Tecnologia Web I

Gabriela Silveira

PROPRIEDADE FLOAT

Vamos alinhar caixinhas (divs) horizontalmente?

Criamos divisões que estão umas em cima das outras. Vamos dar tamanhos para elas e coloca-las lado a lado utilizando a propriedade float. Mas vamos testar primeiro e depois aplicar no nosso site.

Float vem de flutuação.

PROPRIEDADE FLOAT

Nova pasta. Dentro um html e um css. Bloco de notas

Novo arquivo html:

Doctype, <html lang=“pt-br”>, <head>, <metachartset=“UTF-8” />, <title>, <link rel="stylesheet" href=“style.css” />

<body>... Basicão!

PROPRIEDADE FLOAT

```
<body>
        <div>primeira div</div>
        <div>segunda div</div>
        <div>terceira div</div>
        <div>quarta div</div>
        <div>quinta div</div>
</div>
```

PROPRIEDADE FLOAT

Bloco de notas, arquivo style.css:

```
div {
        background-color: black;
        width: 100px;
        height: 100px;
        margin: 5px;
}
```

PROPRIEDADE FLOAT

Bloco de notas, arquivo style.css:

```
div {
        background-color: black;
        width: 100px;
        height: 100px;
        margin: 5px;
}
```

VOLTANDO AO CSS

Delimitando o tamanho do site e suas divisões.

Resoluções que podemos trabalhar – o layout e a disposição dos elementos vai variar de acordo com elas:

320 pixel – Smartphones no formato retrato

480 pixel – Smartphones no formato paisagem

768 pixel – Tablets no formato retrato

960 pixel – Tablets no formato paisagem e alguns monitores mais antigos

1200 pixel – Desktops com monitores widescreen

A div que segura tudo!

E centraliza

Reparem que existem margens na página que estão sobrando.

Vamos tirar todas as margens:

No CSS, na parte relativa a body

```
body {
        margin: 0px;
}
```

VOLTANDO AO NOSSO SITE

Vamos simular um tamanho de site para deixa-lo alinhado

Vamos supor que o layout não seja fluido, e que fique centralizado na tela. No html, criamos a div principal, com tudo dentro. No CSS:

```
#principal {
        width: 1280px;
        margin-left:auto;
        margin-right:auto;    }
```

VOLTANDO AO NOSSO SITE

Vamos simular um tamanho de site para deixa-lo alinhado

O margin-left: auto serve para deixar as margens automáticas, independente da resolução.

Flexbox – a evolução do float

Com os layouts responsivos, precisamos de novas formas de organizar os elementos.

O float organiza os elementos na ordem em que aparecem e usando margins e paddings que as vezes, quando mudamos o tamanho do dispositivo, não se adaptam;

Flexbox – a evolução do float

A ideia é simples: os filhos de um elemento com **flexbox** podem se posicionar em qualquer direção e podem ter dimensões flexíveis para se adaptar.

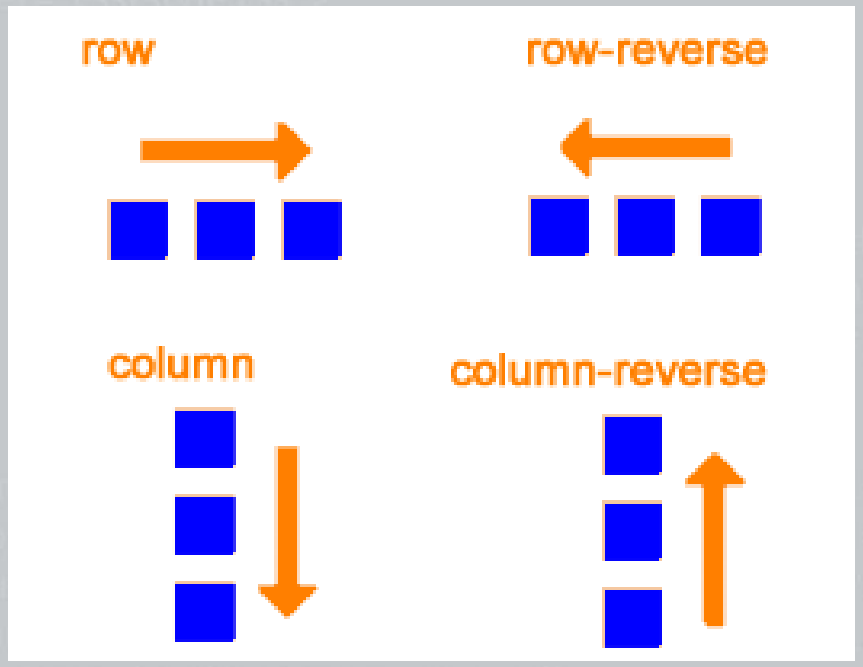
Flexbox – como escrever no css3

- **Propriedade display:** colocamos no nosso box principal, container, main, div principal, que ela é um **display: flex;** Com isso tudo que ficar dentro dela será organizado de forma flexível; A ideia é: o elemento pai será do tipo display:flex; e os filhos dele – que estarão dentro – se comportarão de forma flexível (um ao lado do outro, alinhados, distribuídos...)

Flexbox – como escrever no css3

- **Flex-direction**: ainda dentro do main, podemos especificar:
  - row (padrão): Os itens são organizados em forma de linha da esquerda para a direita;
  - row-reverse: Os itens são organizados em forma exibição em linha da direita para a esquerda;
  - column: Os itens são organizados em forma de colunas iniciando de cima para baixo;
  - column-reverse: Os itens são organizados em forma de colunas iniciando de baixo para cima.

Flexbox – como escrever no css3



Flexbox – como escrever no css3

- **Flex-wrap**: ainda dentro do main, por padrão os itens tentarão se ajustar em uma única linha dentro, mas para isso a sua largura original pode ser ajustada para que todos caibam na largura do elemento pai. Com a propriedade flex-wrap, caso os filhos não caibam na largura do elemento pai, eles irão quebrar pra linha de baixo.

Flexbox – como escrever no css3

- **Flex-wrap:**
- nowrap (padrão): Todos os itens serão dispostos em uma linha;
- wrap: Ocorrerá a quebra de linha e os itens mais à direita serão deslocados para a linha de baixo;
- wrap-reverse: Ocorrerá a quebra de linha e os itens mais à direita serão deslocados para a linha de cima;

Flexbox – como escrever no css3

- **Flex-flow:**
- Serve para combinar as propriedades flex-Direction e flex-wrap!

Ex:

```
.box {
  display: flex;
  flex-flow: row wrap; /* direção em linha com quebra */
 }
```

Flexbox – como escrever no css3

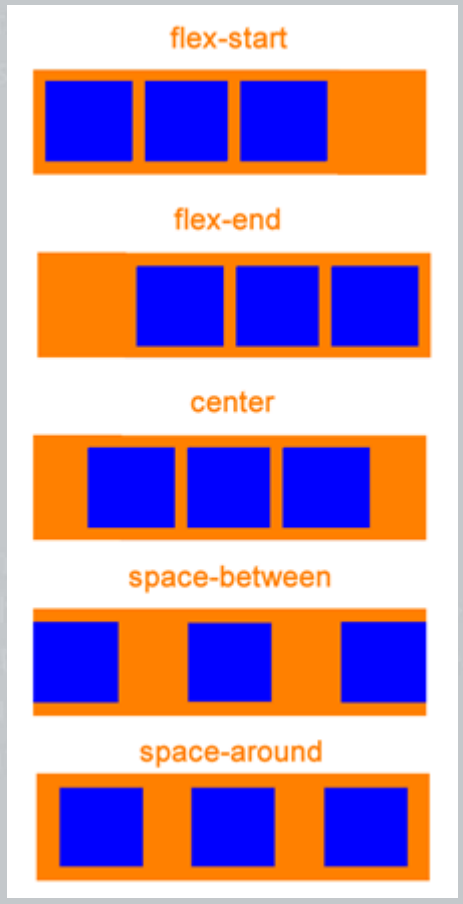
- **justify-content**: A propriedade justify-content define o alinhamento dos itens ao longo do eixo principal do container, se pela esquerda, direita ou centraliza.

Flexbox – como escrever no css3

- **justify-content:**

- flex-start (padrão): Os itens são alinhados a partir do início do eixo principal;
- flex-end: Os itens são alinhados a partir do fim do eixo principal;
- center: Os itens são alinhados ao centro do eixo principal;
- space-between: O primeiro item é deslocado para o início do eixo principal, o último é deslocado para o final do eixo principal e os demais são distribuídos uniformemente entre eles;
- space-around: Os itens são uniformemente distribuídos ao longo do eixo principal. Aqui, porém, são atribuídas margens iguais à esquerda e à direita (ou acima e abaixo, dependendo da direção do eixo principal). Por isso o primeiro e o último item não ficam "colados" nas bordas do container.

Flexbox – como escrever no css3



Flexbox – como escrever no css3

- **Align-items**: Vai alinhar no eixo transversal!
- Flex-start: alinham no topo do container;
- Flex-end: alinham na base;
- Center: centralizam pela altura do container;

- http://flexboxfroggy.com/

Desafio

- Peguem o template que fizemos e, ao invés de float, utilizem o flexbox pra organizar os itens!! Lembrem-se: o elemento pai é o display:flex!!