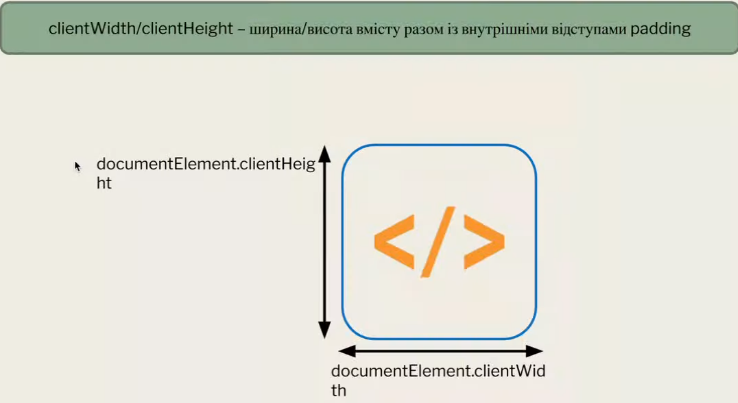
**JavaScript Class And Coordinate**

****

console.log(document.body.clientWidth)

// OutPut: Зависит от HTMLЕлементов, но значением будет ваше разрешение по ширине (не совсем точные цифры)

console.log(document.body.clientHeight)

// OutPut: Зависит от HTMLЕлементов (если нету, то значение будет 0)

JS:

const sizeDiv = document.querySelector('#sizeDiv');

console.log(sizeDiv.clientWidth);

console.log(sizeDiv.clientHeight);

CSS:

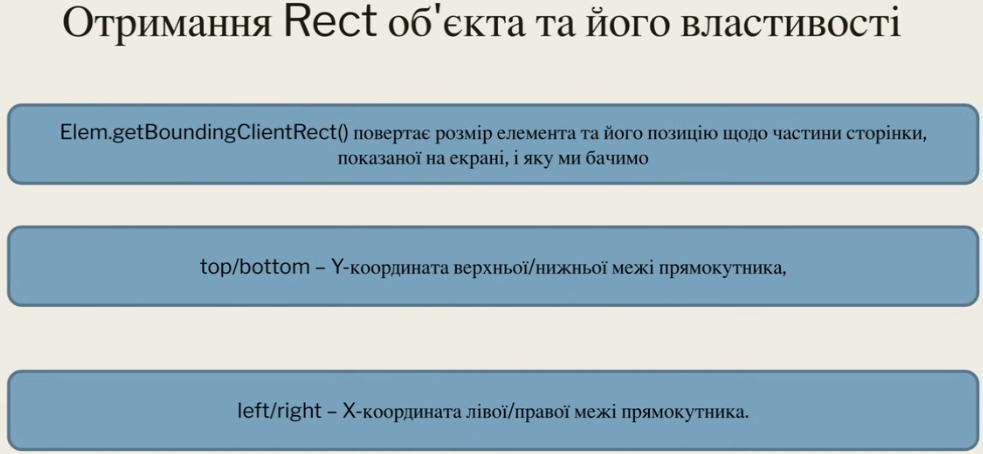
#sizeDiv{

    background-color: red;

    width: 100px;

    height: 200px;

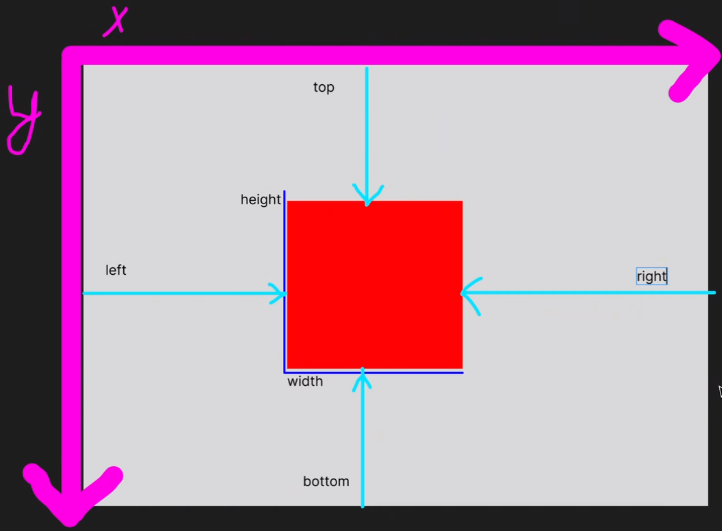
}

****

const rect = sizeDiv.getBoundingClientRect()

console.log(rect)

### OutPut: значение x,y,width,height,top,bottom,left,right



const rect = sizeDiv.getBoundingClientRect()

console.log(`rect.x = ${rect.x},

rect.y = ${rect.y},

rect.width = ${rect.width},

rect.height = ${rect.height},

rect.left = ${rect.left},

rect.top = ${rect.top},

rect.right = ${rect.right},

rect.bottom = ${rect.bottom}`)

CLASS JAVASCRIPT

We compare with python

JavaScript: class == Python: class

JavaScript: constructor ~= Python: \_\_init\_\_

JavaScript: this ~= Python: self

class Transport {

    constructor (name, speed){

        this.name = name

        this.speed = speed

    }

    printName(){

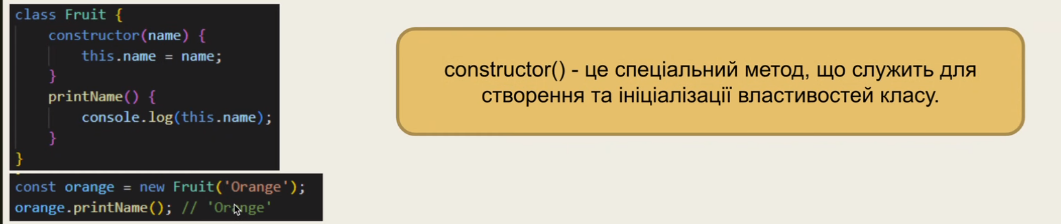
        console.log(this.name)

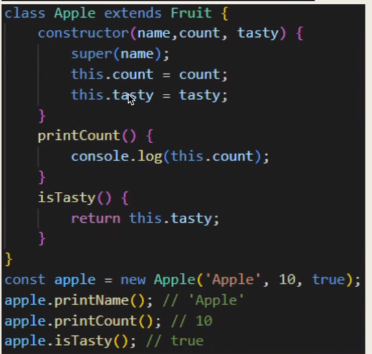
    }

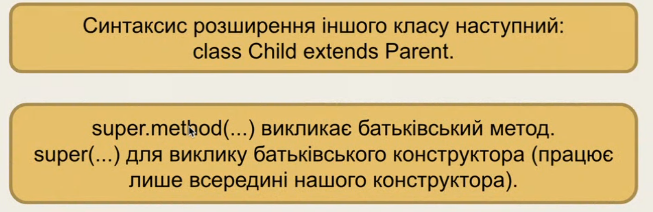
}

const boat = new Transport("boat",30)

boat.printName() //Output: boat



Класс Apple является наследием класса Fruit (Для этого пишем extends название\_родителя)

**Обратите внимание: тип данных из getBoundingClientRect() является DOMElement’ом (DOMRect)**

**Не обязательно писать constructor() для Класса**