.3 BORDER...................................................................................................15

2.5.3.1 BORDER-WIDTH...................................................................................16

2.5.3.2 BORDER-STYLE..............................................................................16,17

2.5.3.3 BORDER-COLOR..................................................................................17

2.5.4 WIDTH......................................................................................................17

2.5.4.1 MAX-WIDTH..........................................................................................18

2.5.4.1 MIN-WIDTH...........................................................................................18

2.5.5 HEIGHT....................................................................................................18

2.5.5.1 MAX-HEIGHT...................................................................................18,19

2.5.5.2 MIN-HEIGHT.........................................................................................19

2.5.6 DISPLAY..................................................................................................19

3.MEDIA QUERIES......................................................................................19,20

3.1 ANATOMIA DA MEDIA QUERY..................................................................21

3.2 FUNCIONALIDADE. DE MÍDIA...................................................................21

3.2.1 WIDTH.................................................................................................21,22

3.2.2 DEVICE-WIDTH........................................................................................22

3.2.3 HEIGTH...............................................................................................22,23

3.2.4 DEVICE-HEIGHT......................................................................................23

3.2.5 ORIENTATION....................................................................................23,24

3.2.6 ASPECT-RATIO......................................................................................24

3.2.7 DEVICE-ASPECT-RATIO........................................................................24

3.2.8 COLOR.....................................................................................................25

3.2.9 COLOR-INDEX.........................................................................................25

3.2.10 MONOCHROME.....................................................................................25

3.2.11 RESOLUTION.........................................................................................26

3.2.12 SCAN......................................................................................................26

3.2.13 GRID.......................................................................................................26

3.3 BREAKPOINT..............................................................................................27

3.4 SUPORTE....................................................................................................27

4.CONCLUSÃO.................................................................................................31

REFERÊNCIA....................................................................................................32

1. **INTRODUÇÃO**

O trabalho a seguir apresenta o que é box model (modelos de caixas), seus tipos básicos de box,as suas propriedades de cada elemento do box model e também apresentado o conteúdo do media querie(consulta ao tipo de mídia ) que possui tipos de Midas, operadores , funcionalidades de mídia e o suporte.

1. **OBJETIVO**

Com esse trabalho é possível aprender cada vez mais sobre o CSS. Mostra como colocar bloco anônimo ,bloco inline e a utilização em diversos dispositivos .

1. **JUSTIFICATIVA**

Trabalho mostra como funciona, as propriedades, suas funções e suportes.

O uso do media queries é muito útil para os sites , por que cada dispositivo depende de um layout diferente.

O trabalho foi fundamentado no livro css3 do Maurício samy e em sites sobre box model e media queries

1. **Módulo CSS Basic Box Model**

No CSS, o termo “Modelo de caixa” ou “Box Model” é usado quando se fala de design e layout (rascunho, espaço ou projeto).

O modelo de caixa CSS é essencialmente uma caixa que envolve todo elemento HTML.

CSS admite que o layout de um documento seja criado a partir de uma árvore de elementos de marcação.

Cada elemento-filho e opcionalmente, uma string de textos antes da lista, uma string de texto depois da lista e uma string de textos entre os itens da lista.

O único elemento que não possui um elemento-pai é o elemento raiz do documento.

Para formatar visualmente o layout de um documento as CSS descrevem como todo elemento e cada string de textos serão dimensionados e posicionados, transformado a árvore do documento, em uma árvore de boxes, cujas dimensões, cujo posicionamento e relacionamento com boxes vizinhos dependem das propriedades de cada um dos boxes.

Há três tipos básicos de boxes:

* **Block-level boxes (boxes nível de bloco):**

São os boxes por um parágrafo.

* **Line-Boxes (boxes em linhas):**

São boxes em linha de textos

* **Inline-level boxes (boxes nível inline):**

São boxes gerados pelas palavras em uma linha de texto.

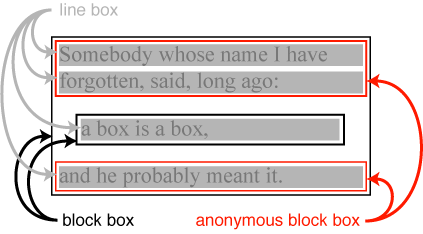


Figura 1 –Box Model.

Fonte: <http://www.w3.org/TR/css3-box/>

1. **Bloco de conteúdo ou container**

É um caixa retangular que possui um ou mais caixas, que na árvore do documento são seus elementos descendentes.

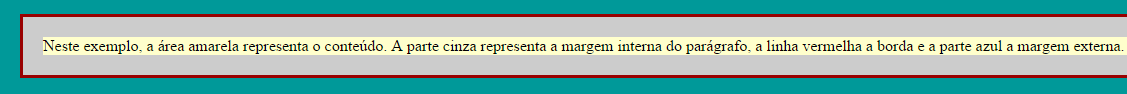


Figura 2 –Bloco de conteúdo.

Fonte:

http://guilhermemuller.com.br/en/elearning/exemplos/html\_basico/2/margens-9

1. **Box nível de bloco anônimo**

A renderização das três linhas de textos se deu como se cada uma delas fosse um Box independente. Isso porque, segundo o Box Model CSS quando um container possui um elemento nível de bloco, todos os conteúdos inseridos diretamente no container se comportam como elemento nível de bloco e são chamados de boxes anônimos.

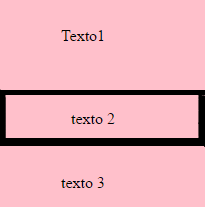


Figura 3 –Box anônimo

1. **Box inline anônimo**

É a renderizaçao de três linhas em uma linha única.

inline2.png

Figura 4 – Box inline anônimo

Fonte: http://guilhermemuller.com.br/en/elearning/exemplos/html\_basico/2/display-2

1. **Diagrama do Box Model**

Cada um dos elementos da marcação HTML possui propriedades CSS cujos valores podem ou não estar definidas por padrão pelo mecanismo nativos do navegador e/ou serem definidos em regras CSS explicitamente declaradas pelo desenvolvedor.Exemplo:

* **HTML**

<body>

<h1>Box Model</h1>

</body>

Todos os elementos de marcação possuem o chamado valor inicial para as propriedades CSS.

Aparentemente, essa propriedade tem os mesmo valor inicial para todos os navegadores mostrados, mais isso não acontece para todas as propriedades CSS e para todos os elementos de marcação. Os valores iniciais podem variar de acordo com o navegador de layout. Quando verificar seus projetos CSS em diferente navegadores, não se esqueça de levar esse comportamento em consideração.

As propriedades que determinam as dimensões dos boxes são margin, bordex, padding, widht e heidht, e as palavras para referência aos quatros lados dos boxes são top, right, bottom, left.

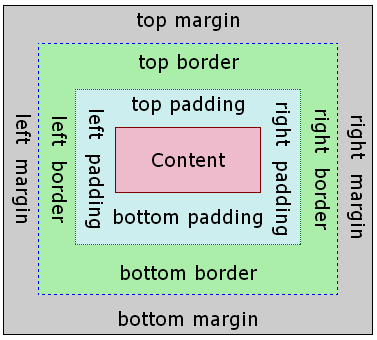


Figura 5- Áreas no Box Model CSS.

Fonte: <https://www.codeproject.com/Articles/567385/CSSplusBoxplusModelplusandplusPositioning>

1. **Propriedade CSS do Box Model**

Determinam a criação do box de um elemento de marcação.

1. **Margin**

Essa propriedade define a espessura das margens de um box. Podemos definir a espessura com uso de declaração individual ou com uso de uma declaração abreviada.

* **CSS**

h1 { h1 {

Margin-top: 45px; margin: 45px 40px 10px 50px;

Margin-right: 40px; Ou }

Margin-bottom; 10px

Margin-left: 50px;

}

Declarações de valores CSS na sua forma abreviada podem ser feitas com uma lista de um, dois, três ou quatro valores separados por espaço em branco, e seus significados são conforme mostrados a seguir.

* **Um Valor:** Valor único é valido para os lados top e bottom, e o segundo top right bottom e left;
* **Dois valores:** o primeiro é valido para o lado top, e o segundo, para right e left
* **Três Valores:** O primeiro é valido para o lado top, o segundo, para os lados right e left, e o terceiro para o lado bottom;
* **Quatro Valores:** Os valores são validos, respectivamente, para os lados top right, bottom e left do box.

1. **Padding**

Essa propriedade define a espessura do enchimento de um box. Chamamos de enchimento a espessura da área que separa a borda do conteúdo.

Podemos definir o enchimento com o uso de declaração individual ou com uso de uma declaração abreviada.

**CSS**

h1{ h1{

padding-top: 15px;

Padding-right: 30px; Ou padding: 15px 30px 10px 40px

Padding-bottom: 10px;

Padding-left: 40px; }

}

1. **Border**

Essa propriedade define a espessura, o estilo e a cor da borda de um box. Podemos definir cada uma dessas três características da borda individualmente e para cada lado do box, com uso de declaração individuais, ou suar uma declaração abreviada

1. **Border-width**

A sintaxe da regra CSS para declarar a largura da borda individualmente.

Exemplo:

**CSS**

H1 {

Border-top-widht: 5px;

Border-right-widht: 10 px;

Border-bottom-widht: 8px;

Border-left-widht: 20px;

}

**Forma abreviada :**

H1{

Border-widht: 5px 10px 8px 20px ;

}

1. **Border- style**

Sintaxe da regra CSS para declarar o estilo de borda individualmente.

Exemplo:

**CSS**

H1{

Border-top-style:solid;

Border-right-style; dished;

Border-bottom-style: dotted;

Border-left-style:ridge

}

**Forma abreviada :**

H1{

Border-style: solid dashed dotted ridge;

}

**Tabela 1- Tipo de border-style**

|  |  |
| --- | --- |
| **Estilo** | **Descrição** |
| **None** | Ausência de borda. |
| **Hidden** | O mesmo que none. |
| **Dotted** | Borda pontilhada. |
| **Dashed** | Borda tracejada. |
| **Solid** | Borda sólida contínua |
| **Double** | Borda em linha dubla. |
| **Groove** | Borda entalhada. |
| **ridge** | Borda em ressalto. |
| **inset** | Borda em baixo -relevo |
| **outset** | Borda em alto - relevo |

**Fonte: Livro-Css3:Desenvolva aplicação web profissionais com uso dos poderosos recursos de estilização das css3.**

1. **Border-Color**

A sintaxe da regra CSS para declarar a cor da borda individualmente.Exemplo:

**CSS**

H1{

Border-top-color: #f00; /\*vermelho\*/

Border-right-color: #0f0; /\*verde\*/

Border-bottom-color: #00f /\*azul\*/

Border-left-color: #000; /\*preto\*/

}

**Forma abreviada:**

H1{

Border –color: #f00 #0f0 #00f #000;

}

1. **Width**

Essa propriedade define a largura da área de conteúdo de box, isto é, não inclui as espessuras de margem, border e padding.Exemplo:

**CSS**

Div{

Widht: 500px;

}

1. **Max-width**

Essa propriedade define a largura máxima de um box.Exemplo:

**CSS**

Div {

Max-widht: 960px;

}

1. **Min-width**

Essa propriedade define a largura mínima de um box.Exemplo:

**CSS**

Div {

Min-width: 600px;

}

1. **Height**

Essa propriedade define a altura da área de um box. Exemplo:

**CSS**

Div {

Height: 300px;

}

1. **Max-height**

Essa propriedade define a altura máxima de um box.Exemplo:

**CSS**

Div {

Max-Height: 768px;

}

1. **Min-height**

Essa propriedade define a altura mínima de um box.Exemplo:

**CSS**

Div {

Min-Height: 450px;

}

1. **Display**

Essa propriedade define o tipo de box a ser gerado para um elemento. Os valores possíveis para essa propriedades.

**Tabela 2- Display**

|  |  |
| --- | --- |
| **Inline** | Define boxes em linhas. |
| **Block** | Define boxes nível de bloco. |
| **Inline-block** | Define boxes nível de bloco disposto no fluxo como boxes inline. |
| **List-item** | Define boxes tipo item de lista |
| **Run-in** | Define boxes de acordo com o contexto. Se o box que se segue for nível de bloco run-in, define um box inline, caso contrário, define um box nível de bloco |
| **Compact** | Define box de acordo com o contexto.Se o box que se segue for nível de bloco compact,define um box nível de bloco disposto no fluxo como box inline,caso contrário,define um box nível de bloco. |
| **None** | Define que o box não será criado para o elemento. |

**Fonte: Livro-Css3:Desenvolva aplicação web profissionais com uso dos poderosos recursos de estilização das css3.**

1. **Media Queries**

HTML foi criado para ser portável, deve ser lido e interpretado por qualquer tipo de dispositivo. A forma do layout do desktop tem que ser diferente do layout do dispositivo móvel .

1. **Anatomia da media query**

Media Queries (consulta ao tipo de mídia), o objetivo é servir a folha de estilo especifica para determinado tipo de mídia mediante uma consulta ao tipo de mídia ao qual a folha de estilo será servida..

Uma Media Queries é uma expressão lógica que retorna os valores true ou false.

Se o tipo de mídia definido em uma Media Query casa com o tipo de mídia do dispositivo no qual o agente do usuário está rodando e se todas as expressões definidas na media query são verdadeiras.

Exemplo:

**Css**

Screen and (color)

Screen: Tipo de mídia

And: Operador

( ):Expressão

Color: Funcionalidade da mídia

**Tabela 3-Tipos de mídias**

|  |  |
| --- | --- |
| **Estilo** | **Aplicação** |
| **All** | É aplicado em todo os dispositivos |
| **Braille** | Para todos os dispositivos táteis |
| **Embossed** | Para dispositivos que imprimem em Braille |
| **Handheld** | Para dispositivos de mão |
| **Pint** | Para impressão em papel |
| **Projection** | Para apresentações. |
| **Screen** | Para dispositivos para telas coloridas e de alta resolução |
| **Speech** | Para sintetizadores de voz. |
| **tty** | Para dispositivos que utilizam uma grade fixa para exibição de caracteres |
| **tv** | para dispositivos como baixa resolução |

**Fonte: Livro-Css3:Desenvolva aplicação web profissionais com uso dos poderosos recursos de estilização das css3.**

**Tabela 4- Tipos de Operador**

|  |  |
| --- | --- |
| **Estilo** | **Aplicação** |
| **And** | Operador de concatenação,retorna true se as condições á esquerda e a direita do operador são ambas verdadeiras. |
| **Only** | Palavras-chaves usada para esconder folhas de estilo dos navegadores antigos que não suportam media query. |
| **Not** | Operador de negação, retorna true se a funcionalidade não existir. |

**Fonte: Livro-Css3:Desenvolva aplicação web profissionais com uso dos poderosos recursos de estilização das css3.**

1. **Funcionalidad de mídia**

Funcionalidades de mídia se assemelham às propriedades CSS. Ou seja, possuem um nome e aceitam determinados valores.

1. **Width**

Essa funcionalidade descreve a largura da área de saída do dispositivo do usuário. Para a mídia continua, é a largura da viewport, incluindo barra de rolagem, se existente, para mídia paginada, é a largura do box da pagina. O valor possível para essas funcionalidade é uma medida de comprimento CSS não negativa. Essa funcionalidade aceita os prefixos min- e max-.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(width:25cm)” href=’’[http://’...](http://'...)/>

@media screen and(min-width: 400px) and (Max-width: 700px){...}

@media handheld and(min-width: 20em) and (Max-width: 20em) {...}

1. **Device-Width**

Essa funcionalidade descreve a largura da área de renderização do dispositivo do usuário. Para mídia contínua, é a largura da tela e para mídia paginada e a largura da pagina. O valor possível para essas funcionalidade é uma medida de comprimento CSS não negativa. Essa funcionalidade aceita os prefixos min- e max-.Exemplo:

<link rel=”stylesheet” media”print and(Max-device-width: 23cm)” href=’’[http://’...](http://'...)/>

@media screen and(Min-device-width: 400px) and (Max-device-width: 700px){...}

@media handheld and(Max-device-width: 20em) and (device-width: 20em) {...}

1. **Height**

Essa funcionalidade descreve a altura da área de saída do dispositivo do usuário. Para mídia contínua, é a altura da viewport, incluído barera de rolagem, se existente, e para mídia paginada, e a altura box da pagina. O valor possível para essas funcionalidade é uma medida de comprimento CSS não negativa. Essa funcionalidade aceita os prefixos min- e max-.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(height :29cm)” href=’’[http://’...](http://'...)/>

@media screen and(min-height: 600px) and (Max-height: 768px){...}

@media handheld and(min-width: 14em) and (Max-width: 14em) {...}

1. **Device-Height**

Essa funcionalidade descreve a altura da área de rederização do dispositivo do usuário. Para mídia continua é a altura da tela para midoa pagiada é a altura da pagina. O valor possível para essas funcionalidade é uma medida de comprimento CSS não negativa. Essa funcionalidade aceita os prefixos min- e max-.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(Max-device-height: 29cm)” href=’’[http://’...](http://'...)/>

@media screen and(Min-device-height :600px) and (Max-device-height: 768px){...}

@media handheld and(Max-device-height: 14em) and (device-height: 14em) {...}

1. **Orientation**

Essa funcionalidade se baseia em um mecanismo de comparação entre largura(funcionalidade width) e a funcionalidade altura(funcionalidade heigth) da área de dispositivo do usuário. Os valores possíveis para essa funcionalidade são landscape e portrait. O valor landscape casa com mídia quando a funcionalidade width é maior que a funcionalidade height; caso contrario o valor é portrat. Essa funcionalidade não aceita os prefixos min- e max-.

Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(orientation: portrait)” href=’’[http://’...](http://'...)/>

@media screen and(orientation: landscape){...}

@media handheld and(orientation:portrait) {...}

1. **Aspect-ratio**

Essa funcionalidade se baseia na razão entre a largura (funcionalidade width) e a altura(funcionalidade height) da área de renderização do dispositivo do usuário. Os valores possíveis para essa funcionalidade são frações do tipo a/b. Essa funcionalidade aceita prefixos min- e Max.

Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(aspect-ratio: 4/3)” href=’’[http://’...](http://'...)/>

@media screen and(aspect-ratio: 16/9){...}

@media handheld and(aspect-ratio: 5/2) {...}

1. **Device-aspect-ratio**

Essa funcionalidade se baseia na razão entre a largura (funcionalidade device-width) e a altura(funcionalidade device-height) da área rederização do dispositivo do usuário. Os valores possíveis para essa funcionalidade são frações do tipo a/b. Essa funcionalidade aceita prefixos min- e max-.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(device-aspect-ratio: 4/3)” href=’’[http://’...](http://'...)/>

@media screen and(device-aspect-ratio: 16/9){...}

@media handheld and(min-device-aspect-ratio: 5/2) {...}

1. **Color**

Essa funcionalidade descreve o número de bits por componente de cor no dispositivo do usuário.Essa funcionalidade aceita os prefixos min- e max, e seu valor deve ser um número inteiro não negativo.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(color)” href=’’[http://’...](http://'...)/>

@media screen and(color: 2){...}

@media screen and(Max-color: 4){...}

@media screen and(color: 0){...}

1. **Color-index**

Essa funcionalidade descreve o número de entradas na paleta de cores disponíveis no dispositivo do usuário. Essa funcionalidade aceita os prefixos min- e max-, e seu valor deve ser um número inteiro não negativo.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(color-index)” href=’’[http://’...](http://'...)/>

@media screen and(color-index: 0){...}

@media screen and(Max-color-index: 4096){...}

1. **Monochrome**

Essa funcionalidade descreve o número de bits por pixel no dispositivo monocromático do usuário. Essa funcionalidade aceita os prefixos min- e max-, e seu valor deve ser um número inteiro não negativo.Exemplo:

**CSS**

<link rel=”stylesheet” media”print and(monochrome)” href=’’[http://’...](http://'...)/>

@media screen and(color: 2){...}

1. **Resolution**

Essa funcionalidade descreve a resolução, ou seja, a densidade de pixels do dispositivo do usuário. Essa funcionalidade aceita os prefixos min- e max-, e seu valor deve ser um número inteiro não negativo representando a densidade de pixels por unidade de medida.Exemplo:

**CSS**

@media screen and(min-resolution: 300dpi){...}

@media screen and(resolution: 118dcm){...}

1. **Scan**

Essa funcionalidade descreve o processo de escaneamento de imagens em mídia do tipo TV. Essa funcionalidade não aceita os prefixos min- e max-, e seu valor deve ser a palavra-chave progressive ou interlace.Exemplo:

**CSS**

@media tv and(scan: progressive){...}

@media Tv and(scan: interlace){...}

1. **Grid**

Essa funcionalidade descreve se a saída do dispositivo é em grid ou bitmap. Essa funcionalidade não aceita os prefixos min- e max, e seu valor deve ser 0 ou 1. O valor 1 ou a omissão de valor define um dispositivo com saída em grid, por exemplo: um terminal tty ou um dispositivo móvel com um só tamanho fixo de fonte.O valor 0 define saída bitmap.Exemplo:

**CSS**

@media handheld and(Max-width15em){...}

@media handheld and(device-max-height: 7em){...}

1. **Breakpoint**

Breakpoint são os pontos em que há quebra do layout em consequência da mudança de resolução e essa definição .

A maneira moderna de se determinar o breakpoint é experimentalmente. Construa seu layout, visualize-o e lentamente vá redimensionando a janela do navegador, observamdo em que os pontos o layout quebra e precisa ser adaptada.

1. **Suporte**

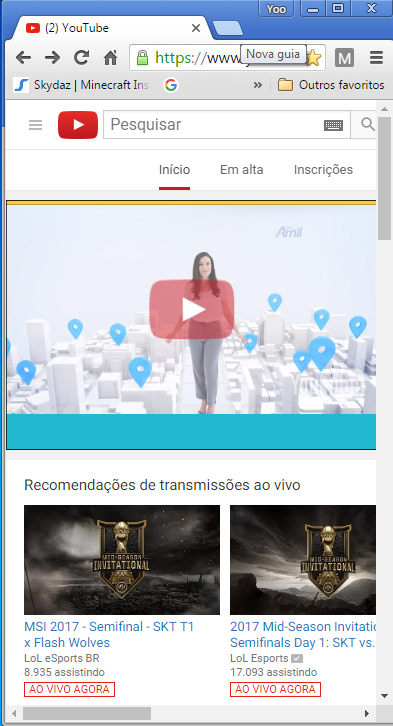
## Compatibilidade no navegador

* Desktop
* Mobile

**Tabela 5- Suporte para media queries**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Característica | Chrome | Firefox (Gecko) | | | Internet Explorer | | | Opera | Safari | |
| Suporte básico | 21 | [3.5](https://developer.mozilla.org/en-US/Firefox/Releases/3.5) (1.9.1) | | | 9.0 | | | 9 | 4 | |
| Grade | ? | Não suportado (grid media type is not supported) | | | ? | | | ? | ? | |
| Resolução | 29 | [3.5](https://developer.mozilla.org/en-US/Firefox/Releases/3.5) (1.9.1) supports [<integer>](https://developer.mozilla.org/pt-BR/docs/Web/CSS/integer) values; [8.0](https://developer.mozilla.org/en-US/Firefox/Releases/8) (8.0) supports [<number>](https://developer.mozilla.org/pt-BR/docs/Web/CSS/number) values, as per the spec. | | | ? | | | ? | ? | |
| Scan | ? | Não suportado (tv media type is not supported) | | | ? | | | ? | ? | |
| Característica | Android | | Firefox Mobile (Gecko) | IE Mobile | | Opera Mobile | Safari Mobile | | |
| Suporte básico | (Yes) | | (Yes) | ? | | (Yes) | (Yes) | | |

**Fonte: https://developer.mozilla.org/pt-BR/docs/Web/Guide/CSS/CSS\_Media\_queries**



**Figura 6-Youtube para mobile.**

**Fonte: https://www.youtube.com/**



**Figura 7-tableless para mobile**

**Fonte: https://tableless.com.br/**

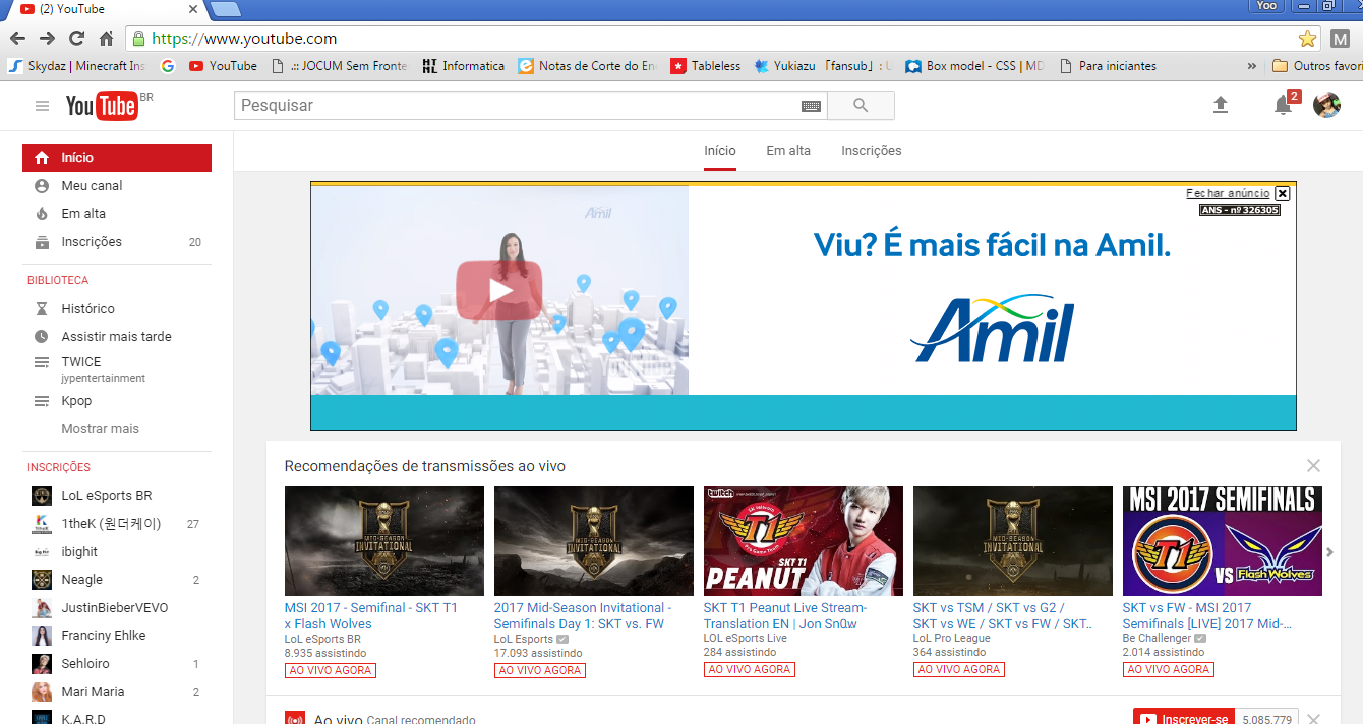


Figura 8- Youtube para desktop

Fonte: https://www.youtube.com/



**Figura 9- Tableless para desktop**

**Fonte: https://tableless.com.br/**

1. **Conclusão**

CSS é uma folha de estilo que apresenta diversas criações como mostra o box model.Pode conter box dentro de box , box anônimos e ate mesmo box em linhas em diversos estilos e tamanho.

CSS vem mudando cada vez mais como, por exemplo, o media queries que possui um layout para cada tipo de dispositivo. Se um aparelho móvel não possui o media queries ele mostraria o mesmo estilo de uma pagina do desktop. É ruim, pois para acessar o que deseja precisaria utilizar o zoom

**Referencias**

<https://www.w3schools.com/css/css_boxmodel.asp>

<https://developer.mozilla.org/pt-BR/docs/Web/Guide/CSS/CSS_Media_queries>

<https://developer.mozilla.org/pt-BR/docs/Web/CSS/box_model>

<http://tableless.github.io/iniciantes/manual/css/box-m>

<http://www.w3.org/TR/css3-box/>

<http://guilhermemuller.com.br/en/elearning/html_css_basico/licao/2/elementos-bloco-em-linha>

<https://www.codeproject.com/Articles/567385/CSSplusBoxplusModelplusandplusPositioning>

Silva, Maurício samy.Css3: desenvolva aplicações web profissionais com uso dos poderosos recursoos de estilização das css3- Box Model.São Paulo:Novatec Editora Ltda ,2012

<https://tableless.com.br/introducao-sobre-media-queries/>

<https://www.w3schools.com/css/css_rwd_mediaqueries.asp>

<http://pt-br.learnlayout.com/media-queries.html>

Silva, Maurício samy.Css3: desenvolva aplicações web profissionais com uso dos poderosos recursoos de estilização das css3- Media Queries .São Paulo:Novatec Editora Ltda ,2012