**תרגיל בית 6**

**שם: אסלאם גיגיני**

**שאלה 1:**

**זה קורה כאשר יש לנו בן שהוא floated ולכן האב יאבד את ה-height שלו ויהיה שווה ל 0**

**זה קורה כאשר Floated elements מוזזים מה flow הנורמלי של המסמך וכך שאב לא ״רואה״ אותם יותר (לא מוכלים בתוך האב).**

**שאלה 2:**

**תופעות זו נקראת collapsing margins**

**ה-height של האב מוגדר לפי ה-height של ה-content ולא לפי ה-margins שעל ה-content וזה אחד ההגדרות של HTML כל שלא כדי לעזור לנו לשמור על מרחק שווה בין ה-elements.**

**שאלה 3:**

CSS:

.container {

display: flex;

width: 50vw;

height: 50vh;

border: 1px solid blueviolet;

justify-content: center;

align-items: center;

}

.box {

width: 100px;

height: 100px;

margin: 5px;

flex:1;

}

.first {

flex-grow: 1;

background-color: lightseagreen;

}

.second {

background-color: indianred;

}

.third {

background-color: palegreen;

}

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>

<link rel="stylesheet" href="/normalize/normalize.css">

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<div class="box first"></div>

<div class="box second"></div>

<div class="box third"></div>

</div>

</body>

</html>

**שאלה 4:**

CSS:

.container {

display: flex;

width: 50vw;

height: 50vh;

border: 1px solid blueviolet;

justify-content: center;

flex-direction: column;

align-items: center

}

.box {

width: 100px;

height: 100px;

margin: 5px;

flex:1;

}

.first {

flex-grow: 1;

background-color: lightseagreen;

}

.second {

flex-grow:2;

background-color: indianred;

}

.third {

background-color: palegreen;

}

**שאלה 5:**

**CSS:**

.container {

display: flex;

width: 50vw;

height: 50vh;

border: 1px solid blueviolet;

justify-content: center;

flex-direction: column-reverse;

align-items: center

}

.box {

width: 100px;

height: 100px;

margin: 5px;

flex:1;

}

.first {

flex-grow: 1;

background-color: lightseagreen;

}

.second {

flex-grow:2;

background-color: indianred;

}

.third {

flex-shrink: 2;

background-color: palegreen;

}

**שאלה 6:**

**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>

<link rel="stylesheet" href="/normalize/normalize.css">

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<div class="box">Here 1</div>

<div class="box">Here 2</div>

<div class="box">Here 3</div>

<div class="box">Here 4</div>

<div class="box">Here 5</div>

<div class="box">Here 6</div>

<div class="box">Here 7</div>

<div class="box">Here 8</div>

<div class="box">Here 9</div>

<div class="box">Here 10</div>

</div>

</body>

</html>

**CSS:**

.container {

display: flex;

width: 50vw;

height: 50vh;

border: 5px solid gold;

gap:5px;

justify-content: center;

align-items: center;

flex-direction: column;

flex-wrap: wrap;

}

.box {

width: 100px;

height: 100px;

background-color: gold;

display: flex;

align-items: center;

justify-content: center;

font-weight: bold;

}

**To be able to get the second output from the question we need to change**

align-items: center;

**To be:**

align-content: center;

**In the .container CSS**

**שאלה 7:**

**ב-grid יותר נהוג להשתמש ב-fr כי ה-fraction הוא יותר נוח ודינמי (לא צריך לדאוג לחלוקה של 100%) בכל עט שרוצים להוסיף שורה או עמודה לא צריכים לדאוג לחישוב ושינוי חלוקה היא ימלא מקומות בהתאם ובכך אנו לא דואגים מבעיות חישוב שיגרמו ל-overflow.**

**לדוגמה אם רושמים:**

**Grid-template-column: 31% 70% כך אנו בבעריה של Overflow**

**שאלה 8:**

|  |  |
| --- | --- |
| **Justify-content** | **יישור כל ה-grid באופן אופקי בתוך ה-container** |
| **Align-content** | **יישור כל ה-grid באופן אנכי בתוך ה-container** |
| **Justify-items** | **יישור כל התוכן בתוך באופן אופקי בתוך ה-grid** |
| **Align-items** | **יישור כל התוכן בתוך באופן אנכי בתוך ה-grid** |

**שאלה 9:**

**HTML:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Layouts</title>

<link rel="stylesheet" href="/normalize/normalize.css">

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="container">

<div class="box first">

<img src="/lib/images/grapes.jpeg" alt="grapes">

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Laudantium distinctio nobis deserunt similique cum ullam corporis amet placeat hic molestias, repellendus, fugiat, inventore aut harum qui? Numquam nulla laborum nemo ipsa distinctio libero consequuntur aut, error voluptatibus blanditiis fuga commodi quidem architecto et sint reiciendis sed iusto, alias, magni accusantium.</p>

</div>

<div class="box second">

<div class="innerbox">

<img src="/lib/images/coffee.jpeg" alt="coffee" class="secondimg">

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Mollitia sunt, vitae non.</p>

</div>

</div>

<div class="box third">

<div class="block red"></div>

<div class="block green"></div>

<div class="block blue"></div>

</div>

<div class="box fourth">

<div class="block red"></div>

<div class="block green"></div>

<div class="block blue"></div>

</div>

</div>

</body>

</html>

**CSS:**

.container{

display: grid;

grid-template-columns: repeat(2, 1fr);

gap: 10px;

padding:10px;

}

.box{

border: 1px solid black;

padding:10px;

background-color: lightblue;

height: 50vh;;

}

img{

float:left;

width: 150px;

height: 150px;

margin: 0 10px 0 0;

object-fit:fill;

}

p{

text-align: justify;

margin: 0;

text-wrap: wrap;

}

.innerbox{

border: 1px solid black;

display: flex;

}

.third{

display: flex;

flex-direction: column-reverse;

justify-content: space-between;

place-content: flex-end;

}

.block{

border: 1px solid black;

height: 100px;

width: 100px;

}

.red{

background-color: red;

align-self: flex-start;

}

.green{

background-color: green;

align-self: center;

}

.blue{

background-color: blue;

align-self: flex-end;

}

.fourth{

display: flex;

flex-direction: column-reverse;

align-content: space-between;

place-content: flex-start;

}

.fourth .red{

order: 3;

align-self: flex-end;

}

.fourth .blue{

order: 1;

align-self: flex-start;

margin-bottom: 20px;

}