

# Flex 布局

讲师：张礼军

# 主要内容

- 传统布局方案存在的问题
- 新的布局方案——Flex 布局
- 经典案例——骰子布局

# 传统布局方案存在的问题

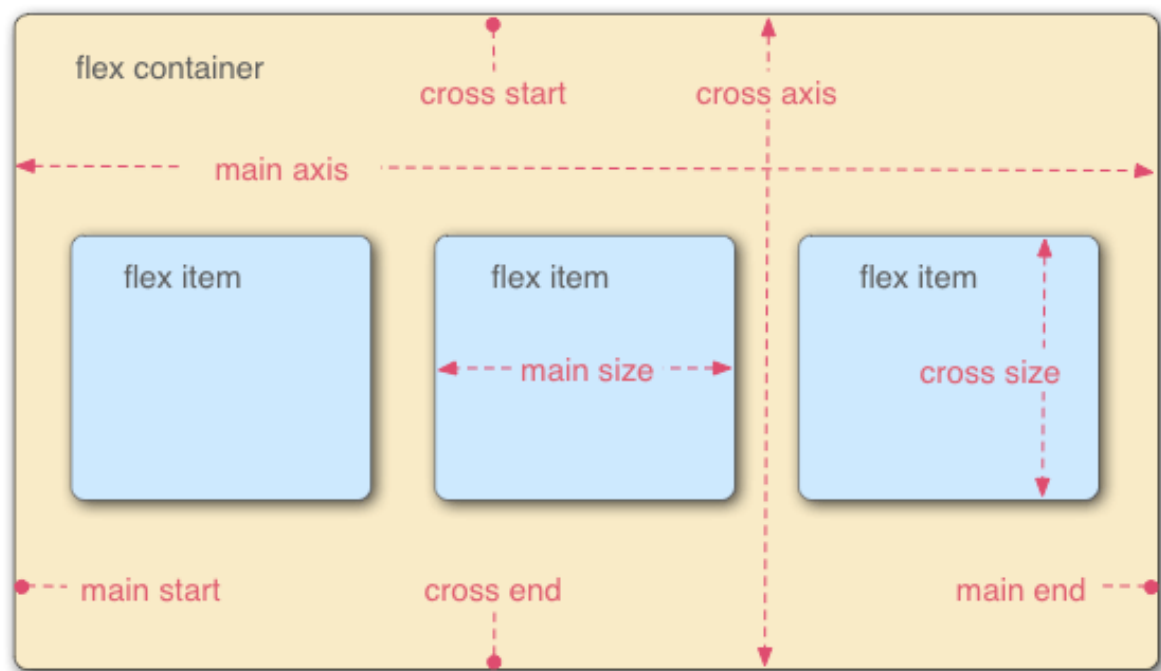
- 传统的布局解决方案是基于盒子模型，主要依赖 `display` 属性、`position` 属性、`float` 属性。但对于那些特殊的布局就非常不方便。比如，垂直居中就不容易实现。
- 案例：传统布局实现 `div` 水平垂直居中。

# 新的布局方案——Flex 布局

- 2009年，W3C 组织提出了一种新的方案——Flex 布局，可以简便、完整、响应式地实现各种页面布局。
- Flex 布局已经成为页面布局的首选方案！
- 不管是什么布局，Flex 往往都可以几行代码搞定！
- 案例：Flex 布局实现 div 水平垂直居中。

# Flex 布局——基本概念

采用 Flex 布局的元素，称为 **Flex 容器**（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为 **Flex 项目**（flex item），简称"项目"。



容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）。主轴的开始位置（与边框的交叉点）叫做 main start，结束位置叫做 main end；交叉轴的开始位置叫做 cross start，结束位置叫做 cross end。

项目默认沿主轴排列。单个项目占据的主轴空间叫做 main size，占据的交叉轴空间叫做 cross size。

# Flex 布局——容器的属性

以下6个属性设置在容器上：

- flex-direction——决定主轴的方向（即项目的排列方向）
- flex-wrap——如果一条轴线排不下，如何换行
- flex-flow——flex-direction属性和flex-wrap属性的简写形式
- justify-content——定义了项目在主轴上的对齐方式
- align-items——定义项目在交叉轴上如何对齐
- align-content——定义了多根轴线的对齐方式（如果项目只有一根轴线，该属性不起作用）

# Flex 布局——项目的属性

以下6个属性设置在项目上：

- `order`——定义项目的排列顺序。数值越小，排列越靠前，默认为0
- `flex-grow`——定义项目的排列顺序。数值越小，排列越靠前，默认为0
- `flex-shrink`——定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小
- `flex-basis`——定义了分配多余空间之前，项目占据的主轴空间
- `flex`——`flex-grow`, `flex-shrink` 和 `flex-basis`的简写，默认值为0 1 auto
- `align-self`——允许单个项目有与其他项目不一样的对齐方式，可覆盖`align-items`属性

# 经典案例——骰子布局

