Day Four

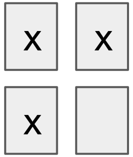
CSS Tables

- CSS Tables
- CSS Box Model
- CSS Layouting

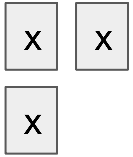
CSS Tables

- empty-cells ➔ show, hide
- border-spacing ➔ [Hpx Vpx], [HVpx]
- border-collapse ➔ collapse, separate

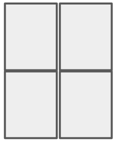
empty-cells: show;



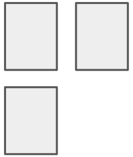
empty-cells: hide;



border-collapse: collapse;



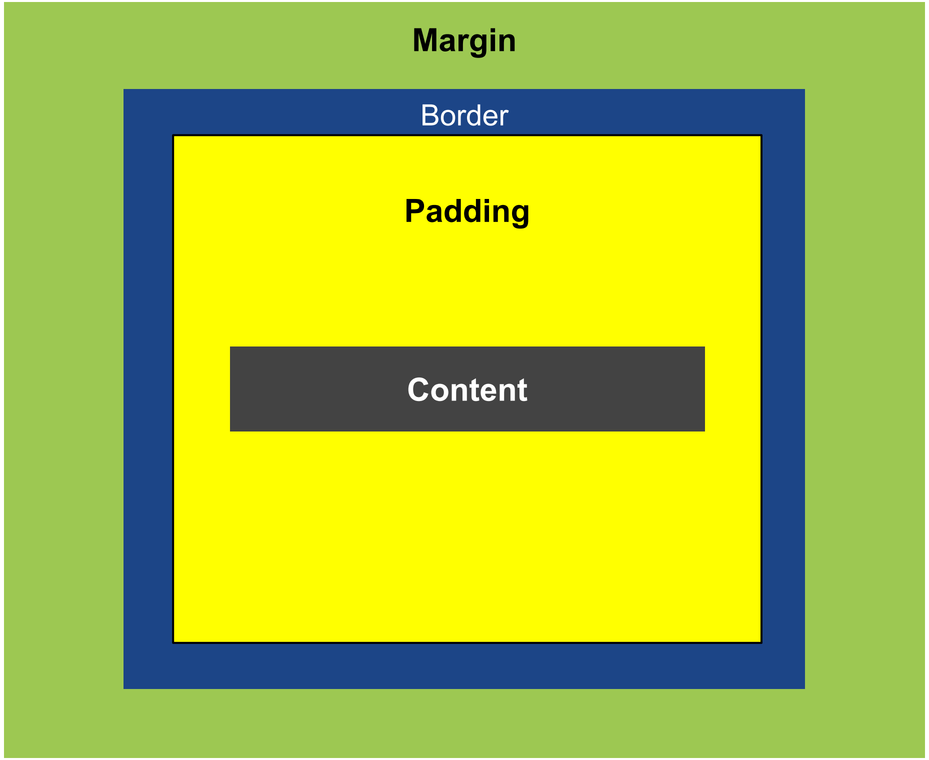
border-collapse: separate;



CSS Layouting

Box Model

- **Content -** The content of the box, where text and images appear
- **Padding** - Clears an area around the content.
- **Border -** A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent



CSS Layouting

Box Model

- height ➔ Height of the content
- width ➔ Width of the content
- margin ➔ Margin of the element, it will not adopt the background color and background-image
- padding ➔ Space between the border and content, it adopts the background-color and background-image
- border ➔ Line that defines the boundaries of the element
- overflow
    - ◆ visible - Default. The overflow is not clipped. The content renders outside the element's box
    - ◆ hidden - The overflow is clipped, and the rest of the content will be invisible
    - ◆ scroll - The overflow is clipped, and a scrollbar is added to see the rest of the content

- box-sizing ➔ Defines how the width and height of an element are calculated: should they include padding and borders, or not (border-box - includes H & W)

- box-shadow ➔ H-offset V-offset Spread Color;

CSS Layouting

Margin/Padding/Border Thickness Shorthands

- margin: 25px 50px 75px 100px;
  - top margin is 25px
  - right margin is 50px
  - bottom margin is 75px
  - left margin is 100px

- margin: 25px 50px;
  - top and bottom margins are 25px
  - right and left margins are 50px

- margin: 25px 50px 75px;
  - top margin is 25px
  - right and left margins are 50px
  - bottom margin is 75px

- margin: 25px;
  - all four margins are 25px

- margin-top: 20px;
- margin-bottom: 25px;
- margin-right: 30px;
- margin-left : 35px;

* border-width property is used to control the thickness of borders

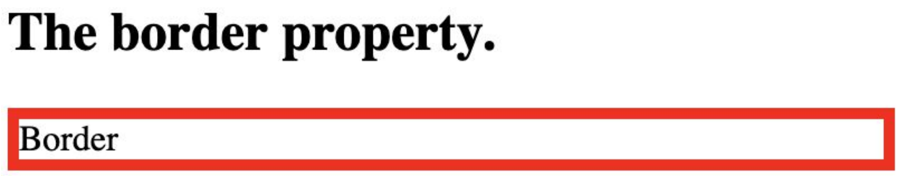
CSS Layouting

Borders

border: [thickness] [style] [color];

Sample:

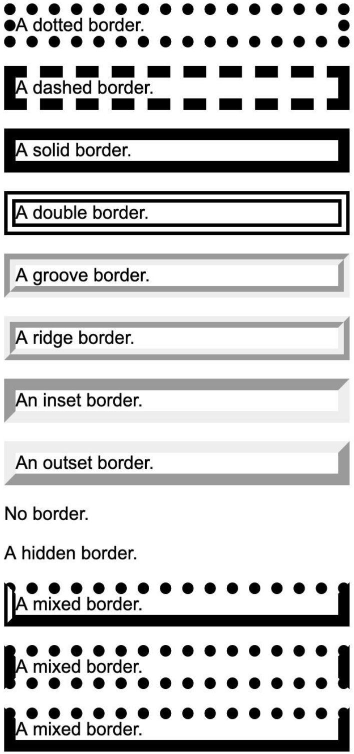
- border: 5px solid red;



The border property.

Border

CSS Layouting



A dotted border. → border-style: dotted;

A dashed border. → border-style: dashed;

A solid border. → border-style: solid;

A double border. → border-style: double;

A groove border. → border-style: groove;

A ridge border. → border-style: ridge;

An inset border. → border-style: inset;

An outset border. → border-style: outset;

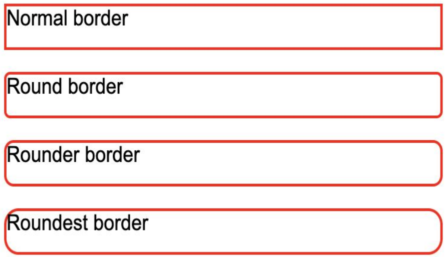
No border. → border-style: none; /* Default */

A hidden border. → border-style: hidden;

A mixed border. → border-style: dotted dashed solid double;

A mixed border. → border-style: dotted dashed;;

A mixed border. → border-style: dotted dashed solid;



Normal border → border-radius: 0px; /*Default*/

Round border → border-radius: 5px;

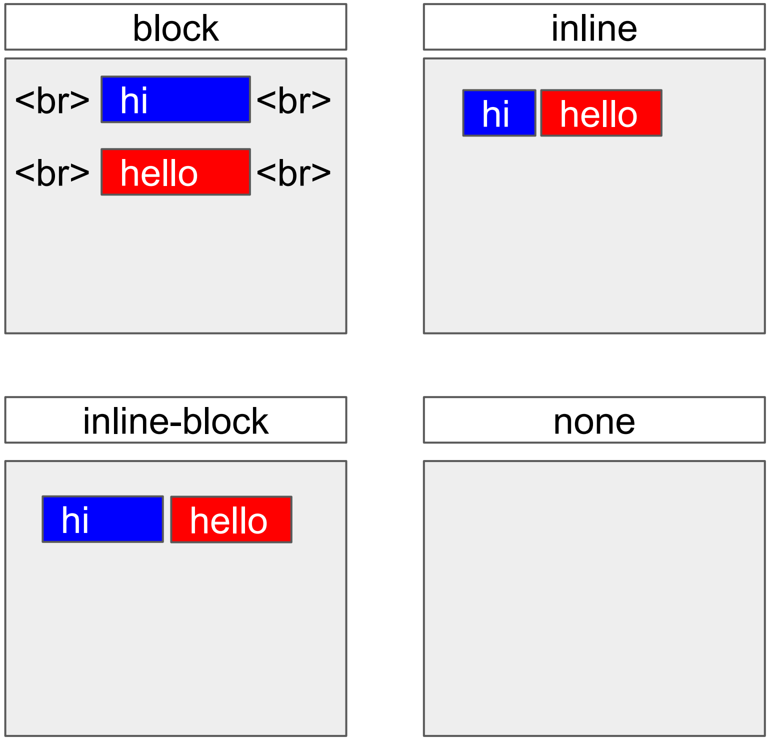
Rounder border → border-radius: 8px;

Roundest border → border-radius: 12px;

CSS Layouting

Display

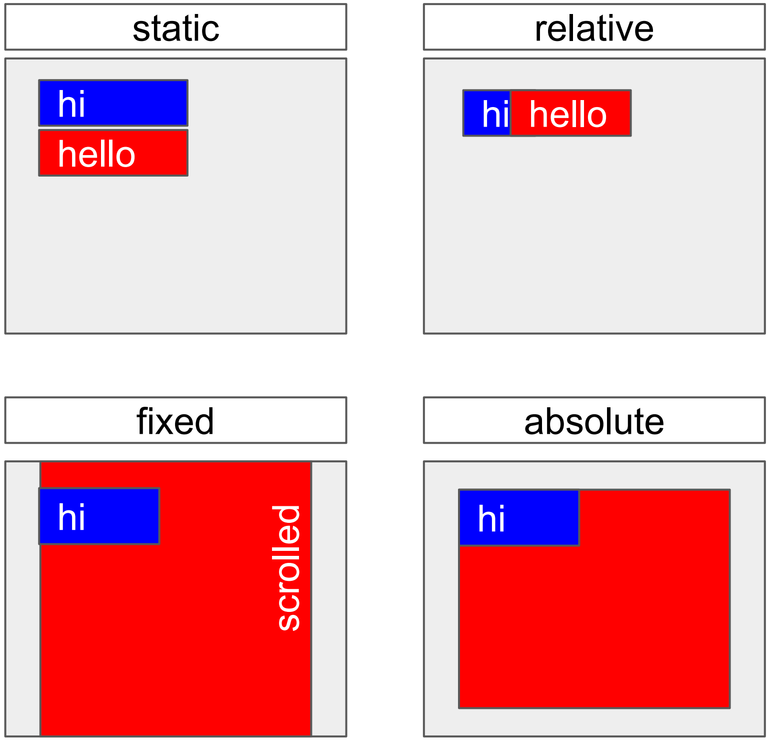
- Sets the layouting behavior of a rectangular box
- Some possible values:
    - **block** - generates line break before and after the element
    - **inline** - does not accept specific height and width
    - **inline-block** - accepts specific height and width
    - **none** - the element won’t be visible
    - plus more advanced values



CSS Layouting

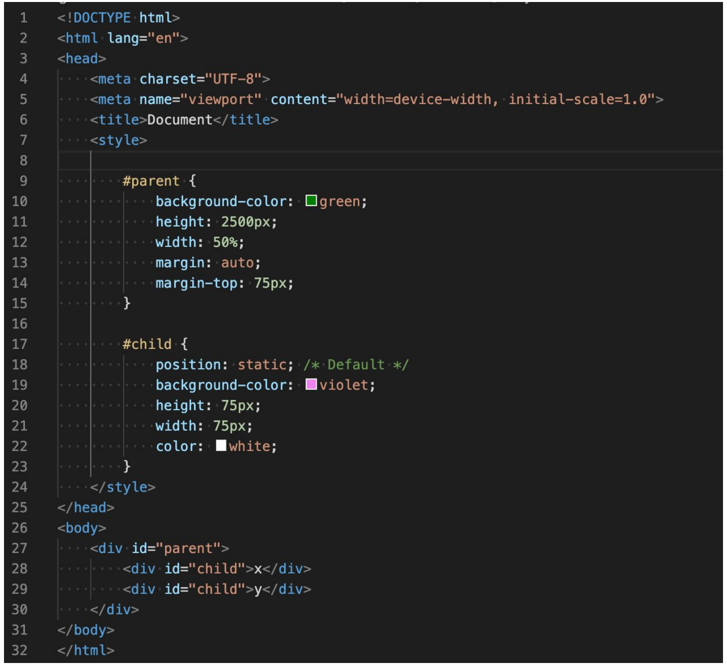
Position

- Specifies the positioning method used for an element
- Some possible values:
  - **static** - Default value. Elements render in order, as they appear in the document flow
  - **relative** - The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
  - **fixed** - The element is positioned relative to the browser window and it won’t follow scrolls. Top, left, right, and bottom properties will still work.
  - **absolute** - It will position itself according to the closest positioned ancestor and it will follow scrolls.

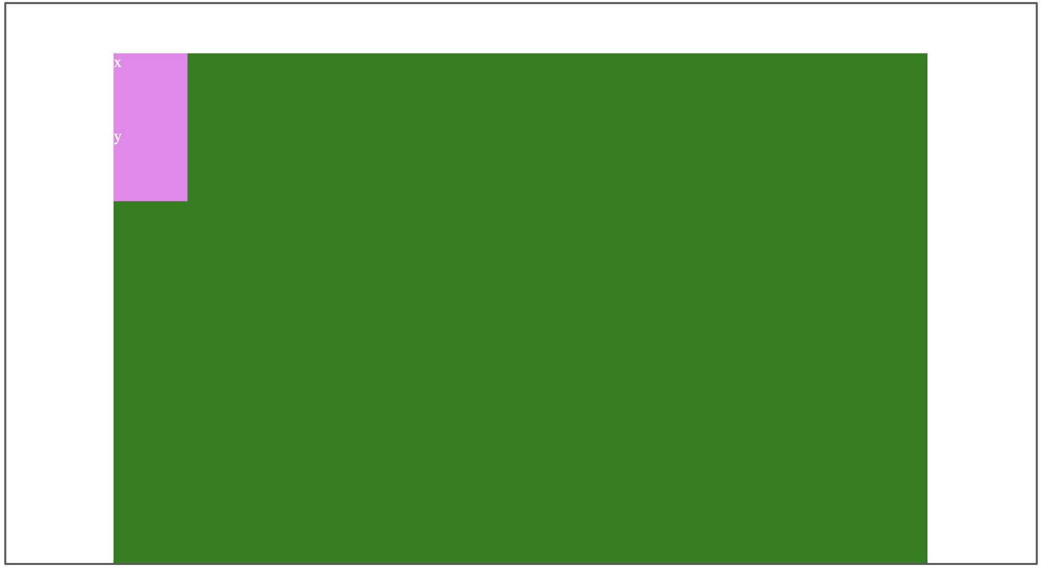


CSS Layouting

position: static;

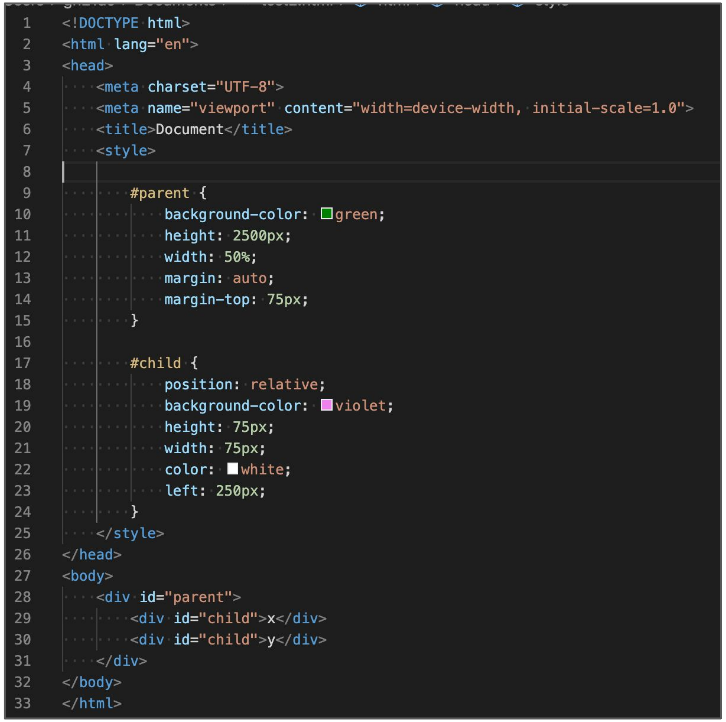


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

        #parent {
            background-color: green;
            height: 2500px;
            width: 50%;
            margin: auto;
            margin-top: 75px;
        }

        #child {
            position: static; /* Default */
            background-color: violet;
            height: 75px;
            width: 75px;
            color: white;
        }
    </style>
</head>
<body>
    <div id="parent">
        <div id="child">x</div>
        <div id="child">y</div>
    </div>
</body>
</html>
```

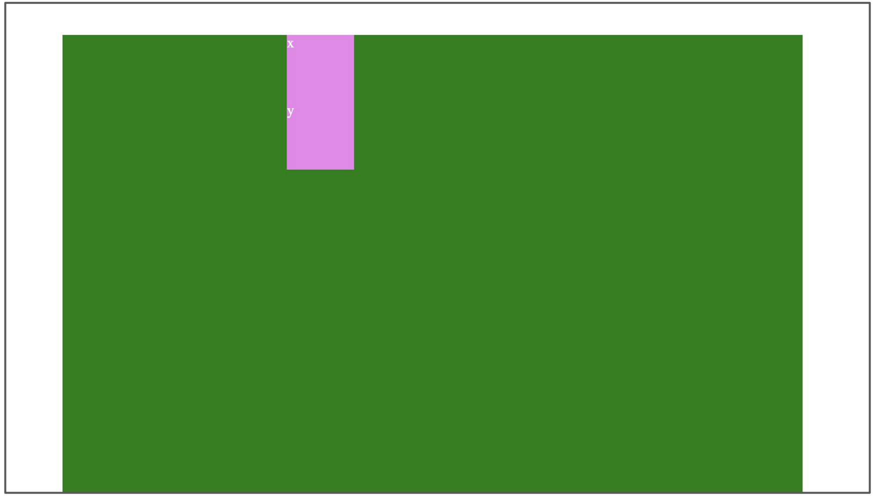


CSS Layouting

position: relative;

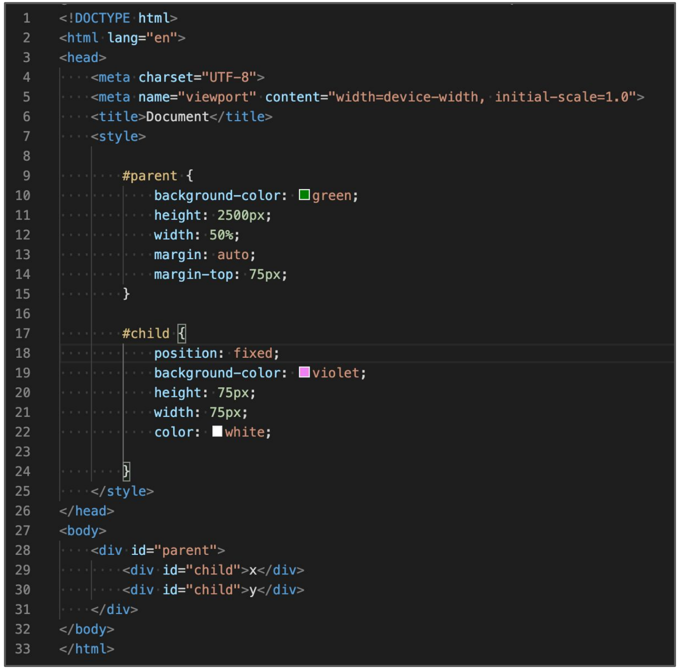


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

        #parent {
            background-color: green;
            height: 2500px;
            width: 50%;
            margin: auto;
            margin-top: 75px;
        }

        #child {
            position: relative;
            background-color: violet;
            height: 75px;
            width: 75px;
            color: white;
            left: 250px;
        }
    </style>
</head>
<body>
    <div id="parent">
        <div id="child">x</div>
        <div id="child">y</div>
    </div>
</body>
</html>
```

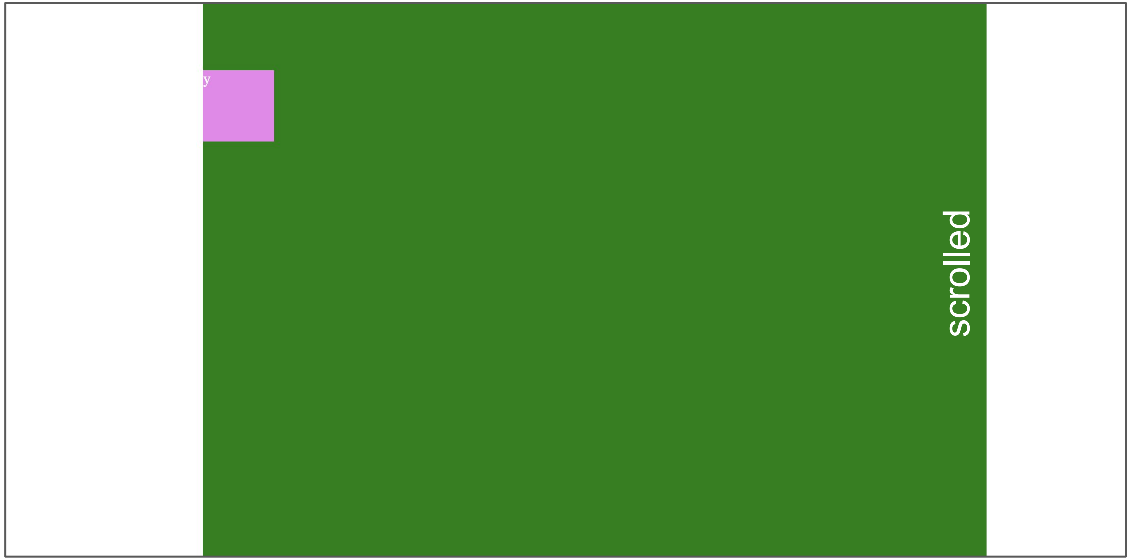


CSS Layouting

position: fixed;

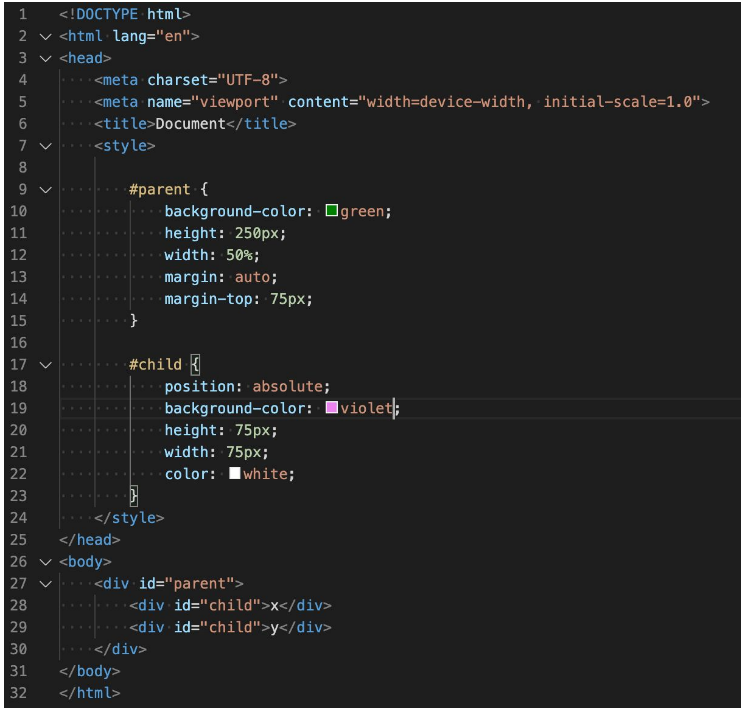


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

        #parent {
            background-color: green;
            height: 2500px;
            width: 50%;
            margin: auto;
            margin-top: 75px;
        }

        #child {
            position: fixed;
            background-color: violet;
            height: 75px;
            width: 75px;
            color: white;

        }
    </style>
</head>
<body>
    <div id="parent">
        <div id="child">x</div>
        <div id="child">y</div>
    </div>
</body>
</html>
```

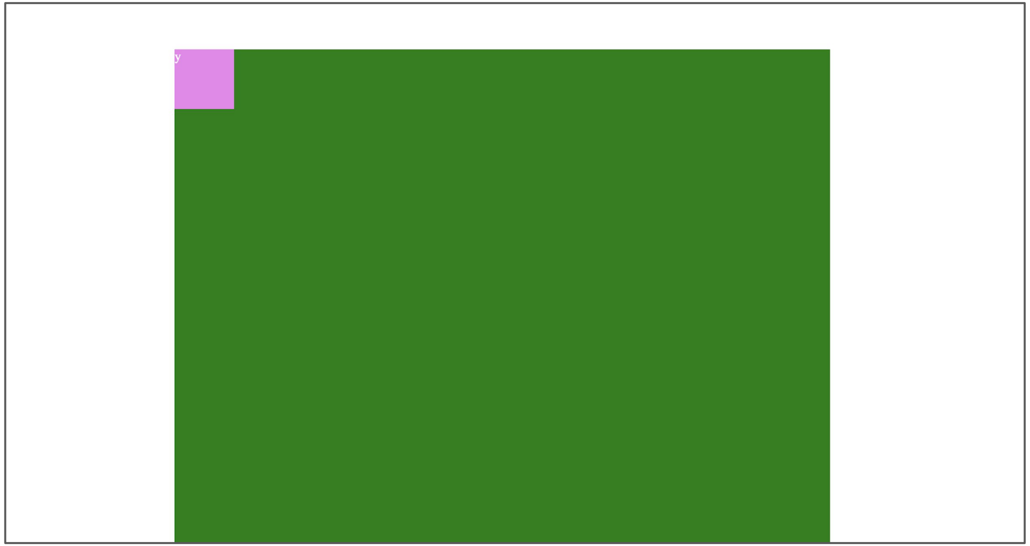


CSS Layouting

position: absolute;

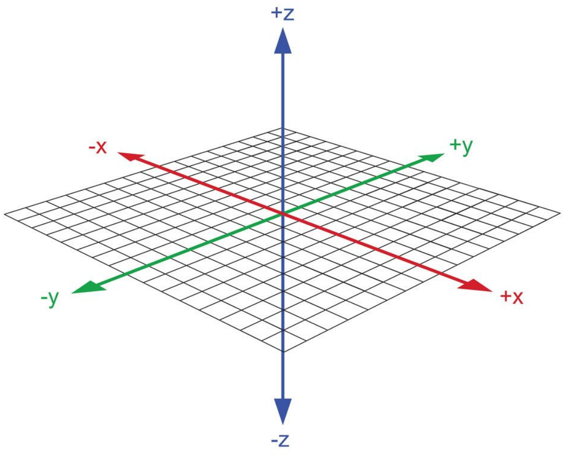


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>

        #parent {
            background-color: green;
            height: 250px;
            width: 50%;
            margin: auto;
            margin-top: 75px;
        }

        #child {
            position: absolute;
            background-color: violet;
            height: 75px;
            width: 75px;
            color: white;
        }
    </style>
</head>
<body>
    <div id="parent">
        <div id="child">x</div>
        <div id="child">y</div>
    </div>
</body>
</html>
```



CSS Layouting

Z-index

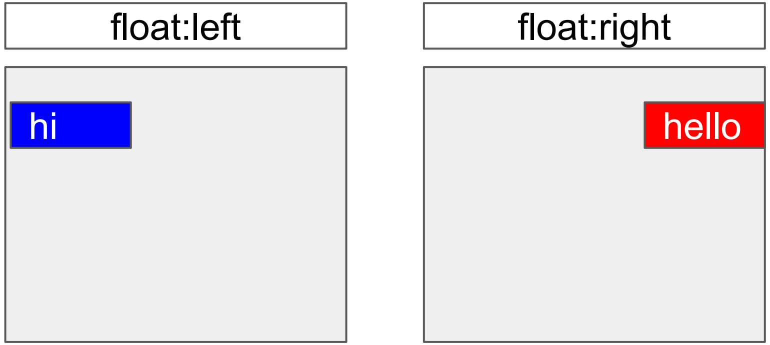


- Specifies the stack order of an element
- Does not work for statically positioned elements
- Default z-index value is 0
- If two elements have the same z-index then the younger element will be positioned in front.

CSS Layouting

Float

- places an element on the left or right side of its container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the page, though still remaining a part of the flow (in contrast to absolute positioning).



float:left

hi

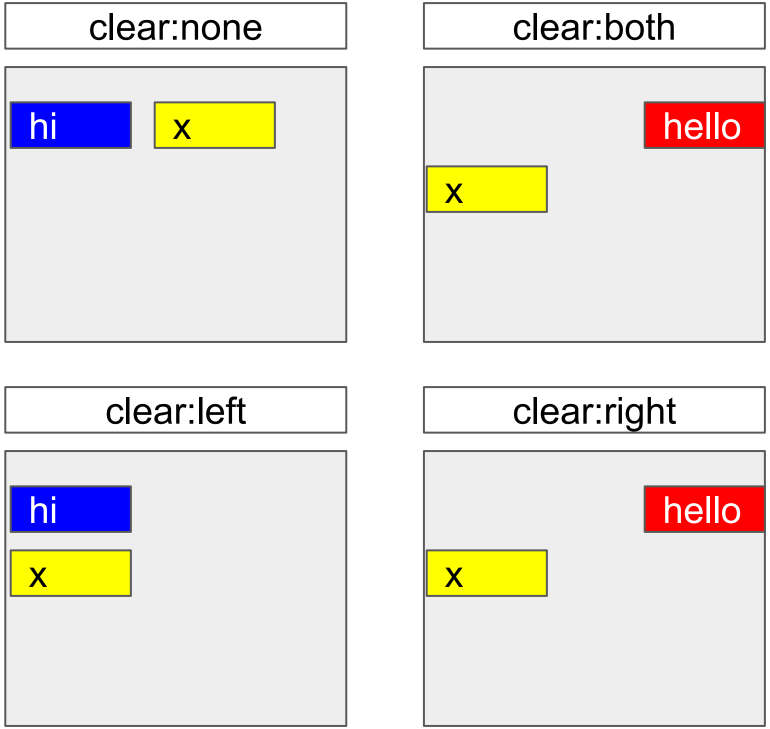
float:right

hello

CSS Layouting

Clear

- Specifies if an element can occupy a line that contains another floating element.
- Possible values:
  - **left** - element won't occupy a line if another element is floating to the left
  - **right** - element won't occupy a line if another element is floating to the right
  - **both** - element won't occupy a line if another element is floating on the same line
  - **none** - element won't care if another element is floating



Topics for tomorrow

- Semantic Tags
- Deployment
- Git
- Github Pages