

CSS – Parte 3

Prof. Tiago Lopes Telecken

IFRS – Rio Grande



Posicionando as caixas – Atributo Display

- Comportamento padrão (default):
 - **display:block**, comportamento de bloco;
 - **As caixas são empilhadas**. A primeira caixa é colocada no canto superior esquerdo e as demais uma abaixo da outra conforme a ordem em que estão no html
 - As caixas terão a **largura máxima** (ocupando toda caixa pai) e **altura mínima** (altura de seu conteúdo)
 - Ao colocar width e height a caixa passa a ter altura e largura definidos nestes atributos. Mesmo tendo espaço ao lado, as caixas são empilhadas

Posicionando as caixas – Atributo Display

- Comportamento lado a lado (de palavra):
 - **display:inline;**
 - **As caixas são colocadas lado a lado.** A primeira caixa é colocada na esquerda e as demais uma ao lado da outra conforme a ordem em que estão no html. Quando não houver espaço ela é colocada na linha de baixo
 - As caixas terão sempre a **largura e altura mínimas** (do tamanho de seu conteúdo, width e height são ignorados).
 - Texto e algumas tags como b e i são **inline** por padrão. A grande maioria é block por padrão

Posicionando as caixas – Atributo Display

- Comportamento bloco lado a lado:
 - **display:inline-block;**
 - As caixas são colocadas lado a lado. A primeira caixa é colocada na esquerda e as demais uma ao lado da outra conforme a ordem em que estão no html. Quando não houver espaço ela é colocada na linha de baixo
 - As caixas terão sempre a largura e altura mínimas (do tamanho de seu conteúdo, **width e heigth não são ignorados**).
- Outras opções de display serão vistas adiante no curso (flex, grid, inline-flex, ...)

Caixas dentro de caixas

- As dimensões (**margin, padding, border**) das caixas podem ser alteradas para mudar o seu posicionamento
- As tags são colocadas umas dentro das outras no código html. As caixas correspondentes a estas tags também ficam visualmente umas dentro das outras.
- **Ao mover a caixa pai as filhas movem-se junto.**
- As dimensões da caixa pai são as dimensões máximas das filhas (se exceder ocorre um overflow)
- Deve-se ter o cuidado de colocar as dimensões das caixas pai maiores que a soma das filhas (evitar um overflow)

Posicionamento - Float

- Com o atributo **float** uma caixa deixa de ser empilhada e passa a flutuar. Ao flutuar os conteúdos das caixas irmãs ficam em sua volta.
 - Com o valor **left** as caixas flutuam uma ao lado da outra a partir da esquerda e somente quando a caixa não couber ela é colocada abaixo
 - Com o valor **right** elas são flutuadas uma ao lado da outra a partir da direita
 - Se todos irmãos forem float eles flutuarão como um inline a esquerda e/ou direita

Position - Fixed, Stricky

```
header {  
  position: fixed; //striky  
  top: 5px;  
  left: 5px;  
  width: 1000px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}
```

O elemento passa a ter uma posição fixa em relação ao topo do navegador.

Fixed: fica sempre em um novo plano. Sem ocupar espaço no plano principal

Stricky: fica em um novo plano no caso de um scroll. Seu espaço no plano principal continua ocupado quando ele vai para um novo plano

Posicionamento - Relativo

- O posicionamento relativo precisa do comando “position: relative” combinado com pelo menos um dos seguintes comandos left, top, bottom, right. O comando position:relative faz com que a caixa em que é aplicado se desloque da sua posição normal. Ao se deslocar a caixa fica fora do empilhamento padrão porém **sua antiga posição não fica livre** para que outras caixas ocupem este lugar

A seguir um exemplo:

```
p.principal { position: relative;
left:70px;
top: 100px }
```

Vai deslocar a caixa de p.principal 70px a esquerda e 100px para baixo

Posicionamento - Absoluto

- O posicionamento absoluto precisa do comando “position: absolute” combinado com pelo menos um dos seguintes comandos left, top. O comando position:absoluto faz com que a caixa em que é aplicado se desloque da sua posição normal. Ao se deslocar a caixa fica fora do empilhamento padrão e **sua antiga posição fica livre para que outras caixas ocupem este lugar**

A seguir um exemplo:

```
p.principal { position: absolute;
left:70px;
top: 100px }
```

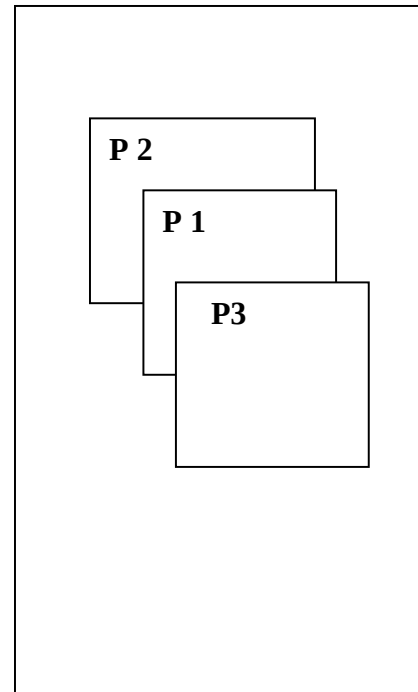
Vai deslocar a caixa de p.principal 70px a esquerda e 100px para baixo

Posicionamento - Index-z

- Quando o posicionamento fixed, absoluto, ou relativo é utilizado, pode acontecer de duas caixas ficarem sobrepostas (uma por cima da outra).
- Nesta situação só uma caixa vai aparecer completamente para o usuário, já as demais ficariam escondidas em baixo da caixa que está em primeiro plano.
- Para controlar qual caixa é mostrada em primeiro plano utiliza-se o comando index-z.
- O valor da propriedade index-z é um numero qualquer. As caixas que tiverem o maior valor do index-z aparecem em primeiro plano. Ou seja a caixa que tem o número z-index maior aparece na frente da caixa que tem o z-index menor. A seguir um exemplo.

Index-z

```
P.1 { position: absolute;  
      left:70px;  
      top: 100px  
      z-index:2}  
P.2 { position: absolute;  
      left:80px;  
      top: 110px  
      z-index:1}  
P.3 { position: absolute;  
      left:90px;  
      top: 120px  
      z-index:3}
```



Posicionamento

- Muito cuidado ao se utilizar os posicionamentos relativo e absoluto com os métodos de posicionamento através de margins e paddings visto anteriormente
- Ao se utilizar os posicionamentos relativo e absoluto suas respectivas caixas saem do fluxo normal de empilhamento e seu comportamento fica mais difícil de ser controlado (como ele ficará em telas de diferentes tamanhos?)
- Posicionamento CSS avançado
 - Flexbox
 - Grid

Posicionamento – Estruturando uma página

- Os arquivos site.html e site.css utilizam os recursos apresentados até aqui para montar uma estrutura mais elaborada que poderia ser a estrutura de uma página web. A seguir comentaremos alguns comandos css utilizados
- O seletor ***** é utilizado para zerar as medidas de todas tags. Os navegadores tem medidas padrão de padding, border e margin diferentes (as vezes é 0px, outras é 1px), por isso, é uma boa prática zerar estas medidas e depois ir alterando quando necessário (**reset** e **normalize** seriam soluções mais completas ou corretas para zerar ou normalizar todo o site)
- Os comandos abaixo, usados em conjunto, centralizam uma caixa com display de bloco
 - margin-left: auto;
 - margin-right: auto;



Posicionamento

- Os demais comandos foram vistos anteriormente.
 - As caixas que estão lado a lado tem a propriedade `display:inline-block`
 - O tamanho do conteúdo foi definido por `width` e `height`
 - Ajustes no posicionamento foram definidos principalmente pelos tamanhos do `padding` e `margin`
 - A organização hierárquica (que caixas vão dentro de que caixas) foi definida no `html`
 - O principal cuidado foi não deixar que aconteça o `overflow`. As dimensões das caixas internas não extrapolam o tamanho das caixas externas

Posicionamento

- A página `site.htm` mostra um estilo de estruturação que usa mais a `vw` (responsiva). Se trocar o `css` da página para o `site2.css` pode-se ver um estilo que usa mais a medida `px` (pouco responsiva).



CSS

Prof. Tiago Lopes Telecken

IFRS - Rio Grande

